

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ
МЕЖДУНАРОДНАЯ АКАДЕМИЯ ИНФОРМАТИЗАЦИИ
СОЮЗ МАШИНОСТРОИТЕЛЕЙ РОССИЙСКОЙ ФЕДЕРАЦИИ
МИНИСТЕРСТВО ПРОМЫШЛЕННОСТИ, ИННОВАЦИОННЫХ
И ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ РЯЗАНСКОЙ ОБЛАСТИ
РЯЗАНСКИЙ ГОСУДАРСТВЕННЫЙ РАДИОТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ
ИМЕНИ В.Ф. УТКИНА

СОВРЕМЕННЫЕ ТЕХНОЛОГИИ В НАУКЕ И ОБРАЗОВАНИИ

СТНО-2021

**IV МЕЖДУНАРОДНЫЙ НАУЧНО-
ТЕХНИЧЕСКИЙ ФОРУМ**

Сборник трудов

Том 4

Рязань
Book Jet
2021

УДК 004 + 001.1 + 681.2+ 681.2+ 681.3+681.5
С 568

Современные технологии в науке и образовании – СТНО-2021 [текст]: сб. тр. IV междунар. науч.-техн. форума: в 10 т. Т.4./ под общ. ред. О.В. Миловзорова. – Рязань: Рязан. гос. радиотехн. ун-т, 2021; Рязань. – 274 с.,: ил.

Сборник включает труды участников IV Международного научно-технического форума «Современные технологии в науке и образовании» СТНО-2021.

В сборнике освещаются вопросы математического моделирования, новых технологий в радиотехнике, телекоммуникациях, электротехнике и радиоэлектронике, вопросы полупроводниковой наноэлектроники, приборостроения, лазерной, микроволновой техники, силовой промышленной электроники, новые технологии в измерительной технике и системах, биомедицинских системах, алгоритмическое и программное обеспечение вычислительной техники, вычислительных сетей и комплексов, вопросы систем автоматизированного проектирования, обработки изображений и управления в технических системах, перспективные технологии в машиностроительном и нефтехимическом производствах, новые технологии и методики в высшем образовании, в т.ч. вопросы гуманитарной и физико-математической подготовки студентов, обучения их иностранным языкам, перспективные технологии электронного обучения, в том числе, дистанционного, вопросы экономики, управления предприятиями и персоналом, менеджмента, а также вопросы гуманитарной сферы.

Авторская позиция и стилистические особенности сохранены.

УДК 004 + 001.1 + 681.2+ 681.2+ 681.3+681.5

ISBN 978-5-7722-0301-9

© Рязанский государственный
радиотехнический университет, 2021
© Издательство «Book Jet»,
макет, 2021

ИНФОРМАЦИЯ О IV МЕЖДУНАРОДНОМ ФОРУМЕ «СОВРЕМЕННЫЕ ТЕХНОЛОГИИ В НАУКЕ И ОБРАЗОВАНИИ» СТНО-2021

IV Международный научно-технический форум «Современные технологии в науке и образовании» СТНО-2021 состоялся 03.03.2021-05.03.2021 в г. Рязань в Рязанском государственном радиотехническом университете имени В.Ф. Уткина.

В рамках форума «Современные технологии в науке и образовании» СТНО-2021 состоялась работа четырех Международных научно-технических конференций:

«Современные технологии в науке и образовании. Радиотехника и электроника», секции

- Радиотехнические системы и устройства;
- Телекоммуникационные системы и устройства;
- Цифровые информационные технологии реального времени;
- Промышленная силовая электроника, электроэнергетика и электроснабжение;
- Физика полупроводников, микро- и наноэлектроника;
- Микроволновая, оптическая и квантовая электроника;
- Актуальные задачи химических технологий;

«Современные технологии в науке и образовании. Вычислительная техника и автоматизированные системы», секции

- Алгоритмическое и программное обеспечение вычислительных систем и сетей;
- ЭВМ и системы;
- Системы автоматизированного проектирования;
- Информационные системы и защита информации;
- Математические методы в научных исследованиях;
- Обработка изображений и управление в технических системах;
- Геоинформационные и космические технологии;
- Автоматизация производственно-технологических процессов в приборо- и машиностроении;

- Информационно-измерительные устройства и системы в технике и медицине.

Стандартизация и управление качеством;

- Информационные системы и технологии;

«Современные технологии в науке и образовании. Экономика и управление», секции;

- Современные технологии государственного и муниципального управления;
- Экономика, менеджмент и организация производства;
- Бухгалтерский учет, анализ и аудит;
- Управление персоналом;
- Экономическая безопасность;

«Современные технологии в науке и образовании. Новые технологии и методы в высшем образовании», секции

- Современные технологии электронного обучения;
- Иностранный язык в техническом вузе;
- Лингвистика и межкультурная коммуникация;
- Направления и формы гуманитаризации высшего образования и гуманитарная

подготовка студентов;

- Методы преподавания и организация учебного процесса в вузе;
- Физико-математическая подготовка студентов;
- Особенности военного образования на современном этапе.

Организационный комитет Форума:

Чиркин М.В., ректор, д.ф.-м.н., проф. – председатель

Гусев С.И., проректор по научной работе и инновациям, д.т.н., проф. – зам. председателя;

Бабаян П.В., проректор по учебной работе и информатизации, к.т.н., доц. – зам. председателя;

Миловзоров О.В., зам. начальника управления организации научных исследований, к.т.н, доц. – координатор;

Устинова Л.С., начальник отдела информационного обеспечения – отв. за информационную поддержку;

Трубицына С.Г., вед. инженер – секретарь оргкомитета;

Благодарова И.А., ведущий программист – секретарь оргкомитета;

члены оргкомитета:

Авилкина С.В., начальник управления по воспитательной работе, к.п.н., доц.;

Алпатов Б.А., профессор кафедры автоматизации и информационных технологий в управлении, д.т.н., проф.;

Бухенский К.В., зав. кафедрой высшей математики, к.ф.-м.н., доц.;

Витязев В.В., зав. кафедрой телекоммуникаций и основ радиотехники, д.т.н., проф.;

Евдокимова Е.Н., зав. кафедрой экономики, менеджмента и организации производства, д.э.н., проф.;

Еремеев В.В., директор НИИ «Фотон», д.т.н., проф.;

Есенина Н.Е., зав. кафедрой иностранных языков, к.п.н., доц.;

Жулев В.И., зав. кафедрой информационно-измерительной и биомедицинской техники, д.т.н., проф.;

Кириллов С.Н., зав. кафедрой радиоуправления и связи, д.т.н., проф.;

Клейносова Н.П., директор центра дистанционного обучения, к.п.н., доц.;

Ключко В.К., профессор кафедры автоматизации и информационных технологий в управлении, д.т.н., проф.;

Коваленко В.В., зав. кафедрой химической технологии, к.т.н., доц.;

Корячко В.П., зав. кафедрой систем автоматизированного проектирования вычислительных средств, д.т.н., проф.;

Костров Б.В., зав. кафедрой электронных вычислительных машин, д.т.н., проф.;

Кошелев В.И., зав. кафедрой радиотехнических систем, д.т.н., проф.;

Круглов С.А., зав. кафедрой промышленной электроники, к.т.н., доц.;

Куприна О.Г., доцент кафедры иностранных языков, к.филол.н., доц.;

Литвинов В.Г., зав. кафедрой микро- и нанoeлектроники, д.ф.-м.н., проф.;

Лукьянова Г.С., доцент кафедры высшей математики, к.ф.-м.н., доц.;

Мусолин А.К., профессор кафедры автоматизации информационных и технологических процессов, д.т.н., проф.;

Овечкин Г.В., зав. кафедрой вычислительной и прикладной математики, д.т.н., проф.;

Паршин Ю.Н., зав. кафедрой радиотехнических устройств, д.т.н., проф.;

Перфильев С.В., зав. кафедрой государственного, муниципального и корпоративного управления, д.э.н., проф.;

Пржегорлинский В.Н., зав. кафедрой информационной безопасности, к.т.н., доц.;

Пылькин А.Н., профессор кафедры вычислительной и прикладной математики, д.т.н., проф.;

Рохлина Т.А., доцент кафедры иностранных языков, к.филол.н., доц.;

Серебряков А.Е., зам. зав. кафедрой электронных приборов, к.т.н.;

Соколов А.С., зав. кафедрой истории, философии и права, д.и.н.;

Таганов А.И., профессор кафедры космических технологий, д.т.н., проф.;

Харитонов А.Ю., нач. военного учебного центра, полковник, к.т.н., доц.;

Холопов С.И., декан ф-та автоматизации и информационных технологий в управлении, к.т.н., доц.;

Чеглакова С.Г., зав. кафедрой экономической безопасности, анализа и учета, д.э.н., проф..

**МЕЖДУНАРОДНАЯ НАУЧНО-ТЕХНИЧЕСКАЯ КОНФЕРЕНЦИЯ
«СОВРЕМЕННЫЕ ТЕХНОЛОГИИ В НАУКЕ И ОБРАЗОВАНИИ.
ВЫЧИСЛИТЕЛЬНАЯ ТЕХНИКА
И АВТОМАТИЗИРОВАННЫЕ СИСТЕМЫ»**

**СЕКЦИЯ «АЛГОРИТМИЧЕСКОЕ И ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ
ВЫЧИСЛИТЕЛЬНЫХ СИСТЕМ И СЕТЕЙ»**

УДК 004.42; ГРНТИ 50.05.13

**ОПТИМИЗАЦИЯ ЗАВИСИМОСТЕЙ В ПРОГРАММЕ
С ПОМОЩЬЮ ПРАВИЛА ДЕМЕТРЫ**

А.В. Пруцков

*Рязанский государственный радиотехнический университет имени В. Ф. Уткина,
Российская Федерация, Рязань, <http://prutzkow.com>*

Аннотация. Оптимизация зависимостей в программе является одним из принципов написания легко изменяемых программ. Правило Деметры ограничивает классы, методы которых можно вызывать, что сокращает количество зависимостей. Приведен пример нарушения правила Деметры и способ устранения этого нарушения. Выполнен обзор публикаций, посвященных этому правилу.

Ключевые слова: легко изменяемые программы, принципы, правило Деметры, нарушение правила.

**DEPENDENCY OPTIMIZATION IN THE PROGRAM
BY THE LAW OF DEMETER**

A.V. Prutzkow

*Ryazan State RadioEngineering University named after V. F. Utkin,
Russian Federation, Ryazan, <http://prutzkow.com>*

Abstract. Optimizing program dependencies is one of principles of easily-modified programs. The law of Demeter restricts the classes, whose methods can be called, which reduces the number of dependencies. We illustrate the law of Demeter by an example of violation of the law and a way to eliminate this violation. We survey papers on this law as well.

Keywords: easily-modified program, criterion, principles, the law of Demeter, law violation.

Введение

Критерием хорошо написанной программы является низкая трудоемкость ее изменения. В [1] были сформулированы критерий и четыре принципа написания легко изменяемых программ. Принцип 4 гласит: «Зависимости между программными элементами должны быть оптимизированы». Под зависимостью понимается упоминание в тексте одного программного элемента другого программного элемента. Целью оптимизации зависимостей является сокращение времени изменения программы. Оптимизация проводится по двум направлениям:

1) сокращение избыточных зависимостей;

2) замена одних зависимостей другими зависимостями так, чтобы сократить время изменения программы.

Проявлением этого принципа является правило Деметры.

Целью работы является выявление значения правила Деметры для написания легко изменяемых программ.

Правило Деметры

Правило Деметры было предложено в [2]. Оно названо в честь одноименного научного проекта программной системы, разработанной в Северо-Восточном университете (г. Бостон, США) [3].

Это правило заключается в следующем. Метод *M* класса *A* (листинг 1) должен вызывать только методы следующих классов:

- а) класса *A* (строка 8);
- б) классов полей класса *A* (строка 10);
- в) классов параметров метода *M* (строка 12);
- г) классов-утилит (строка 14) (сюда также включаются статические методы классов, не изменяющие состояние объекта).

Подразумевается, что объекты, созданные методами класса *A* или другими методами, а также глобальные объекты передаются в метод *M* через параметры.

Листинг 1. Методы классов, которые можно вызывать в методе *M* класса *A* согласно правилу Деметры

```

1  public class A {
2      private B b;
3
4      public void methodA() {
5          }
6
7      public void m(C c) {
8          this.methodA();
9
10         b.methodB();
11
12         c.methodC();
13
14         D.staticMethodD();
15     }
16 }
```

Правило Деметры приводится в формулировке, адаптированной под современные объектно-ориентированные языки программирования и основанной на правиле для языка C++ из [2].

В более поздней работе [4] правило Деметры расширяется следующим образом. Метод *M* класса *A* (листинг 1) должен вызывать только методы следующих классов:

- а) классы полей класса *A*;
- б) классов параметров метода *M*;
- в) классов объектов, созданных в методе *M*;
- г) классов глобальных переменных, используемых в методе *M*.

Обсуждение правила Деметры в публикациях

М. Саккинен [5] ввел деление всех правил стиля написания программ на 3 типа:

- 1) правила хорошего стиля, если мы можем проверить, что программа удовлетворяет этому правилу;
- 2) правила плохого стиля, если мы можем проверить, что программа не удовлетворяет этому правилу;

3) сомнительные правила во всех остальных случаях.

Автор приводит несколько модифицированных формулировок правила Деметры, пытаясь сделать их проверяемыми. Модифицированные формулировки делятся на два вида: для классов и для объектов. Саккинен делает вывод, что формулировки для классов предпочтительней. Его формулировка правила Деметры для языка C++ включает 5 пунктов и дополнение.

Э. Хант и Д. Томас [6] выделяют преимущества и недостатки использования правила Деметры. К преимуществам они относят адаптируемость программы. К недостатком – делегирование части функций другим классам, что приведет к написанию дополнительных классов и методов. Авторы признают, что денормализация структуры классов может использоваться для повышения производительности программы.

Р. Мартин [7] подчеркивает, что метод *M* не должен вызывать методы объектов, возвращаемых разрешенными для вызова методами.

В [4] предлагается программное обеспечение выявления нарушения правила Деметры. С помощью расчетов данных, полученных с помощью этого программного обеспечения, авторы показывают, что нарушения правила Деметры и обслуживаемость программы взаимосвязаны.

Значение правила Деметры для написания легко изменяемых программ

Правила Деметры сокращает количество зависимостей за счет ограничения классов объектов, методы которых можно вызывать.

Это правило вводит ограничения при проектировании методов следующим образом. Метод, формирующий результат, может быть одним из двух типов:

- 1) возвращающий результат в виде объекта (если это статический метод или метод, возвращающий значения поля);
- 2) изменяющий состояние объекта.

Методы класса `String` языка программирования Java нарушают правило Деметры, так как результаты их работы невозможно использовать в вызываемых методах. Чтобы соблюсти это правило, необходимо разделить класс `String` на два класса:

- 1) класс-данное `String`, который предоставляет доступ к массиву символов – строке;
- 2) класс-утилиты `Strings` со статическими методами обработки строк (аналогичный классам `Arrays` или `Collections`).

Эти классы делят функциональность стандартного класса `String`: первый служит только контейнером для строк, а второй включает методы их обработки.

Пусть необходимо получить первую букву значения логического выражения и сделать ее заглавной. Метод (листинг2, строки 1-6), решающий эту задачу, не удовлетворяет правилу Деметры, так как в нем вызываются методы класса `String`, который не соответствует ни одному разрешенному классу. В тоже время метод (строки 8-14), в котором вызываются методы класса-утилиты `Strings`, удовлетворяет правилу Деметры.

Листинг2. Методы обработки строк, не удовлетворяющие и удовлетворяющие правилу Деметры

```
1 public static String getBooleanValueUpperCasedFirstLetter(boolean value) {
2     String valueText = Boolean.toString(value);
3     String valueFirstLetter = valueText.substring(0, 1);
4     String valueUpperCasedFirstLetter = valueFirstLetter.toUpperCase();
5     return valueUpperCasedFirstLetter;
6 }
7
```

```
8 public static String getBooleanValueUpperCasedFirstLetter(boolean value) {
9     String valueText = Boolean.toString(value);
10    String valueFirstLetter =Strings.substring(valueText, 0, 1);
11    String valueUpperCasedFirstLetter =
12        Strings.toUpperCase(valueFirstLetter);
13    return valueUpperCasedFirstLetter;
14 }
```

↳

Соблюдение правила Деметры сокращает количество зависимостей в программе и делает программы легко изменяемыми.

Заключение

1. Приведена адаптированная под современные объектно-ориентированные языки программирования формулировка правила Деметры. В правило добавлен пункт, разрешающий использование методов классов-утилит и статических методов, не изменяющих внутреннее состояние объекта.

2. Выполнен обзор публикаций, посвященных этому правилу. В публикациях приводятся альтернативные формулировки правила, выявляются преимущества и недостатки его применения, сообщается об автоматизации соответствия программы правилу.

3. Исследовано значение правила Деметры для написания легко изменяемых программ. Показано, что класс String языка программирования Java нарушает правило Деметры. Для устранения этого нарушения предложено разделить класс String на два класса: класс-данное String и класс-утилиту Strings.

Библиографический список

1. Prutzkow, A. Criterion and Principles of Easily-modified Program. In Workshop on Materials and Engineering in Aeronautics, IOP Conference Series: Materials Science and Engineering, 2021, 1027, 012025. DOI: 10.1088/1757-899X/1027/1/012025.

2. Lieberherr, K., Holland, I., Riel, A. Object-Oriented Programming: an Objective Sense of Style. In Conference Proceedings on Object-Oriented Programming Systems, Languages and Applications (OOPSLA), 1988, pp. 323–334.

3. Law of Demeter: Principle of Least Knowledge. URL: <https://www.ccs.neu.edu/home/lieber/LoD.html>, last accessed 2020/11/02.

4. Palm, J., Anderson, K., Lieberherr, K. Investigating the Relationship Between Violations of the Law of Demeter and Software Maintainability. In SPLAT: Software Engineering Properties of Languages for Aspect Technologies, 2003.

5. Sakkinen, M. Comments on «the Law of Demeter» and C++. In SIGPLAN Notices, 1988, 23(12), pp. 38–44.

6. Hunt, A., Thomas, D. The Pragmatic Programmer. Addison-Wesley, 2000.

7. Мартин Р. Чистый код: создание, анализ и рефакторинг. Библиотека программиста: пер. с англ. – СПб.: Питер, 2010. – 464 с.

УДК 007:681.512.2; ГРНТИ 28.23.37

НЕЙРОННЫЕ СЕТИ, ИСПОЛЬЗУЮЩИЕ МОДЕЛИ ЗНАНИЙ

И.Ю. Каширин

*Рязанский государственный радиотехнический университет имени В.Ф. Уткина,
Российская Федерация, Рязань, igor-kashirin@maul.ru*

Аннотация. Анонсируется оригинальная концепция проектирования нейронных сетей с использованием классической модели представления знаний в онтологической форме. Для анализа знаний применяется вычисление семантической близости концептов и признаков, использующее прикладную ICF-онтологию. В вычислении семантической близости применяется математический аппарат алгебраической системы иерархических чисел. Проведенные эксперименты на основе инструментария Python v.3 показывают эффективность предложенного подхода.

Ключевые слова: нейронная сеть, ICF-онтология, иерархические числа, семантическая близость, кластеризация, классификация.

NEURAL NETWORKS USING KNOWLEDGE MODELS

I.Yu. Kashirin

*Ryazan State Radio Engineering University named after V.F. Utkin,
Russia, Ryazan, igor-kashirin@maul.ru*

The summary. An original concept for designing neural networks using the classical model of knowledge representation in ontological form is announced. For the analysis of knowledge, the computation of the semantic proximity of concepts and features is applied using the applied ICF ontology. In calculating semantic proximity, the mathematical apparatus of the algebraic system of hierarchical numbers is used. The experiments carried out on the basis of the Python v.3 toolkit show the effectiveness of the proposed approach.

Keywords: neural network, ICF-ontology, hierarchical numbers, semantic proximity, clustering, classification.

Анализ больших данных сегодня использует алгоритмы классификации и кластеризации, работающие с обучающими и тестирующими выборками. Такими алгоритмами могут служить, например, методы k-средних, опорных векторов, иерархической кластеризации, нейронных сетей и т.д. Трудно отнести эти концепции к сфере искусственного интеллекта, более правильно идентифицировать их как разделы прикладной математики.

В то же время результаты, полученные в области классификации и кластеризации в рамках теории представления знаний [1], дают возможность оценить их как весьма эффективный инструментарий.

В рамках технологий разработки гибридных систем в настоящей статье предлагается использовать комбинирование прикладных технологий анализа данных и теории онтологических моделей представления знаний. В частности, можно рассмотреть использование нейронных сетей совместно с ICF-онтологиями [2] в надежде на синергетический эффект.

В качестве примера будет использоваться предметная область «Работа с клиентурой телекоммуникационной компании», взятая из широко известного международного репозитория Kaggle [3]. Фрагмент данных приведен в таблице 1. Задачей интеллектуального анализа данных поставим определение характеристик клиентов, активно пользующихся услугами компании и не собирающимися прекращать с ней отношения.

В предлагаемом подходе предварительная выборка и подготовка данных производится методами, принятыми в известных алгоритмах, реализованных в инструментарии Python 3, Anaconda. Приведем фрагмент программы на Python 3.

```
#-*- coding: utf-8 -*-  
import pandas as pd  
import numpy as np  
df = pd.read_csv("WA_Fn-UseC_-Telco-Customer-Churn.csv")
```


Далее рассмотрим онтологическое представление предложенной предметной области в ICF-онтологии, обобщив ее до области «Сфера оказания услуг» (рисунок 1).

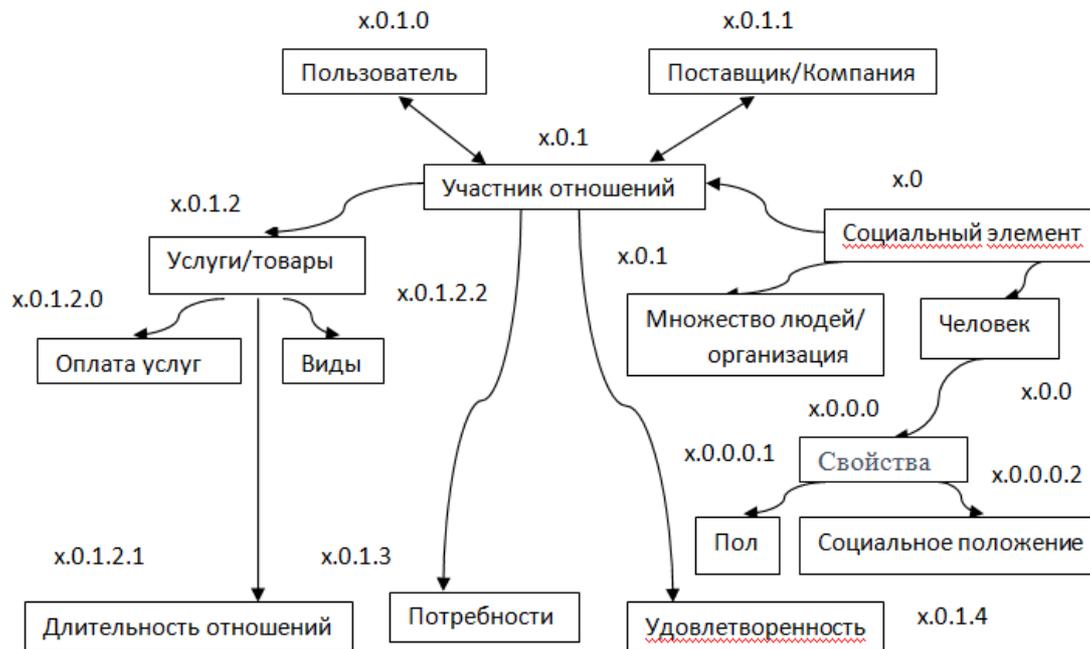


Рис. 1. Онтология предметной области «Сфера оказания услуг»

Здесь стрелками обозначены соответствующие ICF-отношения между концептами, а индексами вершин онтологии являются иерархические числа, используемые в гибридном алгоритме классификации для вычисления семантической близости концептов.

В онтологии на основе предыдущей серии экспериментов с другими предметными областями формулируется правило: чем меньше иерархическое число как расстояние между потребностью и другими концептами, тем более числовые характеристики этих концептов влияют на удержание пользователя в списке потребителей.

Семантически близки, например: уровень доходов, оплата услуг, длительность потребления услуг.

Кроме того, можно воспользоваться библиотекой готовой причинно-следственной таксономии сферы услуг, представленной рисунком 2.

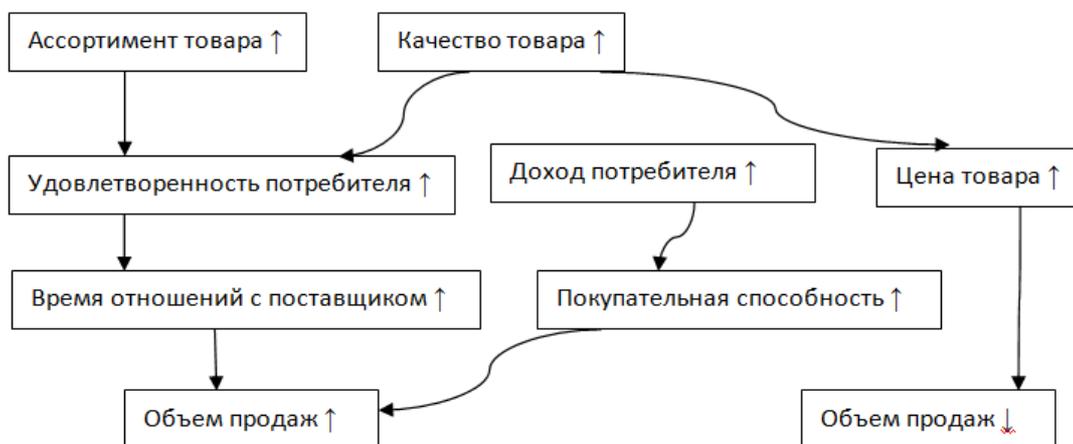


Рис. 2. Причинно-следственная онтология предметной области «Сфера услуг»

В представленном и весьма упрощенном фрагменте таксономии стрелками ↑ обозначено повышение числовой характеристики, а стрелками ↓ – понижение.

Руководствуясь ранее сформулированным правилом влияния семантической близости концептов на соответствующие числовые характеристики предметной области, необходимо выделить концепты и характеристики, наиболее влияющие на результат. В предложенном примере – это информационное поле churn, определяющее, остается ли пользователь компании на телекоммуникационном обслуживании или отказывается от него, соответственно значениями поля являются «yes» или «no».

Рассмотренный метод выбора характеристик для приоритетного анализа реализован в программе экспериментального стенда для методов интеллектуального анализа данных (ЭСМИАД v.12.02.2021), реализованной на языке Python v.3.7 в среде Anaconda3. Эта программа позволяет включать и отключать при анализе данных различные характеристики с последующим вычислением результата обучения.

Реализована предварительная обработка данных: обработка пропусков данных, поиск неявных дубликатов, обнаружение выбросов, числовое кодирование, нормализация, разделение на подвыборки. Алгоритмы различных методов обучаются, анализируют данные и оцениваются на эффективность. Можно использовать для целей обучения и последующего анализа данных нейронную сеть. Фрагмент испытательного стенда приведен далее.

```
#Разделение признаков и меток
features = dfs.drop('Churn', 1)
labels = dfs['Churn']
#Разобъем на обучение и тестинг
from sklearn.model_selection import train_test_split
features_train, features_test, labels_train, labels_test = train_test_split(features, labels,
test_size=0.2, random_state = 27)
# ...
#работа с нейросетью
from keras.models import Sequential
from keras.layers import Dense
import numpy
# задаем для воспроизводимости результатов
numpy.random.seed(2)
# можно разбить dfs на матрицу параметров (X) и вектор целевой переменной (Y)
# X, Y = dfs[:,0:8], dfs[:,8], но уже есть признаки и метки features, labels
# создаем модели, добавляем слои один за другим
model = Sequential()
model.add(Dense(12, input_dim=8, activation='relu'))
# входной слой требует задать input_dim
model.add(Dense(15, activation='relu'))
model.add(Dense(8, activation='relu'))
model.add(Dense(10, activation='relu'))
model.add(Dense(1, activation='sigmoid'))
# сигмоида вместо relu для определения вероятности
# компилируем модель, используем градиентный спуск adam
model.compile(loss="binary_crossentropy", optimizer="adam", metrics=['accuracy'])
# обучаем нейронную сеть
model.fit(features, labels, epochs = 1000, batch_size=10)
# оцениваем результат
scores = model.evaluate(features, labels)
```

```
print("+++++Score of Neural network+++++")
#print("\n%s: %.2f%%" % (model.metrics_names[1], scores[1]*100))
print("\n %.2f%%" % (scores[1]*100))
```

Как показывает эксперимент, характеристики эффективности классификации подтверждают справедливость сформулированного ранее правила выбора характеристик. Далее приводится соответствующий скриншот.

```
0.7203690560681334
0.7459190915542938
[[1015 394]
 [ 0 0]]
      precision    recall  f1-score   support

     0       0.88      0.79      0.83       1121
     1       0.41      0.56      0.48        288

   accuracy          0.75       1409
  macro avg          0.64      0.68      0.65       1409
 weighted avg          0.78      0.75      0.76       1409

+++++Score of Tree+++++
[0.81560284 0.77304965 0.72340426 0.80851064 0.74468085
0.76595745
 0.78723404 0.77304965 0.78014184 0.75      ]
+++++Class Names+++++
[0, 1]
+++++Result+++++
[1 0 1 ... 0 1 1]
0.7721788502484032
```

Добавление информационного поля `gender` (пола) и «Признака пенсионера» делает менее эффективными методы классификации, в частности, нейронной сети. Добавление информационного поля `tenure` делает нейросеть более эффективной.

```
+++++Score of Tree+++++
[0.78014184 0.75886525 0.75177305 0.78723404
0.73758865 0.81560284
 0.79432624 0.78723404 0.85106383 0.75714286]
+++++Class Names+++++
[0, 1]
+++++Score of Neural network+++++
+++
[1 0 1 ... 0 0 0]
0.7899219304471257
```

Библиографический список

1. Каширин Д.И., Каширин И.Ю. Модели представления знаний в системах искусственного интеллекта. Вестник Рязанского государственного радиотехнического университета. № 1 (выпуск 31) Рязань, 2010. – с.36-43.
2. Каширин И. Ю. Иерархические числа для проектирования ICF-таксономий искусственного интеллекта. Вестник Рязанского государственного радиотехнического университета. 2020. – № 71. – С. 71-82.
3. Международный репозиторий для анализа данных и оригинальных технологических решений. [Электронный ресурс]. 2020. Дата обновления: 05.10.2020. URL: <https://www.kaggle.com/> (дата обращения: 05.10.2020).

УДК 005.519.8; ГРНТИ 28.23.37

О РЕШЕНИИ ПРАКТИЧЕСКИХ ЗАДАЧ С ИСПОЛЬЗОВАНИЕМ ГЛУБОКИХ НЕЙРОННЫХ СЕТЕЙ НА БАЗЕ ОБЛАЧНЫХ ТЕХНОЛОГИЙ

Н.И. Цуканова, Н.В. Головкин, О.В. Шурыгина

*Рязанский государственный радиотехнический университет имени В.Ф. Уткина,
Российская Федерация, Рязань, e-mail: ninakorobova77@gmail.com*

Аннотация. В работе рассматриваются вопросы применения облачных и современных информационных технологий к решению практических задач с использованием глубоких нейронных сетей.

Ключевые слова: облачные технологии, виртуальная среда GoogleColaboratory, глубокие нейронные сети, JupyterNotebook, полносвязные нейронные сети, автокодировщики, рекуррентные нейронные сети, язык Python, библиотека Keras .

ON THE SOLUTION OF PRACTICAL PROBLEMS USING DEEP NEURAL NETWORKS CLOUD-BASED

N.I. Tsukanova, N.V. Golovkin, O.V. Shurygina

*Ryazan State Radio Engineering University named after V.F. Utkin,
Russian Federation, Ryazan, e-mail: ninakorobova77@gmail.com*

The summary. The paper deals with the application of cloud and modern information technologies to solving practical problems using deep neural networks.

Keywords: cloud technologies, Google Colaboratory virtual environment, deep neural networks, Jupyter Notebook, fully connected neural networks, autoencoders, recurrent neural networks, Python language, Keras library.

В последние годы одним из самых быстро развивающихся научных направлений стало направление, связанное с нейронными сетями (НС) и их использованием при решении практических задач. Нейронные сети применяются к обработке и анализу изображений, текстов, аудио и видео файлов. Кроме того, они хорошо показали себя при генерации новых объектов: новых изображений, текстов, мелодий и видеоклипов. Все это стало возможным благодаря быстрому развитию аппаратного, математического и программного обеспечения процессов создания и обучения нейронных сетей. В настоящее время такими программными продуктами являются язык Python, библиотеки Keras, TensorFlow, PyTorch, OpenCV и т.п., аппаратными средствами – графические и тензорные процессоры. Развитие нейронных сетей уже идет не по пути наращивания количества нейронов, а по пути сборки НС из различных слоев или сборки системы из нескольких взаимодействующих нейронных сетей. Нейронные сети с большим количеством слоев (больше 2) называются глубокими нейронными сетями (ГНС).

Основные технические и программные инструменты создания глубоких нейронных сетей

Основными элементами являются Google Диск, GoogleColaboratory, персональный компьютер в качестве управляющего звена. Каждый элемент имеет свое назначение. Роль хранилища информации выполняет Google Диск. GoogleColaboratory — это облачный сервис, разработанный командой GoogleResearch, направленный на упрощение исследований в области разработки глубоких нейронных сетей, в основном применяется для машинного обучения, образовательных проектов и анализа данных.

Используя Colaboratory, пользователи получают совершенно бесплатно удаленный доступ к машине с подключенной видеокартой. Благодаря этому все вычисления можно проводить на графическом или тензорном процессоре, что значительно ускоряет обучение ней-

росетей. Кроме того, пользователи получают доступ к программному обеспечению, включающему самые последние версии языка Python и все необходимые для работы с нейросетями библиотеки (Keras, TensorFlow, Numpy и т.д.). Этот сервис позволяет писать и выполнять произвольный код Python с помощью браузера.

Документы, создаваемые в GoogleColaboratory, основаны на платформе JupyterNotebook и называются ноутбуками (блокнотами). В них 2 вида ячеек: текстовые ячейки и ячейки кода на языке Python. В текстовых ячейках могут располагаться 1) поясняющий решение задачи текст; 2) рисунки и т.д. В ячейках кода располагаются фрагменты программ на языке Python, которые могут быть выполнены простым нажатием на стрелку в ячейке.

Разработка нейронных сетей для решения практических задач

В статье рассматриваются два ноутбука (электронных блокнота), разработанных студентами, соавторами данной статьи. Один из ноутбуков (автор Шурыгина Ольга) посвящен вопросам интеллектуальной обработки русскоязычного текста [2], а именно, классификации текста по различным темам. Второй (автор Головкин Никита) - использованию автокодировщиков для предобучения и выделения наиболее важных признаков, что в дальнейшем позволяет более качественно решить задачу классификации изображений полносвязными нейронными сетями.

Разработка ноутбука «Классификация текстовых сообщений с помощью рекуррентных нейронных сетей»

В настоящее время большое внимание уделяется вопросам применения нейронных сетей при интеллектуальной обработке текстовой информации [2], которая, в частности, включает такие задачи, как классификация текста, определение его тональности, кластеризация текста [1] и другие. По этой тематике существует много публикаций применительно к англоязычным текстам, имеются размеченные наборы данных. Однако по обработке русскоязычных текстов опубликованных работ в настоящее время мало, поэтому методика применения глубоких нейронных сетей к интеллектуальному анализу русскоязычного текста является актуальной задачей, требующей решения и исследования.

В дальнейшем рассматривается задача определения тематики статьи по ее заголовку и ее решение выполнено в виде ноутбука, содержащего текстовые ячейки и ячейки кода. В текстовых ячейках пояснены основные этапы обработки русскоязычного текста.

При работе с текстом важен этап его предобработки. Так как нейронная сеть работает только с числовой информацией, то текст надо преобразовать в числа. В данной работе предобработка текста включала следующие шаги: разделение текста на токены (слова), рассматривались различные виды кодирования токенов (кодирование частотой использования, onehot кодирование, кодирование плотными векторами (Embedding)), усечение текста до определенных размеров. Каждый из этих шагов описан и разъясняется в текстовых полях.

В работе рассмотрены две конфигурации нейронной сети: рекуррентная нейронная сеть, основанная на ячейках LSTM, и одномерная свёрточная нейронная сеть [2]. Оценка качества каждой из этих архитектур показала (таблица 1), что большую точность обеспечивает сеть на ячейках LSTM. На новых данных также были получены неплохие результаты.

Таблица 1. Оценка качества нейронных сетей

	Рекуррентная нейронная сеть, основанная на ячейках LSTM	Одномерная свёрточная нейронная сеть
Точность на обучающей выборке	0.9632	0.9530
Точность на проверочной выборке	0.9310	0.9287
Точность на тестовой выборке	0.9301	0.9281

Простонародная мистика позднего Средневековья. (к теории перформативной практики) Результат LSTM: LanguagesAndLiterature Результат сверточной сети: HistoryAndArchaeology Правильный результат: LanguagesAndLiterature
“Трудный рост” Николая Лескова Результат LSTM: LanguagesAndLiterature Результат сверточной сети: LanguagesAndLiterature Правильный результат: HistoryAndArchaeology
Нефть - это глобально! Результат LSTM: HistoryAndArchaeology Результат сверточной сети: EconomicsAndBusiness Правильный результат: HistoryAndArchaeology
Значение оценки компенсации ущерба наземным животным для изучения биоразнообразия и её особенности Результат LSTM: EconomicsAndBusiness Результат сверточной сети: EconomicsAndBusiness Правильный результат: HistoryAndArchaeology

Рис. 1. Примеры ошибок нейронной сети

Для того чтобы понять причину, по которой в некоторых случаях нейронная сеть неправильно классифицирует заголовки статей, были рассмотрены эти заголовки и результаты, полученные с помощью рекуррентной нейронной сети, основанной на ячейках LSTM, и с помощью одномерной сверточной нейронной сети. На рисунке 1 показаны примеры таких заголовков. Из примеров становится понятно, что нейронная сеть совершает ошибки в случаях, когда название статьи неоднозначно и статья может принадлежать нескольким разделам.

Одним из самых важных и сложных этапов решения задач классификации текстов является формирование обучающей выборки. В интернете можно много найти текстовой информации, однако собрать нужные примеры в выборку и пометить каждый пример (указать класс, к которому он принадлежит) оказывается совсем непростой задачей. В данной работе рассматриваются два способа формирования выборки: в первом англоязычная информация извлекается с помощью запроса к базе данных HackerNews на веб-сервисе bigQuery, во втором рассматриваются русскоязычные статьи с сайта CyberLeninka.

Для формирования русскоязычного набора данных написана программа на языке Java, с помощью которой был получен набор (более 700000 примеров) с заголовками статей и указанием источника статьи (всего 5 источников). Этот набор данных был сохранен в файле Articles.csv. В качестве источников статей были выбраны рубрики:

- EconomicsAndBusiness
- ClinicalMedicine
- EducationalSciences
- LanguagesAndLiterature
- HistoryAndArchaeology

Для каждого источника считывалась информация с 7000 страниц (по 20 статей на странице).

Программа обрабатывает 1 страницу в секунду, так что создание файла занимает несколько часов. После создания файла необходимо открыть его текстовым редактором и в

начало дописать заголовок файла source, title. Для более удобного использования программы, был написан графический интерфейс.

Разработка ноутбука «Применение автокодировщиков для предобучения и выделения наиболее важных признаков».

В этой части статьи рассматривается задача классификации изображений. Она состоит в том, чтобы на основе признаков изображения присвоить ему одну из множества известных меток классов. Примерами такой задачи могут служить распознавание лиц камерами на улицах города в целях идентификации людей или распознавание образов беспилотными автомобилями, однако, сфера применения не ограничивается только ими.

В качестве обучающего набора был выбран набор Fashion-MNIST. Он состоит из черно-белых (всего 256 тонов) изображений одежды размером 28 x 28 пикселей. Всего в наборе 70000 экземпляров, где 60000 из них приходится на выборку для обучения, а 10000 - на тестовую выборку. Каждому изображению присвоен один из 10 возможных классов. Каждому классу соответствует 7000 изображений.

Предобработка набора данных включала в себя следующие шаги: преобразование изображения в трехмерный тензор вида (ширина, высота, количество каналов изображения), нормализация данных – преобразование значений из диапазона [0..255] в диапазон [0..1]; преобразование меток в формат OneHotEncoding. Каждый из этих шагов описан и разъяснен в текстовых полях ноутбука.

В работе рассмотрена классификация изображений сверточным автокодировщиком и полносвязной нейронной сетью. Решение задачи выполняется в два этапа. На первом этапе автокодировщик используется для предобучения нейронной сети, а на втором - к автокодировщику добавляется полносвязная нейронная сеть для классификации.

Автокодировщик - это модель глубокого обучения, состоящая из кодировщика и декодировщика, которая преобразует данные многомерного пространства в малоразмерное (в кодировку). Он работает, кодируя данные, независимо от их размера, в вектор (узкое место), а затем декодирует его, чтобы получить исходные данные. При обучении автокодировщика необходимо добиться как можно большей близости данных на выходе к исходным данным [3,4]. Благодаря этому на выходном слое кодировщика (в узком месте) определяются самые важные признаки объектов обучающей выборки. Размерность вектора признаков значительно меньше размерности входного изображения. Можно сказать, что автокодировщик обобщает информацию.

Теперь эти важные признаки нужно подать на полносвязную нейронную сеть и решить задачу классификации [3,4]. Для этого часть, соответствующая кодировщику, замораживается (ставится запрет на изменение его весовых коэффициентов), к ней присоединяются полносвязные слои и вновь созданная сеть обучается [3,4].

Кодировщик в данной работе имеет 4 сверточных блока. Каждый блок это сверточный слой, за которым следует слой нормализации. Слой выбора максимального значения из соседних (MaxPooling) используется после первого и второго блоков свертки. Декодировщик состоит из 3-х сверточных блоков. После второго и третьего блоков свертки используется слой передискретизации (UpSampling). Полносвязная нейронная сеть состоит из двух слоев – входного и выходного. В качестве активационной функции входного слоя используется relu, а активационной функции выходного слоя - softmax.

Классификация изображений с использованием для предобучения НС автокодировщика показала более хорошие результаты. В сравнении с полносвязной нейронной сетью, разница в точности на тестовой выборке составила ~4% (87% и 91%).

Предварительное обучение автокодировщика заняло 2,5 часа, в то время как полносвязную часть нейронной сети обучали 2 минуты.

Применение автокодировщиков для предобучения НС особенно важно в том случае, когда выборка содержит много примеров, но среди них размеченных ограниченное количество. В этом случае с помощью автокодировщика удастся получить в узком месте статистические характеристики полного набора данных (наиболее важные признаки объектов предметной области), затем уже на небольшом количестве размеченных данных обучить классификатор – полносвязную часть нейронной сети.

Библиографический список

1. Гиголаев А.В., Цуканова Н.И. Анализ семантической близости слов с помощью карт Кохонена.// Вестник Рязанского государственного радиотехнического университета. 2018. №2. С.85-91.
2. Бенгфорт Бенджамин, Билбро Ребекка, Охеда Тони. Прикладной анализ текстовых данных на Python. Машинное обучение и создание приложений обработки естественного языка.— Спб.: Питер, 2019. — 368с.
3. Жерон О. Прикладное машинное обучение с помощью Scikit-Learn и TensorFlow: концепции, инструменты и техники для создания интеллектуальных систем/пер. с англ. — СПб.:ООО “Альфа-книга”, 2018. — 688с.
4. Шитова К.Г., Цуканова Н.И. О применении глубоких нейронных сетей к оценке кредитоспособности предприятия. //Вестник Рязанского государственного радиотехнического университета. 2019. № 68. С.44-53

УДК 004.415.25; ГРНТИ 28.23.29

ИНТЕЛЛЕКТУАЛЬНОЕ ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ КЛАССИФИКАЦИИ ДАННЫХ С ИСПОЛЬЗОВАНИЕМ АЛГОРИТМА K-БЛИЖАЙШИХ СОСЕДЕЙ

А.В. Ломаков

*Рязанский государственный радиотехнический университет имени В.Ф. Уткина,
Российская Федерация, Рязань, lomakov97@mail.ru*

Аннотация. В данной работе рассматривается разработка интеллектуального программного обеспечения классификации данных с использованием алгоритма k -ближайших соседей. Приведены библиотеки языка программирования Python, использованные для реализации ПО и процесс разработки. Продемонстрирована работа разработанного интеллектуального программного обеспечения.

Ключевые слова: Python, PyQt, PyCharm, машинное обучение, классификация данных, KNN-классификатор, оценка качества классификации.

INTELLIGENT SOFTWARE DATA CLASSIFICATIONS USING THE ALGORITHM K-NEAREST NEIGHBORS

A.V. Lomakov

*Ryazan State Radio Engineering University named after V.F. Utkin,
Russia, Ryazan, lomakov97@mail.ru*

Abstract. This paper discusses of consider the development of intelligent software for data classification using the k -nearest neighbors algorithm. Libraries of the Python programming language used for software implementation and the development process are presented. The work of the developed intelligent software is demonstrated.

Keywords: Python, PyQt, PyCharm, machine learning, data classification, KNN-classifier, classification quality assessment.

Python – самый популярный из всех языков программирования (ЯП), применяемых в машинном обучении [1].

Для реализации интеллектуального программного обеспечения классификации данных (ИПОКД) было принято решение взять алгоритм классификации k -ближайших соседей (k -NearestNeighbors, kNN). Логика, лежащая в основе алгоритма kNN, является одной из самых простых из всех алгоритмов машинного обучения. Суть метода заключается в вычисле-

нии расстояния от новой точки данных до всех других точек учебного набора данных. Расстояние может рассчитываться по разным метрикам, таким, как евклидова, гауссовская, манхэттенская и т.д. Затем алгоритм выбирает k ближайших точек данных, где k может быть любым целым числом. Наконец, он присваивает точку данных классу, к которому принадлежит большинство из k точек данных.

При разработке ИПОКД были использованы следующие компоненты:

- 1) высокоуровневый ЯП общего назначения Python;
- 2) PyCharm - интегрированная среда разработки для ЯП Python. Предоставляет средства для анализа кода, графический отладчик, инструмент для запуска юнит-тестов;
- 3) PyQt5 - библиотека, которая позволяет использовать фреймворк Qt GUI (графический интерфейс пользователя) в Python. Является одним из наиболее часто используемых модулей для создания GUI приложений в Python;
- 4) PyQt5 Designer – библиотека, необходимая для визуального программирования интерфейса программного обеспечения;
- 5) csv – модуль, в котором реализованы классы для чтения и записи табличных данных в формате .csv;
- 6) pandas – программная библиотека на ЯП Python для обработки и анализа данных;
- 7) scikit-learn – бесплатная библиотека машинного обучения для ЯП Python;
- 8) библиотека pickle, реализующая двоичные протоколы для сериализации и десериализации структуры объектов Python.

В библиотеке scikit-learn находится модуль KNeighborsClassifier, который реализует алгоритм классификации данных kNN. Классификатор имеет параметры, представленные на официальном сайте разработчиков библиотеки [2] и метрики расчета расстояния [3], характеризующиеся определенной функцией.

Для применения любого алгоритма классификации также необходима оценка качества, чтобы оценить, насколько модель пригодна для использования в прогнозировании. Основные показатели качества, внедренные в ИПОКД, и их описание приведены в [4].

Перед написанием программного кода необходимо разработать интерфейс главной формы. Разработанный пользовательский интерфейс представлен на рисунке 1.

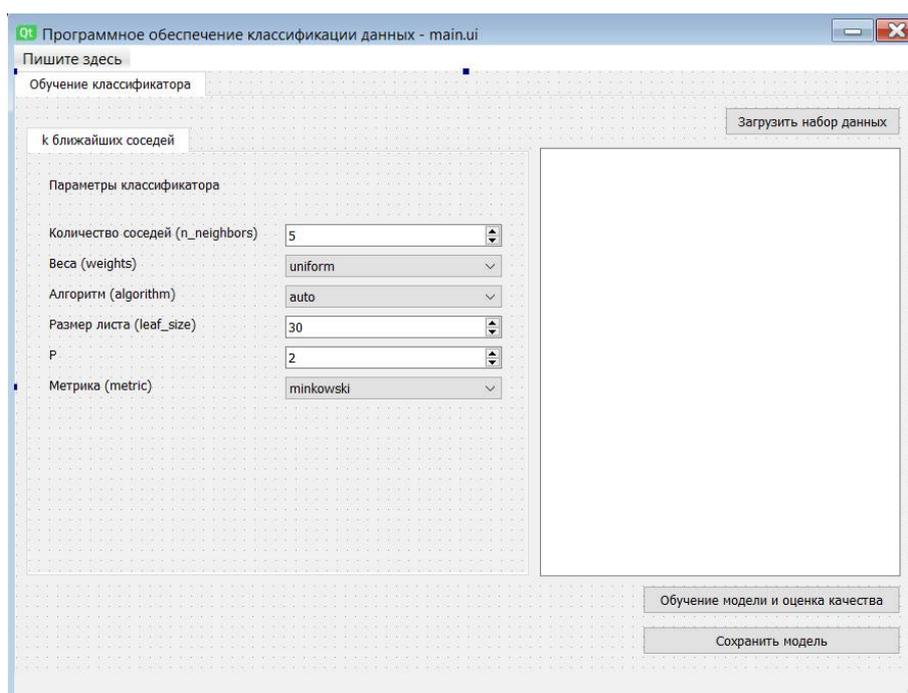


Рис. 1. Главная форма программного обеспечения

На вкладке «Обучение классификатора» располагаются следующие элементы:

- 1) вкладка с алгоритмом машинного обучения k -ближайших соседей, которая содержит основные параметры данного метода, причем выбор параметров строго определен в соответствии типами данных и/или допустимыми значениями;
- 2) текстовое поле, содержащее информацию о наборе данных: название столбца, число пропусков и тип элементов;
- 3) кнопка загрузки набора данных в формате .csv для обучения классификатора;
- 4) кнопка обучения модели и оценки качества классификации;
- 5) кнопка сохранения модели классификации для последующего прикладного использования.

Следующую форму, которую необходимо разработать, необходимо использовать для выбора целевого класса (по которому происходит классификация) пользователем после загрузки набора данных. Окно является модальным, поэтому выполнение действий в приложении запрещается, пока не будет выбран целевой класс. Разработанная форма представлена на рисунке 2.

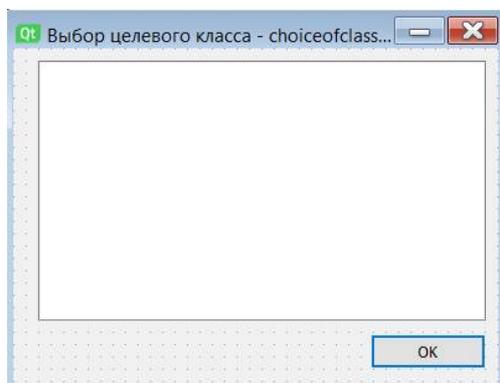


Рис. 2. Форма выбора целевого класса

Содержимое третьей формы, представленной на рисунке 3, необходимо для демонстрации пользователю качества выполненной классификации и оценки обученной модели.

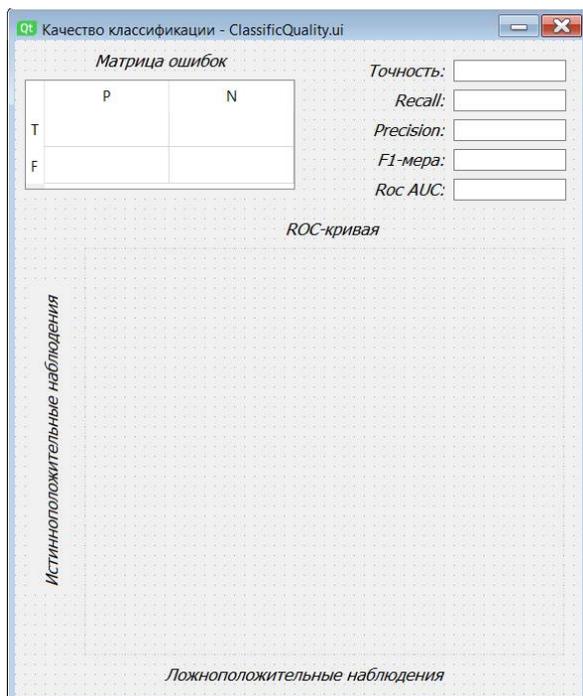


Рис. 3. Разработанная форма «Качество классификации»

Данная форма содержит информацию об основных метриках качества построенного классификатора, в том числе график ROC-кривой. Окно вызывается после нажатия кнопки «Обучение модели и оценка качества» на главной форме.

Главная форма, после загрузки набора данных и указания целевого класса выглядит следующим образом, представленным на рисунке 4.

На форме, возле кнопки «Загрузить набор данных», располагается информация о количестве записей, содержащихся в наборе. В текстовом поле отображена информация о столбцах в наборе, числе пропусков и типе элементов.

Число пропусков и тип элемента позволяют определить, насколько набор данных готов к обучению, т.к. в случае наличия пропусков или нечисловых типов данных обучение невозможно и необходимо проводить предварительную обработку данных.

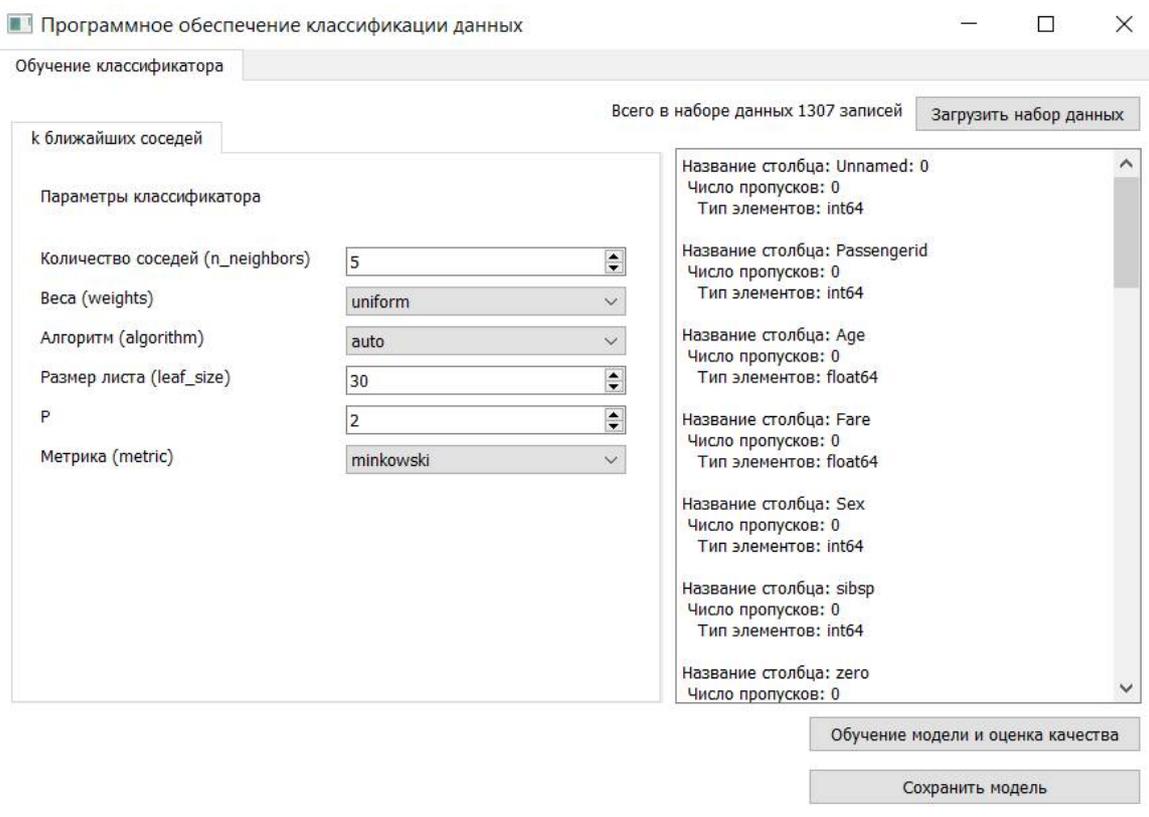


Рис.4. Содержимое главной формы после загрузки набора данных

После нажатия на кнопку «Обучение модели и оценка качества», происходит выполнение действий в следующей последовательности:

- 1) появление сообщения «Обучение классификатора...» около кнопки;
- 2) разделение набора данных на обучающую и тестовую выборку в соотношении 80/20;
- 3) обучение классификатора на обучающем наборе;
- 4) появление сообщения «Обучение завершено. Точность: *%» около кнопки;
- 5) проведение прогноза на тестовой выборке обученной модели классификации;
- 6) обращение к форме «Качество классификации»: расчет показателей качества, построение графика ROC-кривой, вывод вычисленных критериев в соответствующие поля на форме, показ формы.

После обучения классификатора можно сохранить модель, если ее качество соответствует поставленным целям, с помощью нажатия кнопки «Сохранить модель». Необходимо

выбрать место сохранения модели и задать название файла с типом .pkl (используется библиотека pickle) для дальнейшего использования построенного классификатора.

На примере набора данных Titanic (задача бинарной классификации – выжил ли человек после крушения Титаника или нет) [5] продемонстрирована работа разработанного ИПОКД, представленная на рисунках 5-7.

При обучении классификатора удалось добиться наилучшего качества классификации 81,68% со следующими параметрами:

- 1) количество соседей – 20;
- 2) веса – distance;
- 3) алгоритм – ball_tree;
- 4) параметр p – 2;
- 5) метрика расчета расстояния – euclidean;

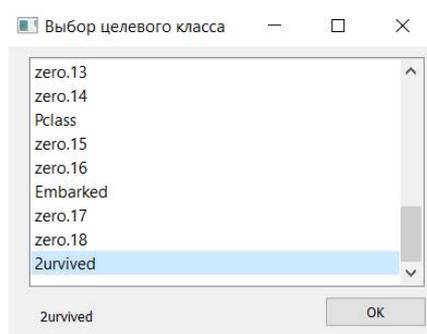


Рис. 5. Демонстрация окна выбора целевого класса

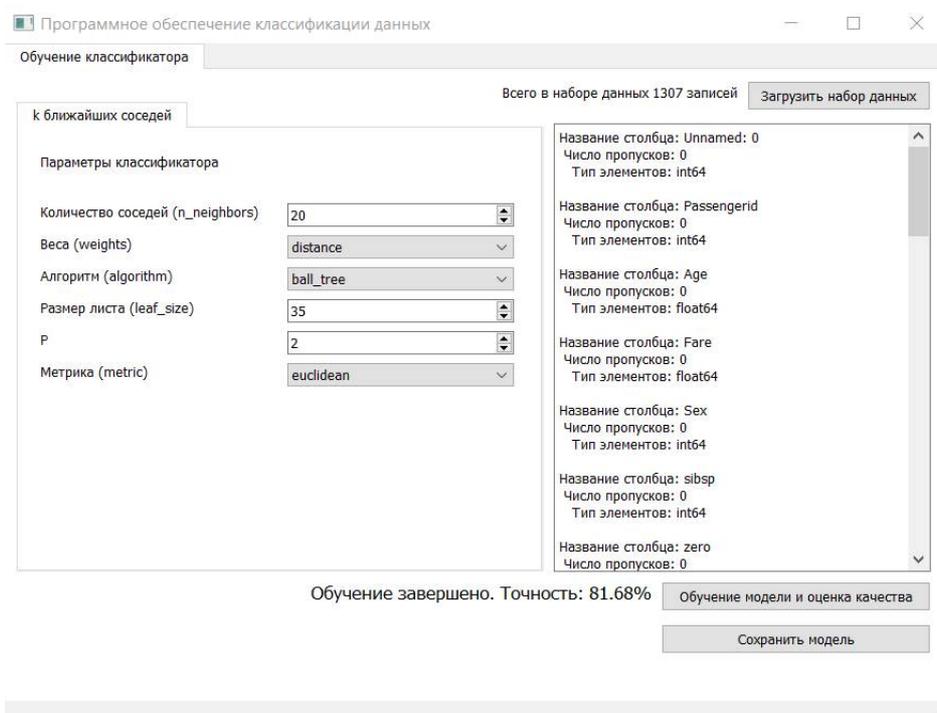


Рис. 6. Демонстрация главного окна ИПОКД

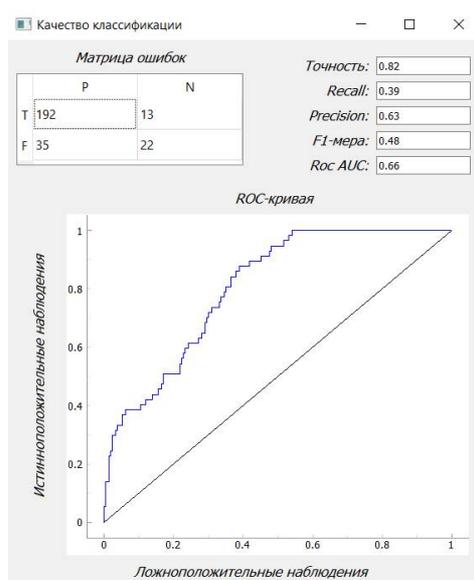


Рис. 7. Демонстрация окна качества классификации

В дальнейшем планируется разработать возможность использовать другие алгоритмы классификации в последующих вкладках, а также функционал для использования сохраненных моделей классификатора в прогнозировании.

Библиографический список

1. Стельмах С. Топ-5 языков для машинного обучения [Электронный ресурс]. itWeek, ООО «ИЗДАТЕЛЬСТВО СК ПРЕСС». – URL: <https://www.itweek.ru/ai/article/detail.php?ID=207784> (дата обращения: 15.02.2021).
2. sklearn.neighbors.KNeighborsClassifier [Электронный ресурс]. – URL: <https://scikit-learn.org/stable/modules/generated/sklearn.neighbors.KNeighborsClassifier.html> (дата обращения: 12.02.2021).
3. sklearn.neighbors.DistanceMetric [Электронный ресурс]. – URL: <https://scikit-learn.org/stable/modules/generated/sklearn.neighbors.DistanceMetric.html#sklearn.neighbors.DistanceMetric> (дата обращения: 12.02.2021).
4. Ломаков А.В. Оценка кредитоспособности заемщика в аналитических приложениях [Текст] // Современные технологии в науке и образовании, научное издание в 10 томах: сборник трудов, том 7. –Рязань.: BookJet, 2020. – С. 55-56.
5. Titanic - Machine Learning from Disaster [Электронный ресурс]. – URL: <https://www.kaggle.com/c/titanic> (дата обращения: 10.02.2021).

УДК 519.816; ГРНТИ 28.29

АНАЛИЗ МЕТОДОВ МНОГОКРИТЕРИАЛЬНОЙ ОПТИМИЗАЦИИ СЕТИ НА ОСНОВЕ ГРАФОВЫХ МОДЕЛЕЙ

А.С. Соловьёва

*Рязанский государственный радиотехнический университет имени В.Ф. Уткина,
Российская Федерация, Рязань, nastyasolovyova97@mail.ru*

Аннотация. В работе рассматривается использование графового метода для решения задач многокритериальной оптимизации сети. Приводятся алгоритмы нахождения оптимального пути, постановка и решение задачи многокритериальной оптимизации на графах.

Ключевые слова: графовое моделирование, многокритериальная оптимизация, оптоволоконная сеть, метод аддитивной свертки, метод Флойда, метод Дейкстры.

ANALYSIS OF METHODS FOR MULTI-CRITERIAL NETWORK OPTIMIZATION BASED ON GRAPH MODELS

A.S. Solovyova

*Ryazan State Radio Engineering University named after V.F. Utkin,
Russia, Ryazan, nastyasolovyova97@mail.ru*

The summary. The paper considers the use of the graph method for solving problems of multicriteria network optimization. Algorithms for finding the optimal path, formulation and solution of the problem of multicriteria optimization on graphs are presented.

Keywords: graph modeling, multicriteria optimization, fiber optic network, additive convolution method, Floyd's method, Dijkstra's method.

Графовое моделирование находит применение в самых разных областях человеческой деятельности, поскольку оно позволяет очень наглядно представить и эффективно описать физические, технические, экономические, биологические, социальные и другие системы.

При этом актуальной является задача многокритериальной оптимизации на графах, для решения которой можно воспользоваться синтезом графовых методов и методов теории принятия решений. В теории графов существует множество методов и алгоритмов решения задач однокритериальной оптимизации, а в теории принятия решений выделяют следующие методы: аддитивной и мультипликативной свертки, уступок, пересечения нечетких множеств, анализа иерархий и другие методы, которые относят к математическим методам многокритериальной оптимизации [1].

Подходы для решения задач многокритериальной оптимизации

Используется следующий подход для решения задач многокритериальной оптимизации на графах: строится обобщенный граф, а каждой его дуге и ребру приписывают вектор весов, размерность которого соответствует числу заданных критериев оптимизации. Значениями компонентов вектора являются оценки по критериям, которым лицо, принимающее решение назначает приоритеты. После этого значения оценок нормализуются и сворачиваются в соответствии с методом аддитивной свертки, впоследствии получают обобщенные веса дуг, ребер графа. Далее применяют однокритериальный оптимизационный алгоритм по отношению к графу с обобщенными весами и, таким образом, осуществляется решение многокритериальной оптимизационной задачи на графах на основе многих критериев.

Проанализируем пример с применением предложенного синтеза методов для решения задачи поиска произвольном взвешенном графе оптимальных по множеству заданных критериев путей.

В настоящее время достаточно часто встает задача многокритериальной оптимизации путей подключения клиентов к мультисервисным сетям на базе оптического волокна. Основными критериями выбора предпочтительных участков маршрута подключения являются: минимальное количество муфт, минимальное количество метров кабеля на пути прокладки;

максимальное количество занятых волокон; минимальное количество занятых волокон и минимальный коэффициент сложности пути.

Представить оптоволоконную сеть можно в виде графа, вершинами которого будут являться соединительные устройства (муфты), необходимые для соединения участков кабеля между собой. При этом в качестве точки, с которой передается сигнал (головная станция) может быть выбрана одна из вершин. В качестве ребер графа выступают отрезки кабеля, по которому оцифрованная информация передается до клиента.

Для того, чтобы решить подобную многокритериальную задачу, использовался метод нахождения кратчайших путей Дейкстры и метод аддитивной свертки критериев, а именно их интеграция [2].

Укрупненный алгоритм нахождения оптимального пути по многим критериям будет выглядеть следующим образом:

1. Составление перечня дуг (ребер) графа.
2. Добавление количества, наименований и направленности (max, min) критериев.
3. Внесение оценок дуг (ребер) по каждому из критериев.
4. Назначение весов для частных критериев, сумма критериев должна быть равняться единице.
5. Нормализация оценок по критериям.
 - 5.1. Поиск максимума из оценок по каждому критерию.
 - 5.2. Нахождение отношения значения оценок по критериям к значению максимума, найденному на шаге 5.1.
 - 5.3. Если критерий стремится к максимальному значению, то вычитают из единицы значение, найденной на этапе 5.2. В противном случае переход на шаг 6.
6. Нахождение обобщенных весов дуг (ребер) графа путем вычисления суммы произведений нормализованных значений оценок дуг (ребер) по критериям на веса частных критериев - аддитивная свертка частных критериев.
7. Оформление обобщенной матрицы смежности графа - перенос найденных на шаге 6 обобщенных весов дуг (ребер) графа в матрицу смежности графа.
8. Вычисление по обобщенной матрице смежности графа значений весов оптимальных путей на основе алгоритма Дейкстры.
9. Отображение оптимальных путей с помощью весов, найденных на предыдущем шаге.

Оптимальные пути, найденные в процессе будут являться искомыми обобщенными оптимальными путями в соответствии с многими критериями, для подключения клиентов к оптоволоконной сети.

Произведем рассмотрение данного подхода для решения задачи по нахождению обобщенных критических путей в модели проекта «Дуга-работа» (сетевая). Изначально находят максимальные по продолжительности пути в ациклическом ориентированном взвешенном графе, в котором продолжительности работ соответствуют веса дуг. Это будет являться критическими путями в сетевой модели. Продолжительность всего проекта в большой степени зависит от критических путей проекта, именно поэтому менеджерам проекта стоит уделять им наибольшее внимание. Обобщенные критические пути – это пути сетевой модели, которые соответствуют множеству критериев, включая критерий максимума продолжительности пути.

При подробном рассмотрении обобщенной сетевой модели проекта [3], в которой пути определяются в соответствии со следующими критериями: максимум продолжительности, максимум риска, минимум качества работ, максимум затрат, вытекает многокритериальная задача нахождения обобщенных критических путей сетевого графа. Длительность всего про-

екта, риски, качество менеджерам проекта позволяют отследить найденные в ходе решения обобщенные пути.

Рассмотрим решение данной задачи с помощью совокупности метода Флойда [4] нахождения кратчайших путей графа и метода аддитивной свертки критериев теории принятия решений с помощью следующего обобщенного алгоритма:

1. Назначить множество частных критериев оценок работ в проекте.
2. Назначить приоритеты для критериев.
3. Построить обобщенную сетевую модель проекта, добавить оценки по выбранным частным критериям (вектор весов) к дугам сетевого графа.
4. Выполнить нормализацию по частным критериям весовых значений дуг таким образом, что все оценки будут приведены к безразмерным величинам с учетом принципа максимума эффективности в интервале $[0,1]$.
5. Посредством метода аддитивной свертки найти обобщенные веса дуг сетевой модели путем свертки нормализованных значений весов дуг графа.
6. Выполнить преобразование сетевого графа – умножить на (-1) обобщенные веса дуг.
7. С помощью алгоритма Флойда по нахождению кратчайших путей в графе, вычислить оптимальные пути преобразованного сетевого графа.
8. Умножить все обобщенные веса дуг и вес найденных оптимальных путей на (-1) , тем самым вернувшись к исходному графу.

Таким образом, обобщенными критическими путями исходной сетевой модели в соответствии с заданным множеством частных критериев будут являться вычисленные оптимальные пути преобразованного графа и будут.

Постановка задачи

Пусть известен ориентированный ациклический граф $G=(V,E)$, описывающий существующую сеть (рисунок 1). V - множество вершин, которым соответствуют узловые элементы сети (коммутаторы). Ребро $(u,v) \in E$ между вершинами u и v задает канал связи и имеет положительную пропускную способность $c(u,v)$, показывающую, какое максимальное количество информации может быть передано по нему в единицу времени.

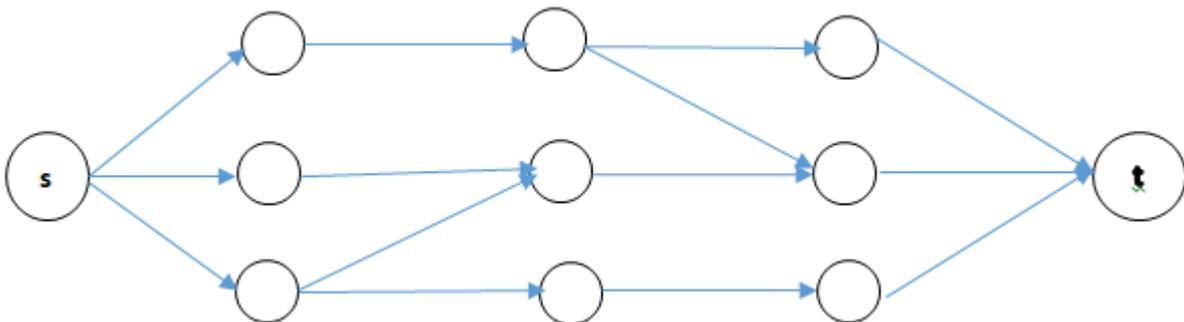


Рис. 1. Пример графа, описывающего сеть передачи данных

В графе выделяются две особые вершины: источник информации s и сток t . Из s ведут каналы от магистрального оператора, соединяющие сеть регионального с сетями доступа. В сток ведут ребра от коммутаторов, к которым подключены сети доступа.

Множество ребер E' ($E \cap E' = \emptyset$) описывает все каналы передачи данных, которые могут быть построены. Для каждого $(u,v) \in E'$ известна пропускная способность $c'(u,v)$ и стоимость строительства $p'(u,v)$. пронумеруем все ребра из множеств E' произвольным образом различными целыми числами от 1 до $|E'|$. Обозначим через e_i ребро с номером i из E' .

Процесс транспортировки данных в сети моделируется с помощью функции потока $f_G: V \times V \rightarrow \mathbb{R}$ (x - декартово произведение, V - множество вещественных чисел). Величина показывает, какое количество информации передается в единицу времени по каналу связи, соответствующего ребру между вершинами u и v . Функция f_G удовлетворяет трем условиям:

1. Ограниченность пропускной способности: $\forall u, v \in V : f_G(u, v) \leq c(u, v)$.
2. Антисимметричность: $\forall u, v \in V : f_G(u, v) = -f_G(v, u)$.
3. Сохранение потока: $\forall u \in V \setminus \{s, t\} : \sum_{v \in V} f_G(u, v) = 0$ [5].

Через $|f_G|$ обозначим величину потока в графе G , показывающую какое количество информации может быть передано из источника в сток за единицу времени. Тогда $|f_G|$ вычисляется следующим образом:

$$|f_G| = \sum_{v \in V} f_G(s, v) .$$

Через $|F_G|$ обозначим максимальную величину потока в графе G среди всех возможных значений f_G .

Планом проектирования сети передачи данных назовем множество ребер $E^* \subseteq E$. Оно соответствует каналам связи, которые будут построены. При принятии решения о реализации плана проектирования исходная сеть трансформируется в сеть, описываемую графом $G^* = (V, E \cup E^*)$.

Через $Q_1(E^*)$ обозначим стоимость реализации плана проектирования E^* :

$$Q_1(E^*) = \sum_{(u,v) \in E^*} p'(u, v) .$$

Через $Q_2(E^*)$ обозначим величину максимальной пропускной способности сети, описываемой графом G^* , взятую со знаком минус: $Q_2(E^*) = -|F_{G^*}|$.

Тогда задача оптимизации формулируется следующим образом: По заданным ациклическим ориентированным графам $G = (V, E)$ и множеству ребер E' ($E \cap E' = \emptyset$), матрицам пропускных способностей $c(u, v)$ и $c'(u, v)$ и матрице стоимости ребер $p'(u, v)$ найти множество Парето-оптимальных решений задачи $\min_{E^*} (Q_1(E^*))$, $\min_{E^*} (Q_2(E^*))$.

Выбор этих критериев оптимизации обусловлен следующими причинами. Стоимость является ключевым фактором при реализации любого проекта. Так как наибольший вклад вносят затраты, связанные со строительством канала, то другие издержки, как, например, стоимость обслуживания оборудования, в расчет не берутся. Стоит отметить двойственность понятия стоимости: с одной стороны это ограничение в виде бюджета, выделенного на реализацию модернизации, а с другой - критерий оптимизации. Зная максимальную пропускную способность модифицированной сети, можно оценить, какое наибольшее количество конечных пользователей может ее использовать.

Таким образом, совокупность графовых методов однокритериальной оптимизации и многокритериальных методов теории принятия решений позволяет эффективно решать задачи многокритериальной оптимизации на графах.

Библиографический список

1. Андрейчиков, А.В. Анализ, синтез и планирование решений в экономике/ А.В. Андрейчиков, О.Н. Андрейчикова.- М.: Финансы и статистика, 2000.-363с.
2. Арсеньев, Ю.Н. Принятие решений. Интеллектуальные интегрированные системы: Учебное пособие для вузов/ Ю.Н. Арсеньев, С.И. Шелобаев, Т. Ю. Давыдова.-М.: ЮНИТИ ДАНА, 2006,-447 с.
3. Бережная, Е.В. Математические методы моделирования экономических систем: Учебное пособие/ Е.В. Бережная, В.И. Березной. - М.: Финансы и статистика, 2001.-368 с.
4. Мазур, И.И. Управление проектами / И.И Мазур., В.Д. Шапиро., Н.Г. Ольдерогге.- М.: Омега-Л, 2004.-664 с.
5. Харари, Ф. Теория графов. Пер. с англ./ Ф. Харари - Едиториал УРСС, 2006.- 501 с.

УДК 004.032.26; ГРНТИ 28.23.37

ОБОСНОВАНИЕ АКТУАЛЬНОСТИ РАЗРАБОТКИ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ КЛАССИФИКАЦИИ ЖАНРОВ

И.В. Васильев

*Рязанский государственный радиотехнический университет имени В.Ф. Уткина,
Российская Федерация, Рязань, ilya.vasilev.vladislavovich@gmail.com*

Аннотация. В работе рассматриваются существующие методы определения жанров аудио треков с помощью нейронных сетей. Приводятся их основные особенности, достоинства и недостатки. Также рассматриваются основные особенности работы с аудиофайлами.

Ключевые слова: аудиофайл, жанр, классификация, нейронные сети.

JUSTIFICATION OF THE RELEVANCE OF DEVELOPMENT SOFTWARE FOR GENRE CLASSIFICATION

I.V. Vasilev

*Ryazan State Radio Engineering University named after V.F. Utkin,
Russia, Ryazan, ilya.vasilev.vladislavovich@gmail.com*

The summary. The paper discusses exist classification music genre methods using neural networks. Given main features, advantages and disadvantages. The main features of working with audio files are also considers.

Keywords: audio, genre, classification, neural network.

Классификация музыкальных жанров чаще всего выполняется с помощью фильтрации пользовательских предпочтений, но более перспективным и менее разработанным является направление применения для данной процедуры машинного обучения [1].

Рассмотрим теорию обработки звука [2]. Отдельная нота – это гармонический сигнал с определённой частотой и амплитудой. Как правило, мелодию, речь или иной звуковой сигнал можно представить как сумму гармонических сигналов. Шумом в таком случае называют слагаемые, соответствующие любым нежелательным звукам. Преобразование Фурье позволяет разложить исходный сигнал на гармонические составляющие, то есть на частоты.

На практике частоты не используются, так как амплитудно-частотная характеристика человеческого уха не является прямой. Для измерения высоты звука, воспринимаемой человеческим ухом, используется специальная психофизическая единица высоты звука – мел. Название единицы происходит от слова мелодия.

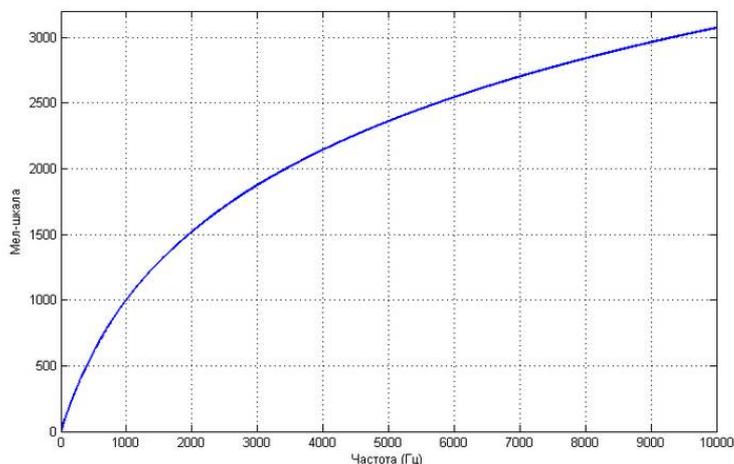


Рис. 1. График зависимости высоты звука в мелах от частоты колебаний (для чистого тона)

После разложения аудиофайла на частоты в виде спектрограмм данные приводятся в мел-спектрограммы – обычной спектрограммы, где частота выражена не в Гц, а в мелах.

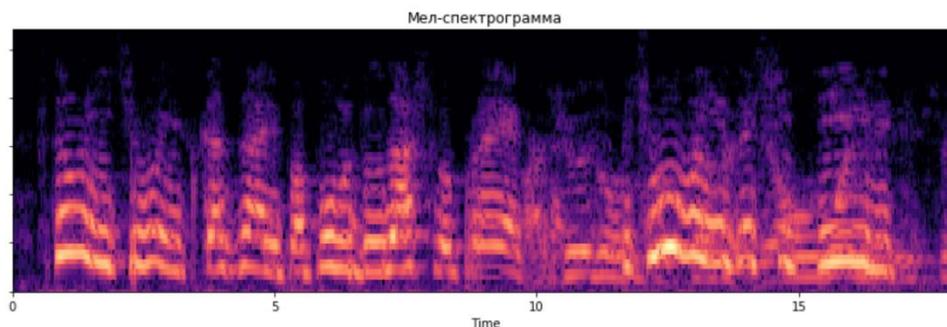


Рис. 2. Пример мел-спектрограммы

Были рассмотрены и проанализированы работы, рассматривающие данную проблему и предлагающие пути её решения.

В [3] была предложена техника распознавания музыкальных жанров, основывающаяся на соответствии последовательности аккордов определённым жанрам. Авторы предлагают вначале промаркировать аккорды, используя бинарную последовательность, – «1» используется для обозначения звука, который идёт вверх в музыкальной последовательности, а «0» используется для звука, который идёт вниз. Там, где звук не изменяется, звук маркируется «0».

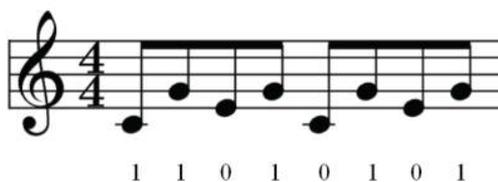


Рис. 3. Пример маркировки нот бинарным способом

Далее, авторы предлагают сопоставление жанров с использованием метода последующего сопоставления алгоритмом `k-windowsSubsequenceMatching` для поиска соответствия подпоследовательностей в музыкальной последовательности. В заключении процесса классификации было предложено классифицировать методом дерева решений. Из недостатков данной работы можно выделить слишком малое количество жанров – всего 6 жанров.

В [1] автор сравнивает методы и техники классификации жанров. Происходит сравнение методов предварительной обработки данных, анализ разницы различных длин выборки, количества жанров, выбора набора данных и структуры нейронной сети.

В анализе метода предварительной обработки данных автор выбирает между MFCC (Mel Frequency Cepstral Coefficients), FCC (Fast Fourier Transform) и мел-спектрограммой. Был выбран самый лучший метод мел-спектрограммы, так как у него самая высокая точность – 48%. Далее, автор сравнивает длительность образцов обучения, и приходит к тому, что наилучшая длительность образца обучения – пять секунд.

В следующей части работы произведено сравнение моделей нейронных сетей между сверточными и рекуррентными. Оказалось, что свёрточные нейронные сети значительно повышают точность модели.

Далее указывается, что свёрточные нейронные сети ограничивают количество жанров до 7, поэтому в рекуррентной модели нейросети был произведён анализ, какое количество жанров оптимально, чтобы модель выдавала более точные результаты – 10 или 14. При 10 жанрах модель показывала точность 32%, а при 14 точность составляла 23%.

Далее происходит сравнение и анализ наборов данных GTZAN и MSD. В этой части сравнение происходит по обоим типам нейронных сетей и наборов данных, различному количеству жанров. Результаты приведены в таблице 1.

Таблица 1. Сравнение точности различных моделей и выборки, используя мел-спектрограмму

Тип нейронной сети	RNN	RNN	CNN	CNN
Набор данных	GTZAN	MSD	GTZAN	MSD
Количество жанров	10	10	10	7
Точность	50%; 33%	40%; 32%	95%; 49%	85%; 68%

В итоге происходит сравнение тестовой выборки – выбрана тестовая выборка GTZAN. Итоговая наилучшая точность модели является 59%.

В конце автор производит сравнение точности классификации жанров на графиках.

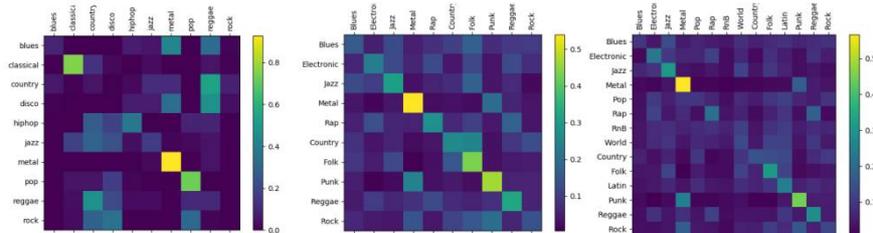


Рис. 4. Сравнение графиков точности классификации жанров: рекуррентная нейронная сеть с использованием выборки GTZAN, MSD на 10 и на 14 жанров

Таким образом, автор приводит к выводу, что наиболее эффективной архитектурой модели является 3 секундная выборка набора данных GTZAN, мел-спектрограмма и свёрточная нейронная сеть. Данная модель позволяет классифицировать с точностью 59%.

В [4] используется довольно простой способ классификации аудиофайлов с помощью глубокого обучения нейронной сети с прямой связью. В данной работе авторы используют язык программирования R и библиотеку 'H2O'. Выборка разделена на 3 части: обучающая выборка 60%, валидационная 20%, и тестовая 20%.

Авторы используют классификацию на 4 жанра, что, естественно, облегчает работу, но, в то же время, очень смазывает итоговую точность, так как нет данных, как поведёт себя модель, к примеру, при большем количестве жанров. В итоге, можно сделать вывод, что данную модель следует обучить ещё большему количеству жанров.

Таким образом, можно сделать вывод о высокой актуальности проблемы разработки программного обеспечения, реализующего алгоритмы классификации аудио треков по жанрам, необходимости дальнейших исследований в этой предметной области и нахождения моделей с более высокой точностью.

Библиографический список

1. Jingqing Yang Music Genre Classification with Neural Networks: An Examination of Several Impactful Variables. – Trinity University, Computer Science Honors Theses, 2018 – 35 pages. [https://digitalcommons.trinity.edu/cgi/viewcontent.cgi?article=1044&context=compsci_honors, 20.02.2021]
2. ВолковецА.И., Создание и обработка звука при разработке интерактивных приложений. Конспект лекций для студентов специальности «Информационные системы и технологии (в игровой индустрии)» – Минск, Белорусский государственный университет информатики и радиоэлектроники, 2018. [https://www.bsuir.by/m/12_100229_1_127400.pdf, 19.02.2021]
3. Nasridinov A., Young-Ho ParkA Study on Music Genre Recognition and Classification Techniques. – International Journal of Multimedia and Ubiquitous Engineering, 2014–pp 31-41.[https://www.researchgate.net/publication/273903999_A_Study_on_Music_Genre_Recognition_and_Classification_Techniques, 20.02.2021]
4. JawaherlalnehruG., JothilakshmiS. Music Genre Classification using Deep Neural Network. - International Journal of Scientific Research in Science, Engineering and Technology, Annamalai University, 2018 - pp 935-940. [http://ijsrset.com/paper/4076.pdf, 21.02.2021]

УДК 004.415.2.043; ГРНТИ 20.23.21

ИНТЕРФЕЙС РАЗРАБОТАННОЙ ИС ДЛЯ УПРАВЛЕНИЯ СЕБЕСТОИМОСТЬЮ ПРОДУКЦИИ НА ПРЕДПРИЯТИИ

М.Э. Мерзлякова

*Рязанский государственный радиотехнический университет имени В.Ф. Уткина,
Российская Федерация, Рязань, mmerzlyak@gmail.com*

Аннотация. В данной работе рассматривается разработанный интерфейс информационной системы для управления себестоимостью продукции на предприятии. Продемонстрировано, что данная информационная система предназначена для автоматизированного анализа статей затрат калькуляции и последующего расчета себестоимости продукции в расчете на единицу продукции.

Ключевые слова: ИС, информационная система, интерфейс ИС, себестоимость.

INTERFACE OF THE DEVELOPED INFORMATION SYSTEM FOR PRODUCT COST MANAGEMENT IN THE ENTERPRISE

M.E. Merzlyakova

*Ryazan State Radio Engineering University named after V.F. Utkin,
Russia, Ryazan, mmerzlyak@gmail.com*

Abstract. This paper discusses of consider the development interface of the information system for managing the cost of production at the enterprise. It has been demonstrated that this information system is intended for the automated analysis of cost items of the calculation and the subsequent calculation of the cost of production per unit of production.

Keywords: IC, information system, information system interface, cost.

ИС – это система, которая построена на объектно-реляционном отображении, программист в ней работает не напрямую с реляционным представлением, а с объектами. Это значительно снижает время на разработку программного обеспечения, что является несомненным плюсом. Программист создает прикладные объекты конфигурации (справочники, документы, константы, перечисления, регистры и прочее). Справочники, константы и перечисления содержат все необходимые данные, которые будут использованы далее при работе с документами. Для документов можно создавать печатные формы с помощью встроенного конструктора, кроме того, имеется возможность редактирования макета печатной формы для приведения его к требуемому виду. Сведения из документов сохраняются в регистрах различного вида. Платформа ИС позволяет расширять функционал созданной конфигурации или внедрять его в другую конфигурацию [1].

На данной платформе ИС первичными понятиями выступают объекты мета-данных: константы, справочники, документы и регистры сведений, совокупность которых формирует внутреннее строение базы данных и структуру информации, размещенной в ней [2].

Разрабатываемая информационная система предназначена для автоматизированного анализа статей затрат калькуляции и последующего расчета себестоимости продукции в расчете на единицу продукции. Система позволяет добавлять все необходимые характеристики, дает возможность оценки стратегической важности представленных проектов, а также относительной важности критериев оценки. Система осуществляет расчет интегрального показателя, на основании которого формирует упорядоченный список проектов и распределяет финансовые средства в соответствии с ним.

Таким образом, для разработки информационной системы использовалась система ИС, т.к. данная система имеет необходимый инструментарий для решения задач по управлению себестоимостью продукции, т.е. нацелена на работу с бухгалтерией предприятия и раздельным учетом.

Для выполнения входа в систему для начала необходимо авторизоваться. Для этого в поле «Пользователь» необходимо выбрать тип пользователя в соответствии со своим стату-

сом, а также ввести пароль в соответствующем поле. Окно с доступом в информационную систему представлено на рис.1.

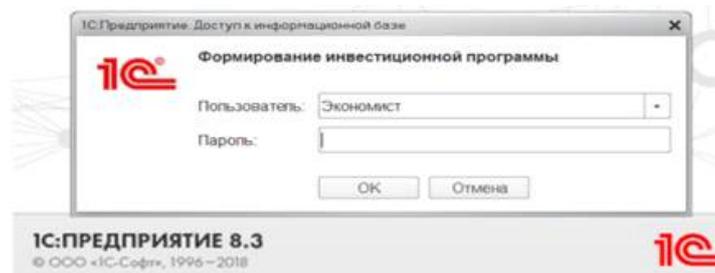


Рис. 1. Вход в информационную систему

В начале работы экономист заполняет константы в виде критериев оценки. Для этого пользователю необходимо перейти на «Сервис» — «Вес критерия доставка» и в открывшемся окне (см. рис. 2) написать удельный вес данного критерия. После подтверждения ввода и нажатия кнопки «Записать и закрыть» результатом действия впоследствии станет расчет интегрального показателя у поставщиков при выборе среди них наилучшего.

В окнах «Вес критерия качество», «Вес критерия риски» и «Вес критерия цена» ввести данные соответственно.

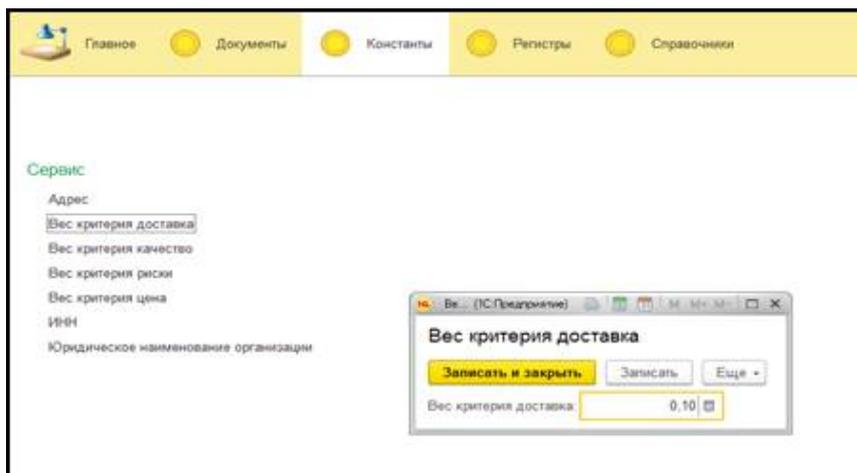


Рис. 2. Окно заполнения константы системы «Вес критерия доставка»

Следующим этапом является размещение наименования материалов в справочнике «Материалы» и их количества на единицу изделия в регистре «Количество материалов» соответственно (см. рис. 3 – 4). Нормы расхода материала на единицу будут внесены в периодический регистр сведений, учитывая тот факт, что со временем эти показатели могут меняться.

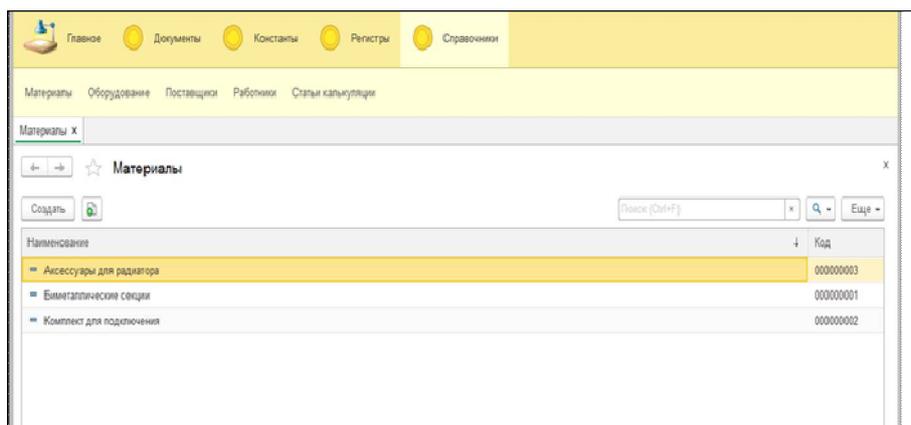


Рис. 3. Добавление наименования материалов в систему

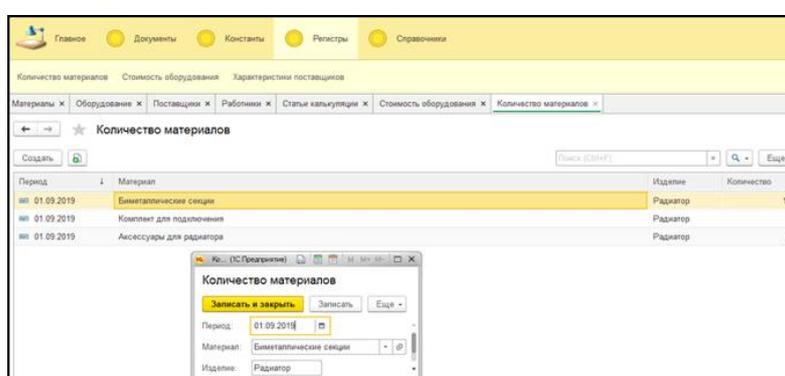


Рис. 4. Добавление расхода материалов на 1 ед. изделия в систему

Для того, чтобы добавить изделие и материалы, пользователю необходимо нажать на кнопку «Создать», вследствие чего откроется окно и будут доступны поля для заполнения (см. рис. 5).

В данном окне необходимо добавить материал из соответствующего справочника «Материалы». Поле «Изделие» и «Количество» заполняется вручную. После внесения информации необходимо подтвердить действия, нажав кнопку «Записать и закрыть».

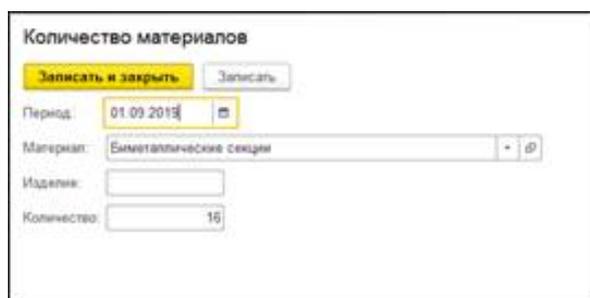


Рис. 5. Информация о материале, записанная в системе

В справочнике «Поставщики», представленном на рисунке 6, записываются и группируются поставщики по поставляемым комплектующим.

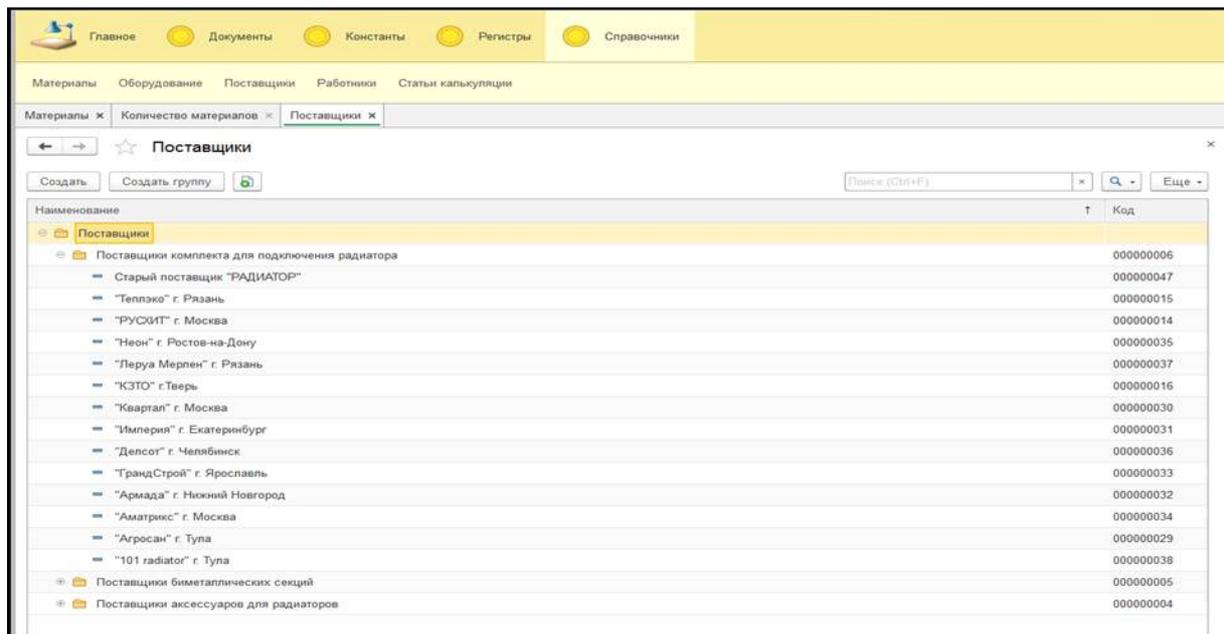


Рис. 6. Добавление поставщиков в систему

После заполнения справочника «Поставщики» экономист переходит к документу «Оценка поставщиков» и заполняет соответствующие поля «цена», «доставка», «качество» и «риски» (см. рис. 7, 8).

Характеристики поставщиков

Записать и закрыть Записать

Период:

Поставщик: "Агросан" г. Тула

Материал: Аксессуары

Цена: 380,00

Доставка: 9

Качество: 7

Риски: 20 %

Рис. 7. Добавление характеристики поставщиков в систему

Период	Поставщик	Материал	Цена	Доставка	Качество	Риски
01.12.2019	"Леруа Мерлен" г. Рязань	Аксессуары для радиатора	350 рублей	9	7	10 %
01.12.2019	"Агросан" г. Тула	Аксессуары для радиатора	380 рублей	9	7	20 %
01.12.2019	"101 radiator" г. Тула	Комплект для подключения	183 рублей	6	4	24 %
01.12.2019	"Леруа Мерлен" г. Рязань	Комплект для подключения	190 рублей	9	8	10 %
01.12.2019	"Делсот" г. Челябинск	Комплект для подключения	206 рублей	6	6	43 %
01.12.2019	"Неон" г. Ростов-на-Дону	Комплект для подключения	199 рублей	4	5	15 %
01.12.2019	"Аматрикс" г. Москва	Комплект для подключения	192 рублей	4	10	20 %
01.12.2019	"ГрандСтрой" г. Ярославль	Комплект для подключения	187 рублей	6	9	45 %
01.12.2019	"Армада" г. Нижний Новгород	Комплект для подключения	232 рублей	6	4	70 %
01.12.2019	"Империю" г. Екатеринбург	Комплект для подключения	220 рублей	5	7	32 %
01.12.2019	"Кварган" г. Москва	Комплект для подключения	196 рублей	7	9	35 %
01.12.2019	"Агросан" г. Тула	Комплект для подключения	185 рублей	7	6	56 %
01.12.2019	"Otagint" г. Москва	Биметаллические секции	157 рублей	8	7	60 %
01.12.2019	"АгроСан" г. Тула	Биметаллические секции	215 рублей	6	7	50 %
01.12.2019	"Аматрикс" г. Москва	Биметаллические секции	212 рублей	4	10	20 %
01.12.2019	"Новый век" г. Москва	Биметаллические секции	230 рублей	8	9	10 %
01.12.2019	"Евро Сервис" г. Тула	Биметаллические секции	210 рублей	3	4	45 %
01.12.2019	"Неон" г. Ростов-на-Дону	Биметаллические секции	185 рублей	4	5	15 %
01.12.2019	"Армада" г. Нижний Новгород	Биметаллические секции	167 рублей	6	4	70 %
01.12.2019	"Волка" г. Сочи	Биметаллические секции	190 рублей	9	10	30 %

Рис. 8. Добавление характеристики поставщиков в систему

Так как каждый критерий оценки имеет свою степень значимости для предприятия, зафиксированную в константах программы, после присвоения ранга каждой характеристике поставщика происходит расчет интегрального показателя, показывающего иерархию от наиболее выгодного поставщика к менее выгодному.

Далее пользователь переходит в документ «Анализ материалов» (см. рис. 9), где для начала происходит выбор материала, цену которого необходимо рассчитать, выбор в графу «Первый поставщик» поставщика с наивысшим интегральным показателем, а в графу «Второй поставщик» соответствующие цены старого поставщика предприятия.

N	Материал	Первый поставщик	Второй поставщик	Итог оптимальный	Итог фактический	Вывод
1	Биметаллические секции	"Волка" г. Сочи	Старый поставщик "СЕКЦИОН"	3 040	4 680	Превышение по цене
2	Комплект для подключения	"Леруа Мерлен" г. Рязань	Старый поставщик "РАДИАТ..."	190	235	Превышение по цене
3	Аксессуары	"Тайм" г. Подольск	Старый поставщик "АКСЕСС..."	1 300	1 680	Превышение по цене

Рис. 9. Создание документа «Анализ материалов» в информационной системе

Для того, чтобы проанализировать изменение показателя себестоимости продукции при выборе нового поставщика, необходимо перейти в документ себестоимость продукции (см. рис. 10), где автоматически будут рассчитываться графы «Цены до внедрения предло-

жения», «Цены после внедрения предложения», а также экономия на единицу изделия будет рассчитываться автоматически.

№	Статья	Цена до внедрения предложения	Цена после внедрения предложения	Экономия на одну единицу
1	Сырье и материалы	6 603,00	4 530	2 073
2	Транспортно-заготовительные расходы	1 650,75	1 132,5	518
3	З/л основных производственных рабочих	610,78	610,78	0
4	Отчисления во внебюджетные фонды	183,23	183,23	0
5	Амортизация	30,69	30,69	0
6	Накладные расходы	1 221,56	1 221,56	0

Рис. 10. Расчет себестоимости продукции и экономии на единицу при выборе нового поставщика

Разработанная информационная система предназначена для анализа любой статьи затрат себестоимости. Рассмотрим на примере анализа статьи затрат «Зарботная плата». В справочник «Зарботная плата» (см. рис. 11) заносятся данные с наименованием работника, его разрядом, трудоемкостью выполнения работы и часовой тарифной ставкой. После того, как информация внесена, необходимо нажать на кнопку «Записать и закрыть».

Рис. 11. Добавление работника в систему

После заполнения справочника «Работники» пользователь может перейти к документу «Зарботная плата» и начать заполнение полей.

1) Поле «Работник», «Разряд рабочего», «Трудоемкость» - заполняется автоматически.

2) Показатель поля «Фактические часы» вводится вручную.

После заполнения вышеуказанных полей происходит сравнение трудоемкости фактической с оптимальной и вывод полученного расчета.

Таким образом, в данной статье была рассмотрена разработанная на платформе 1С информационная система для поддержки механизма управления себестоимостью продукции

на предприятии. Разработанная система позволяет проводить анализ статей затрат, а в следствии рассчитывать и саму себестоимость продукции, а также производить расчет интегрального показателя, необходимого для дальнейшего выбора оптимального поставщика для предприятия. Данная ИС позволяет выявлять проблемные места при анализе статей затрат себестоимости продукции и может контролировать не только рассматриваемую статью «Сырье и материалы», но и другие. Было проведено тестирование данной ИС и сделан вывод о том, что система работоспособна и полностью отвечает всем предъявленным к ней требованиям.

Библиографический список

1. wiseadvice. Разработка на 1С [Электронный ресурс]. – URL: <https://wiseadvice-it.ru/okompanii/blog/articles/razrabotka-na-1s-predpriyatiya-8-3/> (дата обращения: 05.02.2021).
2. 1С ИТС. Объекты конфигурации [Электронный ресурс]. – URL: <https://its.1c.ru/db/v837doc/bookmark/dev/TI000000167> (дата обращения: 05.02.2021).

УДК 004.852; ГРНТИ 20.19.27

ОПРЕДЕЛЕНИЕ ТОНАЛЬНОСТИ ТЕКСТА НА ОСНОВЕ РЕКУРРЕНТНЫХ ИСКУССТВЕННЫХ НЕЙРОННЫХ СЕТЕЙ КАК ОДИН ИЗ МЕТОДОВ, ИСПОЛЪЗУЕМЫХ ПРИ ВЫЯВЛЕНИИ ПРИЗНАКОВ МОШЕННИЧЕСТВА

М.А. Лащилин

*Рязанский государственный радиотехнический университет имени В.Ф. Уткина,
Российская Федерация, Рязань, mlashchilin@yandex.ru*

Аннотация. В работе рассматривается подход к определению тональности текста с использованием рекуррентных нейронных сетей, как один из методов, который может быть использован при выявлении признаков мошенничества. Приводятся основные особенности такого анализа, достоинства и недостатки подхода, связанного с рекуррентными нейросетями.

Ключевые слова: машинное обучение, рекуррентные нейронные сети, определение тональности текста.

SENTIMENT ANALYSIS OF THE TEXT ON THE BASIS OF RECURRENT ARTIFICIAL NEURAL NETWORKS AS ONE OF THE METHODS USED IN IDENTIFYING FRAUD SIGNS

M.A. Lashchilin

*Ryazan State Radio Engineering University named after V.F. Utkin,
Russia, Ryazan, mlashchilin@yandex.ru*

The summary. The paper discusses the approach in sentiment analysis of the text using recurrent neural networks as one of the method, which may be used in identifying fraud signs. Given main features, advantages and disadvantages of this analysis with recurrent neural networks.

Keywords: machine learning, recurrent neural networks, sentiment analysis of the text.

Введение

Одной из важнейших задач банковской сферы является выявление признаков мошенничества. Следует сказать, что под понятием «выявление мошенничества» возможно широкое толкование, начиная от поиска признаков, указывающих с определённой вероятностью на обман в финансовых операциях [1], проводимых клиентом, заканчивая комплексным мониторингом на основе сложных программных систем. Одной из важнейших проблем выявления признаков мошенничества является их обнаружение в текстах на естественном языке. Например, это может быть диалог в виде переписки между человеком, представляющимся

сотрудником банка, и клиентом. Тогда перед программой, решающей данную задачу, стоит цель определить вероятность того, что данный текст является попыткой мошенничества по отношению к клиенту. Данная задача является комплексной и включает в себя определенные подзадачи.

Одной из таких задач, которая позволяет повысить точность выявления признаков мошенничества в тексте, является определение тональности текста. Задача определения тональности в рассматриваемом контексте, являющаяся объектом данного исследования, позволяет определить тон, который потенциально соответствует тону человека, пытающегося осуществить мошенническое действие по отношению к тому, к кому был направлен анализируемый текст. Исходя из того, что на данный момент увеличивается количество попыток мошенничества [2], можно говорить, что рассматриваемая задача актуальна.

Цель данной работы заключается в рассмотрении метода определения тональности текста на основе рекуррентных искусственных нейронных сетей, определении его достоинств и недостатков для выяснения целесообразности дальнейшей реализации в виде программного модуля системы по выявлению признаков мошенничества.

Постановка задачи

Задача определения тональности текста представляет собой задачу компьютерной лингвистики, которая заключается в определении эмоциональной окраски текста и в выявлении эмоциональной оценки автора текста к объектам в нём [3].

В общем виде рассматриваемая задача относится к задачам классификации текстов. Классами, как правило, выступают тональные оценки. Одним из возможных случаев оценок может быть следующая шкала: позитивная, нейтральная или негативная оценка. Допускается более сложный случай с промежуточными оценками, также называемый как классификация по многополосной шкале. В данной статье далее будет рассматриваться первый вариант.

Полученный на выходе результат предполагается передавать на вход системы выявления признаков мошенничества в тексте, как один из параметров, для дальнейшего определения вероятности мошенничества.

Для решения задачи определения тональности текста, существует несколько типов методов: методы, основанные на правилах; методы, с использованием словарей и методы машинного обучения с учителем или без него. Одними из перспективных на данный момент являются языковые модели машинного обучения с учителем, которые могут решать различные задачи [4], поскольку данный тип методов позволяет получить достаточную точность и не требует такой трудоёмкой работы, как составление правил или словарей для первых двух методов. Однако одним из недостатков данного метода является необходимость наличия достаточной обучающей выборки для модели.

Рекуррентные нейронные сети

Методы машинного обучения с учителем предполагают обработку поступающей на вход информации с использованием искусственных нейронных сетей. Однако с обработкой последовательностей нейросети стандартной архитектуры прямого распространения справляются недостаточно эффективно. Для таких задач используют рекуррентные нейронные сети.

Рекуррентные нейронные сети представляют собой нейронные сети, в которых связи между элементами образуют направленную последовательность.

Преимущества рекуррентных нейронных сетей, образующих циклы обработки сигналов, в том, что они эффективнее в обработке последовательностей, так как нейросети данного типа имеют возможность запоминать информацию. За счёт этого увеличивается точность

работы нейросети, поскольку при обработке естественных языков важен порядок слов, которые подаются на вход, предварительно преобразованных к векторному представлению.

На рисунке 1 представлена обобщённая архитектура слоя рекуррентной нейросети, развёрнутая во времени.

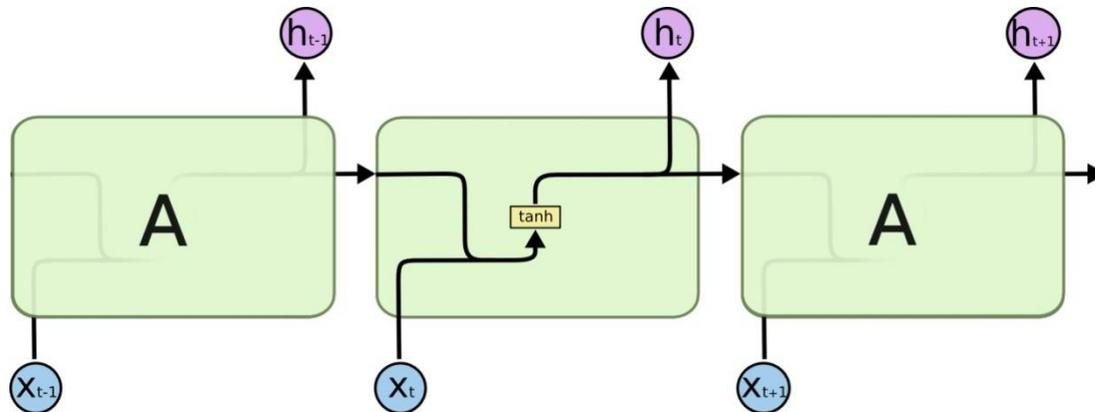


Рис. 1. Архитектура слоя рекуррентной нейронной сети

Однако классическая архитектура рекуррентной нейронной сети не позволяет запоминать достаточно длинные последовательности. Для того чтобы обойти данное ограничение используют модифицированные архитектуры рекуррентных нейронных сетей: LSTM (Long Short-Term Memory) [5] и GRU (Gated Recurrent Units) [6]. На рисунке 2 представлена архитектура сети LSTM. Архитектура GRU детально в данной статье не рассматривается, так как она лучше подходит для решения задач моделирования музыкальных и речевых сигналов [6].

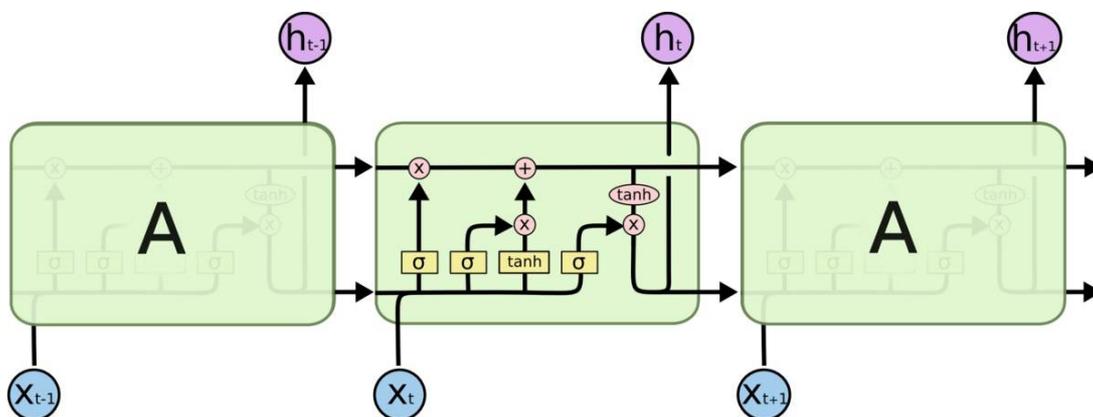


Рис. 2. Архитектура сети LSTM

В LSTM-сетях состояние ячейки передаётся по всей цепочке модулей, и при помощи входного фильтра и фильтра забывания определяется, какая информация передаётся в следующие ячейки, а какая удаляется. Таким образом, это позволяет максимально учитывать зависимости между входными сигналами.

Описание решения задачи

Рассматриваемый метод, основанный на рекуррентных нейронных сетях, представляет собой задачу классификации текстов по тональности. Предполагается выполнять классификацию текстов по трём классам: позитивная, нейтральная или негативная тональность.

Для решения рассматриваемой задачи при помощи машинного обучения с учителем требуется программно реализовать архитектуру LSTM нейронной сети, поскольку, как уже говорилось выше, стандартные нейронные сети прямого распространения неэффективны в задачах обработки естественных языков. Предполагается разработать программный модуль анализа тональности текста на языке программирования Python с использованием библиотеки Keras [7], которая позволяет достаточно гибко конструировать искусственные нейронные сети.

Кроме того требуется программный модуль, который будет производить преобразование текста, а именно выполнять задачи перевода символов в один из регистров, удаления цифр, знаков пунктуации, пробелов и прочих сторонних символов, векторизации слов, так как нейронной сети требуется получать на вход вектор, который представляет слово. Для реализации также предполагается использовать язык Python, поскольку на текущем этапе такой подход может упростить интеграцию данного модуля с самой моделью. Однако в случае, если потребуется высокая производительность, то следует задуматься о реализации на другом языке программирования.

Также требуется подготовить набор данных для обучения и тестирования реализованной модели. Разметка данных представляет собой одну из трудоёмких задач.

Далее требуется оптимизировать структуру используемой модели машинного обучения в зависимости от достигаемых результатов в точности работы рекуррентной нейросети.

Заключение

В статье рассмотрена задача определения тональности текста на основе рекуррентных нейронных сетей. Выявлены недостатки, связанные с использованием архитектуры данных сетей в непосредственном виде. Поэтому была рассмотрена модификация в виде LSTM-сети, которая позволит получить меньшую ошибку при работе модели. Рассмотрено возможное описание решения задачи, в котором выявлены основные моменты и сложности, которые могут возникнуть в процессе реализации.

Разработанный и протестированный модуль предполагается в дальнейшем использовать как часть системы по выявлению признаков мошенничества в текстах на естественном языке.

Кроме того при необходимости возможна адаптация данного модуля для применения в других похожих программно-информационных системах финансово-банковской сферы.

Библиографический список

1. Определение сценария заведения расследований | 21B943050 | ЦФТ-AML [Электронный ресурс]. – Режим доступа: <http://catalog.cft.ru/aml/SitePages/App.aspx?Code=21B943050>, свободный. – Дата доступа: 01.02.2021.
2. ЦБ назвал число пострадавших от банковского мошенничества | Новости | Известия | 18.01.2021 [Электронный ресурс]. – Режим доступа: <https://iz.ru/1112751/2021-01-18/tcb-nazval-chislo-postradavshikh-ot-bankovskogo-moshennichestva>, свободный. – Дата доступа: 01.02.2021.
3. Agarwal, B., Nayak, R., Mittal, N., Patnaik, S. Deep Learning-Based Approaches for Sentiment Analysis, Springer Nature, 2020.
4. Лащилин М. А., Селиванов Е. В. Основные модели машинного обучения в финансово-банковской сфере. – Новые информационные технологии в научных исследованиях, 2020.
5. Hochreiter, S., Schmidhuber, J. Long short-term memory. Neural Computation 9(8), 1997, pp. 1735-1080.
6. Chung, J., Gulcehre, C., Cho, K., Bengio, Y. Empirical Evaluation of Gated Recurrent Neural Networks on Sequence Modeling, arXiv:1412.3555.
7. Developer guides [Электронный ресурс]. – Режим доступа: <https://keras.io/guides/>, свободный. – Дата доступа: 10.02.2021.

УДК 004.896; ГРНТИ 28.23.29

ПРОЕКТИРОВАНИЕ ИНТЕЛЛЕКТУАЛЬНОЙ СИСТЕМЫ ПОДБОРА КРЕДИТНЫХ ПРОДУКТОВ

К.Г. Шитова

*Рязанский государственный радиотехнический университет имени В.Ф. Уткина,
Российская Федерация, Рязань, kshitova@inbox.ru*

Аннотация. В работе описывается актуальность автоматизации задачи выдачи кредита и рекомендации подходящего продукта клиенту. Определяются прецеденты, описывающие функциональность системы, и связи между ними. Выполняется построение диаграммы прецедентов и диаграмм деятельности.

Ключевые слова: кредитоспособность, актор, диаграмма прецедентов, ассоциация, отношение включения, отношение расширения, диаграмма деятельности.

DESIGNING AN INTELLIGENT SYSTEM FOR SELECTING LOAN PRODUCTS

K.G. Shitova

*Ryazan State Radio Engineering University named after V.F. Utkin,
Russian Federation, Ryazan, kshitova@inbox.ru*

Abstract. The paper describes the relevance of automating the task of issuing a loan and recommending a suitable product to the client. Use cases are defined that describe the functionality of the system and the relationships between them. Precedent and activity diagrams are being built.

Keywords: creditworthiness, actor, use case diagram, association, inclusion relation, expansion relation, activity diagram.

Оценка кредитоспособности предприятия – это одна из важнейших операций банков. Имея возможность наиболее точно оценить уровень кредитоспособности заемщика, банки смогут выбрать себе более выгодных клиентов, которые в срок погасят кредит. Для этого банкам требуется система, которая позволит быстро и без лишних усилий определить, можно ли одобрить заявку на кредит конкретному заемщику или нет. А также огромным плюсом будет, если программа будет формировать рекомендации по предлагаемому кредитному продукту, который подходит конкретному заемщику. Задача оценки кредитоспособности относится к области предиктивной аналитики [4–7].

Предметная цель проекта – повышение производительности труда сотрудников банка и уменьшение количества ошибок при оценке заемщика и формировании наиболее подходящего кредитного продукта. Без такой системы сотрудникам банка придется вручную обрабатывать данные заемщиков, что займет большое количество времени, что не позволит обслужить больше клиентов. Используя разрабатываемую систему, сотрудник банка сможет проанализировать характеристики клиента за пару минут и перейти к обслуживанию следующего. Таким образом, клиентов у банка станет больше, так как возрастет производительность труда сотрудников, и поскольку программа будет подбирать кредитный продукт, основываясь на данных клиента. Так же при расчете вручную человек может допустить ошибку, например, кредит будет одобрен совершенно некредитоспособному клиенту, и он не сможет его погасить. Разрабатываемая система поможет сотрудникам банка избежать такого поворота событий и одобрить кредит только надежным заемщикам.

Средством достижения поставленной цели является разработка информационной системы, определяющей возможность выдачи кредита и подбирающей кредитный продукт. В основе разрабатываемой системы лежит многослойная нейронная сеть для распознавания надежности заемщиков.

Для разработки такой системы необходимо построить диаграмму прецедентов, которая предназначена для описания функционального назначения системы. Основные элементы диаграммы – варианты ее использования и акторы, потребности которых они удовлетворяют [1–3]. Диаграмма прецедентов представлена на рисунке 1.

Актор «Работник банка» представляет пользователя системы. Он использует единственную значимую для него функцию системы – «Формирование рекомендаций по кредитному продукту».

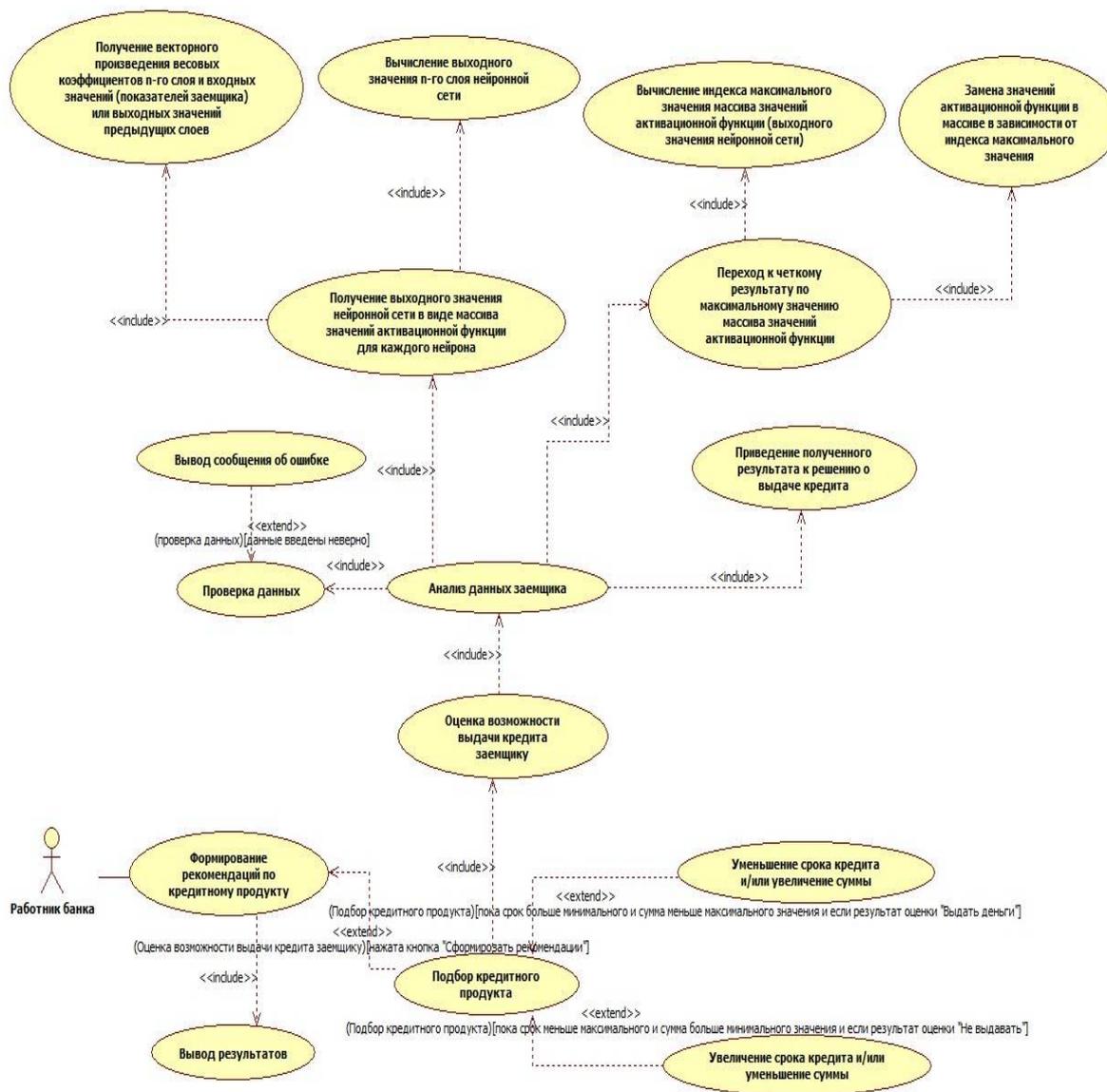


Рис. 1. Диаграмма прецедентов «Формирование рекомендаций по кредитному продукту»

«Формирование рекомендаций по кредитному продукту» – базисный прецедент, связанный с актором обязательной ассоциацией. Это единственная функция системы, доставляющая актору значимый для него результат.

Остальные прецеденты реализуют функциональности базисного прецедента.

Отношение между актором «Работник банка» и прецедентом «Формирование рекомендаций по кредитному продукту» является ассоциацией.

Отношение между базисным прецедентом «Формирование рекомендаций по кредитному продукту» и прецедентом «Вывод результата» является разновидностью отношения зависимости, называемой включением, что символизируется стереотипом «include». Это отношение отражает тот факт, что функциональность прецедента «Вывод результата» является обязательной составляющей прецедента «Формирование рекомендаций по кредитному продукту», т.к. при оценке кредитоспособности результат оценки всегда выводится на экран.

Так же отношение включения можно наблюдать между прецедентами, которые соединены стрелкой со стереотипом «include».

Отношение между базисным прецедентом «Формирование рекомендаций по кредитному продукту» и прецедентом «Подбор кредитного продукта» также является разновидностью отношения зависимости, но называемым расширением, что символизируется стереотипом «extend». Это отношение отражает тот факт, что функциональность прецедента «Подбор кредитного продукта» является необязательной составляющей прецедента «Формирование рекомендаций по кредитному продукту»: функциональность «Подбор кредитного продукта» реализуется только тогда, когда пользователь нажал на кнопку «Сформировать рекомендации». Так же отношение расширения можно наблюдать между прецедентами, которые соединены стрелкой со стереотипом «extend».

Для реализации вариантов использования строится диаграмма деятельности. Диаграмма деятельности так же, как и спецификация прецедента, отображает алгоритм его реализации, но более наглядным образом [1–3]. На рисунке 2 приведена диаграмма деятельности реализации прецедента «Формирование рекомендаций по кредитному продукту». Из нее можно наблюдать, что после того как пользователь запустил программу, система загружает форму программы, а также поля ввода данных и кнопку «Сформировать рекомендации».

Далее пользователь вводит показатели заемщика. Если после этого нажата кнопка «Сформировать рекомендации», то система переходит к выполнению прецедента «Подбор кредитного продукта», после чего выводит результат оценки пользователю на экран.

Если пользователь нажал на кнопку выхода, то система завершает работу программы. Иначе система ожидает действия пользователя.

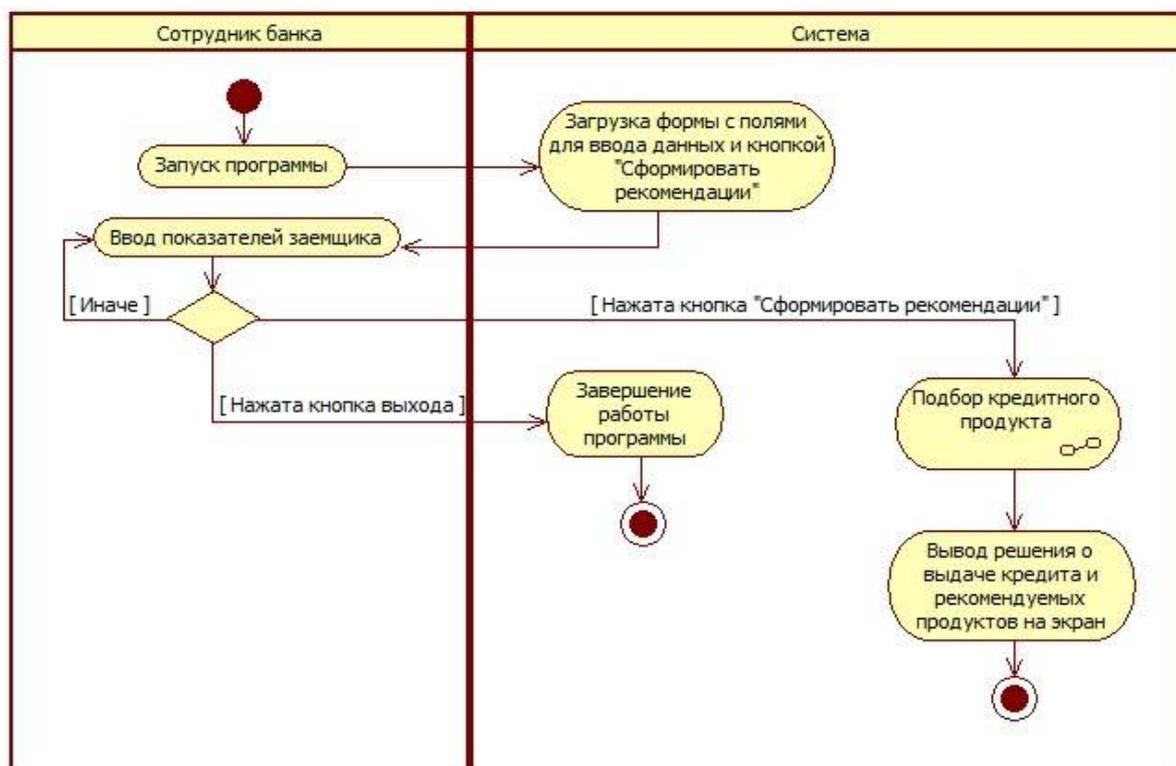


Рис. 2. Диаграмма деятельности для прецедента «Формирование рекомендаций по кредитному продукту»

На рисунке 3 приведена диаграмма деятельности реализации прецедента «Подбор кредитного продукта».

На этой диаграмме показан алгоритм подбора параметров кредитного продукта, подходящих конкретному клиенту.

Сначала система выполняет оценку возможности выдачи кредита клиенту. Затем, в зависимости от принятого решения, вычисляет срок и сумму определенным алгоритмом.

Если оценка клиента положительная, то алгоритм пытается рассчитать и другие варианты кредитного продукта, подходящие клиенту, увеличивая сумму и уменьшая срок. Так клиент сможет узнать, какую максимальную сумму он может получить и с каким сроком в зависимости от его параметров.

При этом выполняются попытки увеличения срока кредита и уменьшения суммы. Если будут найдены такие параметры продукта, при которых клиенту будет одобрен кредит, то система определит, к какому продукту относятся найденные характеристики и предложит клиенту новые условия.

По данной работе можно сделать следующие выводы:

- 1) было обосновано, что решение задачи выдачи кредита и рекомендации подходящего кредитного продукта необходимо автоматизировать;
- 2) были определены прецеденты, описывающие функциональное назначение системы;
- 3) были определены связи между полученными прецедентами;
- 4) были построены диаграммы прецедентов и деятельности, которые в дальнейшем будут положены в основу разрабатываемого программного продукта

Если же оценка была получена отрицательная, то алгоритм пытается найти такие параметры кредитного продукта, которые будут одобрены банком для данного клиента.

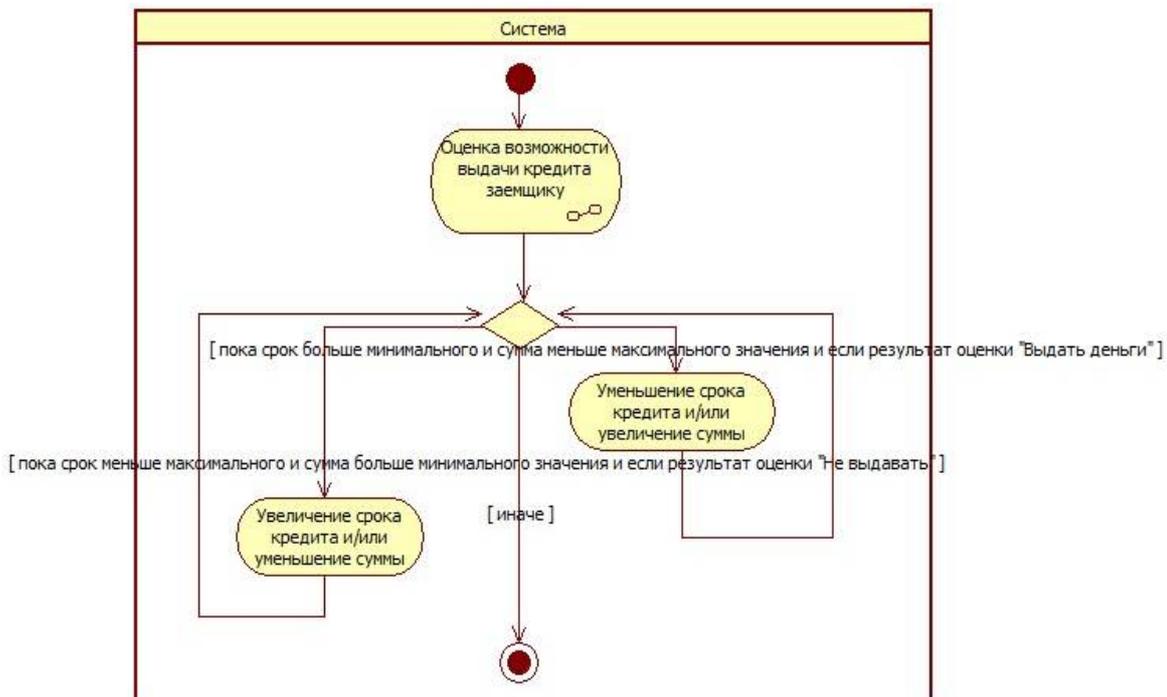


Рис. 3. Диаграмма деятельности для прецедента «Подбор кредитного продукта»

Библиографический список

1. Проектирование информационных систем: учебник / В.В. Белов, В.И. Чистякова. – М.:КУРС, 2018. – 400 с.
2. Визуальное моделирование систем в StarUML: Учебное пособие/ А.В. Каюмова. Казань. – Казанский федеральный университет, 2013. – 104 с.
3. Иванов Д.Ю., Новиков Ф.А. Основы моделирования на UML: Учеб. пособие. – СПб.: Изд-во Политехн. ун-та, 2010. – 249 с.

4. Шитова К.Г. Разработка интеллектуальной рекомендательной системы подбора кредитных продуктов для предприятий // Новые информационные технологии в научных исследованиях: материалы XXIV Всероссийской научно-технической конференции студентов, молодых ученых и специалистов. Т. 1 / РГРТУ. – Рязань, 2019. – С. 172-173.

5. Шитова К.Г. Разработка интеллектуальной рекомендательной системы подбора кредитных продуктов для предприятий на основе нейронных сетей // Современные технологии в науке и образовании – СТНО-2020: сб. тр. III междунар. науч.-техн. форума: в 10 т. Т. 5./ под общ. ред. О.В. Миловзорова; РГРТУ. – Рязань, 2020; Рязань. – С. 112-115.

6. Шитова К.Г. Интеллектуальная система подбора кредитных продуктов // Новые информационные технологии в научных исследованиях: материалы XXV Юбилейной Всероссийской научно-технической конференции студентов, молодых ученых и специалистов; РГРТУ – Рязань: ИП Коняхин А.В. (Book Jet), 2020 – С. 143-144.

7. Селиванов Е.В. Новые методы в предсказательной аналитике // Современные технологии в науке и образовании – СТНО-2020: сб. тр. III междунар. науч.-техн. форума: в 10 т. Т. 5./ под общ. ред. О.В. Миловзорова; РГРТУ. – Рязань, 2020; Рязань. – С. 152-155.

УДК 004.896; ГРНТИ 28.23.29

ИСПОЛЬЗОВАНИЕ МИКРОСЕРВИСНОЙ АРХИТЕКТУРЫ ПРИ РЕАЛИЗАЦИИ ИНФОРМАЦИОННОЙ СИСТЕМЫ ДЛЯ ОПТИМИЗАЦИИ РАБОТЫ ПАТРУЛЕЙ

Л.А. Иванов

*Рязанский государственный радиотехнический университет имени В.Ф. Уткина,
Российская Федерация, Рязань, o.oleonid@outlook.com*

Аннотация. В работе рассматривается разработка микросервисной архитектуры при реализации информационной системы для оптимизации работы патрулей. Происходит анализ и выявление требований к архитектуре. Приводится разработанная архитектура программной системы. Описываются выделенные сервисы. Поясняются выбранные средства разработки.

Ключевые слова: микросервисная архитектура, монолит, Java, Spring, распределенная система, REST.

USE OF MICROSERVICE ARCHITECTURE IN IMPLEMENTATION OF THE INFORMATION SYSTEM FOR OPTIMIZATION PATROL WORKS

L.A. Ivanov

*Ryazan State Radio Engineering University named after V.F. Utkin,
Russia, Ryazan, o.oleonid@outlook.com*

Abstract. The paper considers the development of a microservice architecture when implementing an information system to optimize the work of patrols. The architecture requirements are analyzed and identified. The developed architecture of the software system is presented. Dedicated services are described. Selected development tools are explained.

Keywords: microservice architecture, monolith, Java, Spring, distributed system, REST.

Большинство реализованных информационных систем на текущий момент спроектированы на монолитной архитектуре – система представляет собой одно приложение, модули и компоненты которого работают с единой базой данных. Такой подход удобен как с точки зрения разработки системы, все необходимые программные модули находятся в одном месте, так и в её развертывании и поддержке. Но данный подход имеет и ряд недостатков. Наиболее явный недостаток заключается в масштабировании таких систем при увеличении нагрузки. Для решения проблемы с возрастающей нагрузкой используют два подхода масштабирования: вертикальный – увеличение вычислительной мощности серверов, где развернута система, и горизонтальное – увеличение количества серверов с развернутой системой. Так как первый подход имеет свой предел, обычно применяют горизонтальный подход. Масшта-

бирование приложений, реализованных на монолитной архитектуре, имеет один, наиболее важный недостаток – не все модули монолитного приложения подвержены нагрузке. Увеличение количества серверов менее выгодно, чем масштабирование наиболее нагруженных модулей приложения. Для исправления данной проблемы используется микросервисной архитектура [1]. Рассмотрим её использование в рамках высоконагруженной информационной системы для оптимизации работы патрулей [2].

Анализ и выявление требований

Одним из наиболее важных требований к информационной системе, реализуемой на микросервисной архитектуре, является корректное разделение. Полученные микросервисы должны быть независимыми и легко масштабируемыми.

Доступ к сервисам должен быть контролируемым – взаимодействие между сервисами должно происходить во внутренней сети. Необходимо учесть маршрутизацию при распределении трафика между сервисами.

При реализации взаимодействия между сервисами необходимо учесть отказоустойчивость и кеширование ответов.

Необходимо предоставить облегченный интерфейс для взаимодействия с системой.

Конфигурация системы должна находиться в отдельном сервисе и должна быть легко изменяемой.

Система должна включать в себя сервисы по отслеживанию состояния.

Необходим сервис для централизованного сбора данных журнала.

Архитектура информационной системы

На рисунке 1 представлена микросервисная архитектура информационной системы. Основные сервисы выделены во внутренней сети. Взаимодействие с внешней сетью осуществляется через шлюз – *Gateway*.

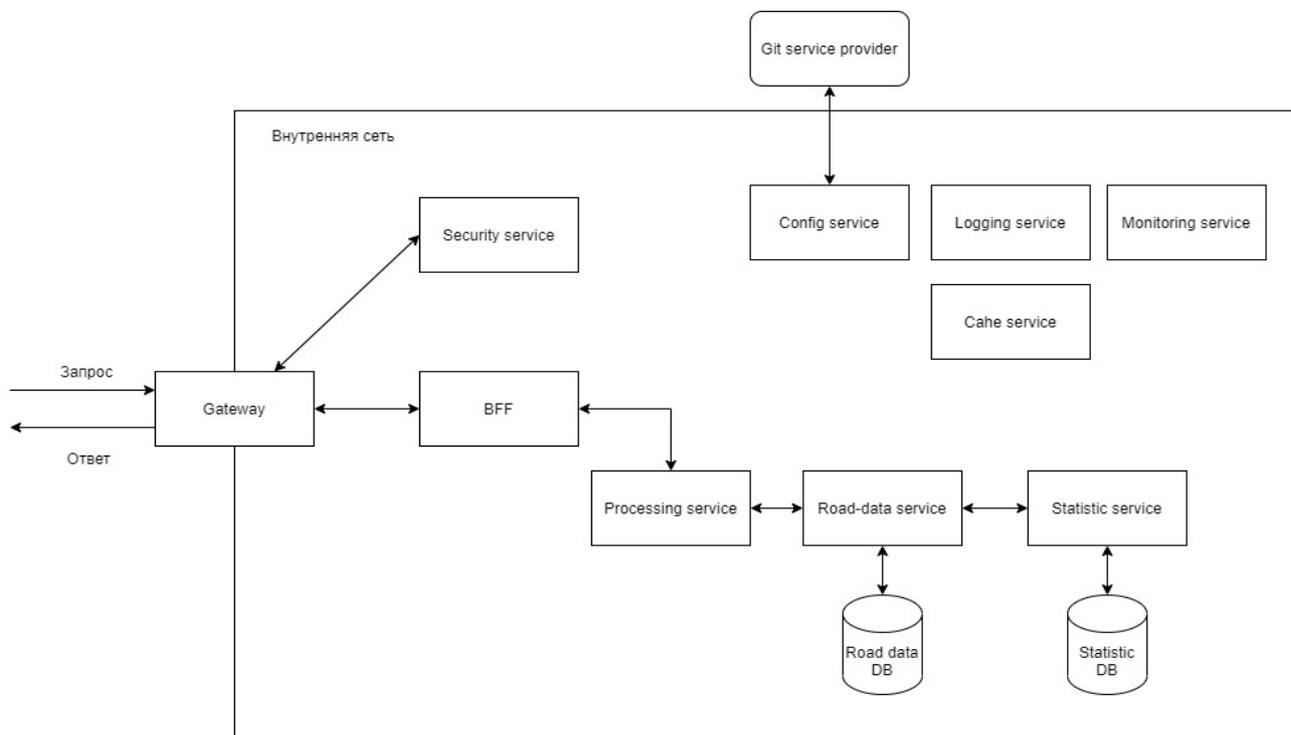


Рис. 1. Микросервисная архитектура информационной системы

После того, как приходит запрос в систему, сначала проверяется наличие прав доступа – запрос в *Securityservice*. Если доступ к системе есть, происходит маршрутизация запроса.

Общим интерфейсом взаимодействия с системой является сервис *BFF* (*BackendForFrontend*). Данный сервис предоставляет публичный API для взаимодействия с системой. Внешние запросы могут взаимодействовать только с ним.

Сервисы *Processingservice*, *Road-dataservice* и *Statisticsservice* являются основными логическими сервисами системы. Они содержат логику по вычислению оптимального расположения патруля в зависимости от статистики вызовов и дорожной ситуации в регионе.

К обслуживаемым сервисам относятся:

- *Configservice* – сервис по предоставлению конфигурационных настроек для всех микросервисов в системе. Изменение настроек осуществляется через *Git* репозиторий.

- *Loggingservice* – сервис централизованного сбора файлов журналов и взаимодействия с ними.

- *Monitoringservice* – сервис мониторинга состояния системы.

- *Cacheservice* – сервис кеширования необходимых данных.

Взаимодействие между сервисами осуществляется с помощью *RESTAPI*. Конфигурирование адресных данных сервисов, их распределение и увеличение экземпляров в текущей реализации осуществляется с помощью внутренней сети.

Средства разработки

В качестве основного языка программирования был выбран *Java11* и фреймворк *Spring*[3]. Модуль *SpringCloud* позволяет легко реализовать микросервисную архитектуру.

В качестве шлюза используется реактивный *SpringCloudGateway*. Он обладает гибкой настройкой маршрутизации и фильтрации входящих запросов.

Для реализации *Security* сервиса был выбран *Keycloak* – готовый продукт, обладающий реализацией *singlesign-on* с возможностью управления доступом. Взаимодействие с *SpringCloudGateway* осуществляется спомощью открытого протокола авторизации *OAuth 2.0*.

Для реализации *Config* сервиса был выбран *SpringCloudConfig*. Он обеспечивает поддержку на стороне сервера и на стороне клиента внешней конфигурации в распределенной системе. Настройки конфигурации хранятся в приватном *git* репозитории на платформе *GitLab*.

Для реализации централизованного логирования был выбран технологический стек *ELKStack: Elasticsearch, Logstash, Kibana* – данный стек позволяет собирать файлы журналов в единое хранилище различными средствами (*Logstash*), осуществлять полнотекстовый поиск по всем данным (*Elasticsearch*) и визуализировать результат в удобном *web*-терминале (*Kibana*).

Для реализации сервиса мониторинга был выбран стек *Prometheus* совместно с *Grafana*. Сбор данных о состоянии микросервисов осуществляется посредством гибких *HTTP* запросов (*Prometheus*). Отображение полученных данных происходит в *web*-терминале (*Grafana*).

В качестве сервиса кеширования был выбран *Hazelcast – In-MemoryDataGrid* решение с открытым исходным кодом на основе *Java*. Выделение кеша в отдельный сервис позволяет пользоваться им во всех сервисах информационной системы.

BFF, Processing service, Road-data service и *Statistic service* реализованы с помощью *Spring MVC*.

Для автоматизации развёртывания, масштабирования и управления сервисами был выбран *Kubernetes* – портативная расширяемая платформа с открытым исходным кодом для управления контейнеризованными рабочими нагрузками и сервисами.

Для контейнеризации разработанных сервисов используется *Docker*.

Заключение

Использование микросервисной архитектуры позволило разделить информационную систему для оптимизации работы патрулей на независимые сервисы. Используемый подход контейнеризации и развертывания системы в *Kubernetes* упростил развёртывание и масштабирование системы.

Библиографический список

1. Клеппман М. Высоконагруженные приложения. Программирование, масштабирование, поддержка. – СПб.: Питер, 2018. – 640 с.
2. Иванов Л.А. Разработка информационной системы для оптимизации работы патрулей экстренных служб //Современные технологии в науке и образовании – СТНО-2020 [текст]: сб. тр. III междунар. науч.-техн. форума: в 10 т. Т.3./ под общ. ред. О.В. Миловзорова. – Рязань: Рязан. гос. радиотехн. ун-т, 2020; Рязань. – 162 с., ил.
3. Козмина Ю., Харроп Р., Шефер К., Хо К. Spring 5 для профессионалов.: Пер. с англ. – СПб.: ООО «Диалектика» 2019. – 1120с.

УДК 681.5; ГРНТИ 50.43

РАЗРАБОТКА ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ ДЛЯ ИССЛЕДОВАНИЯ МЕТРОЛОГИЧЕСКИХ ХАРАКТЕРИСТИК СОЛЕМЕРОВ

П. Николов

*Рязанский государственный радиотехнический университет имени В.Ф. Уткина,
Российская Федерация, Рязань, takisnikolov@gmail.com*

Аннотация. Солемер является составной частью автоматизированных систем контроля и управления системами как первичное средство сбора и обработки информации. Каждый прибор перед эксплуатацией проходит ряд проверок, одной из проверок является корректное считывание солености раствора. В работе производится описание процесса поверки солемеров и архитектура программного обеспечения для удаленной поверки и градуировки.

Ключевые слова: автоматизированная система, удаленное управление.

DEVELOPMENT OF SOFTWARE FOR RESEARCH OF METROLOGICAL CHARACTERISTICS OF SALINOMETERS

P. Nikolov

*Ryazan State Radio Engineering University named after V.F. Utkin,
Russian Federation, Ryazan, takisnikolov@gmail.com*

Abstract. Salinometer is an integral part of automated systems control and administration systems as a primary tools of information collection and processing. Each device, before exploitation, passes a series of checks, one of the checks is a correct reading of the salinity of the solution. The paper discusses the salinometer verification process and the software architecture for remote verification and calibration.

Keywords: automated system, remote control.

Введение

Автоматизация процесса сокращает затраты предприятия, как финансовые, так и человеко-часы. Один из способов автоматизации процесса – реализация программного обеспечения для поверки и градуировки приборов. Произведенный измерительный прибор перед эксплуатацией проходит ряд проверок и при необходимости градуируется. Процесс поверки занимает продолжительное время, во время которого идет сбор и анализ данных. Один из таких приборов солемер.

Солемер – специальный прибор, предназначенный для измерения концентрации солей в жидкости, является составной частью автоматизированных систем, контроля и управления как первичное средство сбора и обработки информации. Применяют солемеры в системе автоматического контроля параметров энергетических установок.

Цель работы – увеличение производительности путем повышения скорости поверки солемеров и понижения времени затрачиваемого на контроль поверки. Для достижения поставленной цели необходимо решить следующие задачи:

- избавление процесса поверки прибора от постоянного контроля;
- предоставление многопользовательского доступа;
- поверка нескольких солемеров одновременно;
- анализ данных солемера.

Для достижения поставленной цели предлагается разработка программного обеспечения для исследования метрологических характеристик солемеров.

Процесс поверки и градуировки солемера

После производства солемер проходит ряд проверок. Процесс поверки состоит из нескольких этапов. В рамках данной работы рассмотрим этап поверки солемера на считывание данных о солесодержании раствора.

Солемер подключается к ванной с водой, концентрация солености которой около 0,5 мг/л (полученная в лабораторных условиях). Также к системе подключается лабораторный прибор для измерения точного солесодержания в растворе. Схема подключения поверяемого солемера приведена на рисунке 1, стрелками показано движение раствора.



Рис. 1. Схема подключения поверяемого солемера к системе поверки

После подключения считываются показания с поверяемого солемера. Далее изменяется солесодержание раствора (добавляется более концентрированный раствор) и снова считываются показания.

Операция изменения солености раствора и считывания показаний повторяется n раз.

Прибор градуируется при обнаружении недопустимого отклонения показаний поверяемого солемера от лабораторного.

Анализ существующих решений

На данный момент предприятие-изготовитель сбора и градуировки солемеров использует программу, разработанную штатным программистом, для взаимодействия с солемером (см. рисунок 2).

Основные недостатки:

1. Нет возможности контролировать процесс удаленно.
2. Отсутствует возможность многопользовательского доступа.
3. Отсутствует возможность подключить несколько солемеров.
4. Градуировочные данные хранятся не структурированно.
5. Главное окно программы имеет перегруженный интерфейс.

Основные достоинства:

1. Настройка COM порта.

2. Скорость выполнения команд при взаимодействии с солемером.

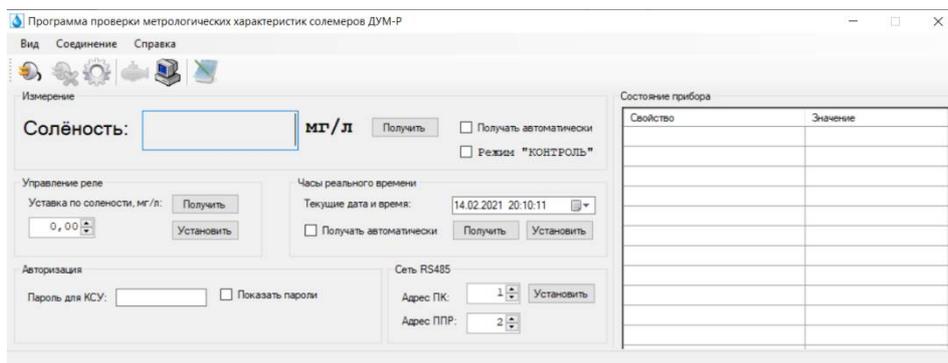


Рис. 2. Программа для поверки солемеров

Обоснование актуальности работы

Разработка автоматизированной системы удаленного управления процессом поверки и градуировки солемеров позволит решить следующие задачи:

- избавление процесса поверки прибора от постоянного контроля;
- предоставление многопользовательского доступа;
- возможность подключения нескольких солемеров;
- удаленный доступ;
- анализ и хранение градуировочных данных в реляционной базе данных.

Решение всех перечисленных задач позволит повысить производительность поверки и градуировки солемеров.

Удаленный доступ позволяет оператору находиться на любом рабочем месте, при условии, что рабочее место оборудовано компьютером с доступом к локальной сети.

Избавление процесса поверки прибора от постоянного контроля позволяет оператору заниматься другими делами, пока идет сбор данных.

Многопользовательский доступ и возможность подключения нескольких солемеров позволяет работать нескольким операторам с несколькими солемерами одновременно.

Одним из ключевых факторов в разрабатываемой системе является ее интерфейс, так как через него происходит взаимодействие с системой. Интерфейс не только решает проблему взаимодействия с приложением, но и делает это взаимодействие максимально комфортным.

Поскольку разрабатываемая программная система решает вышеперечисленные задачи, то можно сделать вывод, что разработка является актуальной и необходимой с точки зрения повышения производительности.

Структура разрабатываемой системы

Целью проекта является разработка программного обеспечения для градуировки и поверки солемеров, для реализации которого необходимо реализовать возможность удаленного управления.

Разрабатываемое программное обеспечения состоит из 2 частей (выделено прямоугольником на рисунке 3):

- серверная часть;
- клиентская часть.

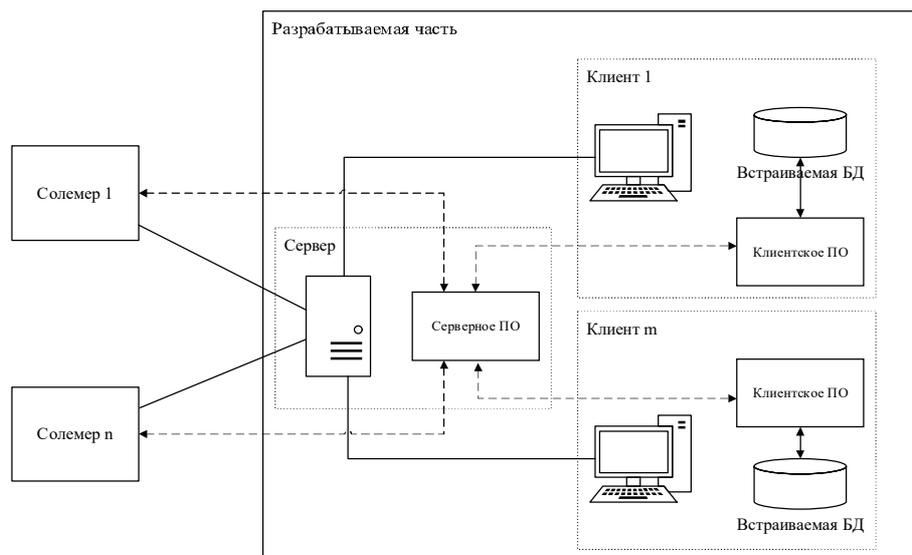


Рис. 3. Структура разрабатываемой системы

Основная задача сервера – обеспечение связи между клиентом и солемером. Сервер должен быть реализован таким образом, чтобы к нему могли подключаться несколько клиентов и солемеров для реализации многопользовательского доступа и возможности контролировать несколько солемеров одновременно.

Клиентская часть должна выполнять отправку команд солемеру, хранение градуировочных данных, анализ данных, позволять автоматически считывать данные с солемера.

Хранение данных будет реализовано с помощью встраиваемой базы данных, которая обеспечивает хранение данных локально и имеет «легкий» вес. Также, при работе со встраиваемой БД нет необходимости в администрировании. Градуировочные данные представляют набор различных данных, хранение которых проще реализовать с помощью реляционной БД.

Анализ градуировочных данных должен состоять из построения различных графиков (например, график ошибок, график точек), расчета аппроксимационных точек, расчета аппроксимационных коэффициентов функции.

Архитектура «клиент-сервер»

Архитектура «клиент-сервер» определяет общие принципы организации взаимодействия в сети, где имеются серверы, узлы-поставщики некоторых специфичных функций (сервисов) и клиенты (потребители этих функций).

В программной системе для контроля процесса поверки солемеров сервер играет промежуточную роль, что характерно для трехуровневой архитектуры клиент-сервер.

Трехуровневая архитектура приложений – это модульная клиент-серверная архитектура, которая состоит из уровня представления, уровня приложения и уровня данных. Диаграмма трехуровневой архитектуры «клиент-сервер» представлена на рисунке 4.

Уровень данных – поверяемый солемер. Данными, в такой ситуации, являются ответ солемера на команду от клиента.

Уровень приложения – сервер, промежуточная функция, интерпретирует запросы и реализует взаимодействие между клиентом и солемером.

Уровень представления – клиент, настольная программа с графическим интерфейсом для взаимодействия с пользователем.



Рис. 4. Диаграмма клиент-серверной архитектуры

Процесс поверки и градуировки прибора солемер состоит из расчета и анализа градуировочных данных. Весь процесс поверки и градуировки выполняется на стороне клиента, что делает его «толстым».

«Толстый» клиент – выполняет основную часть обработки в клиент-серверных приложениях на стороне клиента.

Протоколы передачи данных

В данном проекте процесс обмена данными между клиентом и солемером состоит из двух частей:

1. Обмен данными между клиентом и сервером, по локальной сети.
2. Обмен данными между сервером и солемером, через COM порт.

Протокол передачи данных между клиентом и сервером

Обмен данными между клиентом и сервером будет осуществляться по локальной сети.

Локальными сетями называют частные сети, размещающиеся, как правило, в одном здании или на территории какой-либо организации. Их часто используют для объединения компьютеров и рабочих станций в офисах компании или предприятия бытовой электроники для предоставления совместного доступа к ресурсам (например, принтерам) и обмена информацией [1].

В локальной сети каждое устройство называется узлом, которое идентифицируется уникальным сетевым адресом – IP-адресом.

Основной критерий выбора протокола передачи данных по локальной сети между клиентом и сервером – это надежность передачи данных. Протокол TCP среди существующих протоколов наиболее подходящий для осуществления надежного потока данных.

Протокол TCP (Transmission Control Protocol – протокол управления передачей) был специально разработан для обеспечения надежного сквозного байтового потока по ненадежной интернет-сети. Объединенная сеть отличается от отдельной сети тем, что ее различные участки могут обладать сильно различающейся топологией, пропускной способностью, значениями времени задержки, размерами пакетов и другими параметрами. При разработке TCP основное внимание уделялось способности протокола адаптироваться к свойствам объединенной сети и отказоустойчивости при возникновении различных проблем [1].

TCP/IP – сетевая модель передачи данных, представленных в цифровом виде. Модель описывает способ передачи данных от источника информации к получателю [2].

Протокол передачи данных между сервером и солемером

Информация о протоколе передачи данных солемера была взята из технических условий к прибору.

Протокол обмена данными между солемером и системой управления верхнего уровня осуществляется по интерфейсу RS-485.

Солемер обменивается данными в режиме запрос-ответ, где каждый запрос и ответ является пакетом байтов. Для идентификации начала и конца пакета используются специальные символы. Для обнаружения ошибок в пакете данных используется циклический избыточный код.

Каждый солемер имеет свой адрес. При получении запроса солемер сверяет адрес получателя и при совпадении адреса формирует и отправляет ответ.

Для работы сервера по интерфейсу RS-485 используется COM порт.

Параметры COM порта [3]:

1) количество бит – количество битов для представления одного символа данных;
2) скорость передачи (бит/с) – число битов данных, передаваемых по линии связи за одну секунду;

3) число стоповых битов – используется для определения конца символа;

4) контроль четности – бит контроля чётности двоичного числа, значения:

none – не создается бит контроля четности;

even – четное, общее число двоичных единиц в символе. Если число четное, то 0, иначе 1;

odd – нечетное, аналогично опции контроля по четности, но число двоичных единиц должно быть нечетным.

Заключение

В данной статье описан способ автоматизации процесса поверки измерительного прибора – солемера. Был проведен анализ существующих решений и выявлены преимущества и недостатки. Также, исходя из поставленных задач, была разработана структура системы и выбраны архитектура и протокол передачи данных.

Библиографический список

1. Таненбаум Э., Уэзеролл Д. Компьютерные сети. 5-е изд. – СПб.: Питер, 2012. – 960 с.: ил.
2. TCP/IP – Википедия [Электронный ресурс]. URL: <https://ru.wikipedia.org/wiki/TCP/IP>.
3. Serial communication parameters (Параметры последовательной связи) [Электронный ресурс]. URL: https://www.ibm.com/support/knowledgecenter/ssw_aix_72/network/asynch_serialcomm_params.html

УДК 004.852; ГРНТИ 28.23.25

ВЫЯВЛЕНИЕ МОШЕННИЧЕСКИХ ОПЕРАЦИЙ С КРЕДИТНЫМИ КАРТАМИ С ПОМОЩЬЮ НЕЙРОСЕТЕВЫХ ТЕХНОЛОГИЙ

А.С. Танцев

*Рязанский государственный радиотехнический университет имени В.Ф. Уткина,
Российская Федерация, Рязань, tochta4@gmail.com*

Аннотация. В данной работе рассматривается алгоритм выявления мошеннических операций с кредитными картами. Описываются нейросетевые технологии, применяемые в этом алгоритме.

Ключевые слова: кредитная карта, мошенничество, классификация, нейронная сеть, несбалансированные данные, передискретизация, шумоподавляющий автокодировщик.

DETECTION OF FRAUDULENT CREDIT CARD TRANSACTIONS USING NEURAL NETWORK TECHNOLOGIES

A.S. Tantsev

*Ryazan State Radio Engineering University named after V.F. Utkin,
Russia, Ryazan, tochta4@gmail.com*

The summary. This paper discusses an algorithm for detecting fraudulent transactions with credit cards. The neural network technologies used in this algorithm are described.

Keywords: credit card, fraud, classification, neural network, unbalanced data, oversampling, denoising autoencoder.

В настоящее время мошенничество – это враг номер один в финансовом мире. При этом существуют различные виды мошенничества: страховое мошенничество, мошенничество с кредитными картами, мошенничество с выписками, мошенничество с ценными бумагами и т. д. Из всех видов мошенничества мошенничество с кредитными картами является наиболее распространенным видом. Оно заключается в несанкционированном использовании счета кредитной карты. Это происходит, когда владелец карты не знает, что карта используется третьим лицом. Мошенники могут получить товар без оплаты или получить незаконный доступ к средствам со счета.

Выявление такого рода мошенничества касается большого числа финансовых учреждений и, в первую очередь, банков, поскольку это преступление обходится им примерно в 67 миллиардов долларов в год. Постоянный рост этой проблемы активно способствует развитию новых технологий противодействия мошенникам [1].

Одним из вариантов решения проблемы обнаружения мошенничества с кредитными картами является искусственная нейронная сеть. Именно разработка и внедрение нейросети позволит обеспечить эффективный и быстрый мониторинг транзакций мошенничества с кредитными картами [2].

Однако наблюдается проблема классового дисбаланса при обучении нейронной сети, обнаруживающей мошенничество с кредитными картами. Эта проблема заключается в чрезвычайно несбалансированном и сильно искаженном распределении данных. Иными словами, соотношение мошеннической или преступной деятельности значительно меньше, чем законной и подлинной.

Классовый дисбаланс создает огромную проблему в выявлении характеристик мошеннической деятельности и выявлении моделей мошенничества. Причина в том, что большинство алгоритмов классификации, таких как логистическая регрессия, предполагают, что число элементов каждого класса примерно одинаково. Когда данные сильно не сбалансированы, алгоритм начнет делать прогнозы в пользу класса большинства. Но, к сожалению, именно меньшинство является тем классом, который необходимо идентифицировать. Если алгоритм классификации не способен выявить мошеннические схемы, то незаконные операции считаются законными, что наносит серьезный финансовый ущерб.

Поэтому для решения проблемы несбалансированной классификации необходимо использовать алгоритм передискретизации, который позволяет сбалансировать выборки между классом большинства и классом меньшинства. Данный алгоритм синтезирует новые образцы данных класса меньшинства, но при этом он может привести к появлению шума. В связи с этим нужно не только передискретизировать выборку класса меньшинства из-за неправильной классификации, но и удалить образовавшийся шум. А для удаления шума применяются специальные автокодировщики, которые называются шумоподавляющими.

Автокодировщик – это искусственная нейронная сеть, процесс обучения которой происходит без учителя. Целью автокодировщика является изучение представлений для восстановления характеристик набора данных, как правило, с целью уменьшения размерности. Простейшей формой автокодировщика является рекуррентная нейронная сеть с прямой связью, аналогичная многослойному персептрону. Как показано на рисунке 1, этот автокодировщик состоит из двух частей: кодировщик, декодировщик. Эти части содержат входной слой, один или несколько скрытых слоев и выходной слой. Существенная разница между автокодировщиком и многослойным персептроном заключается в том, что выходной слой автокодировщика имеет такое же количество нейронов, как и входной слой. То есть цель состоит в том, чтобы реконструировать свои собственные входные данные, а не предсказывать целевое значение по заданным входным данным.

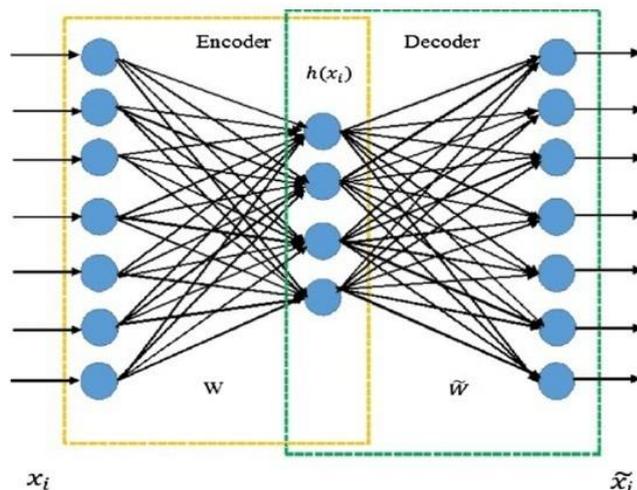


Рис. 1. Архитектура нейронной сети автокодировщика

В автокодировщике сетевая структура имеет связи между слоями, но не имеет связи внутри каждого слоя. Обучение нейронной сети автокодировщика заключается в оптимизации ошибки восстановления, используя заданные образцы данных. Функция стоимости нейронной сети автокодировщика равна:

$$J = \frac{1}{m} \sum_{i=1}^m \left(\frac{1}{2} \|\hat{x}_i - x_i\|^2 \right),$$

где x_i – входной образец данных;
 \hat{x}_i – выходная характеристика;
 m – количество входных образцов.

Также стоит отметить, что существует некая вариация автокодировщика – это шумоподавляющий автокодировщик. Нейронная сеть этого автокодировщика может научиться удалять шум и восстанавливать входные данные. То есть функция потерь пытается минимизировать разницу между выходными и исходными данными так, чтобы автокодировщик имел возможность устранять влияние шума и извлекать характеристики из поврежденных данных. При этом обычно шумом является гауссовский шум.

Передискретизация – это метод, используемый для работы с несбалансированным набором данных, его целью является создание специальной выборки классов, чтобы распределение классов исходного набора данных могло быть сбалансированным. Преимущество использования передискретизации показано на рисунке 2. Оранжевые точки на рисунке представляют случаи мошенничества, синие точки – законные операции. X и Y – это основные признаки классифицируемых транзакций.

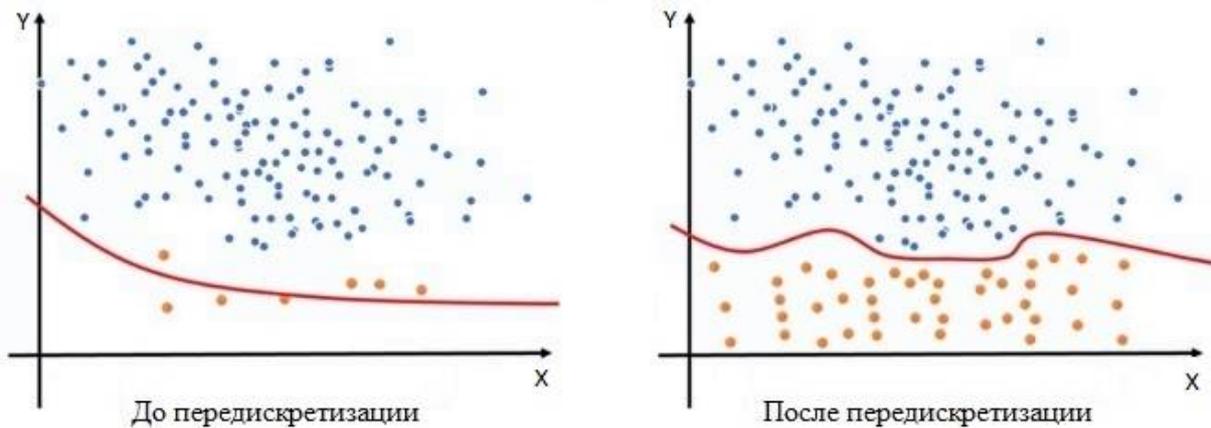


Рис. 2. Преимущество использования передискретизации

SMOTE (метод передискретизации синтетического меньшинства) – это один из самых популярных методов передискретизации. Для того, чтобы создать синтетическую точку данных, сначала нужно найти кластер k -ближайших соседей в пространстве объектов, затем случайным образом найти точку в этом кластере, наконец, использовать среднее взвешенное для «подделки» новой точки данных.

Для решения же самой задачи классификации часто используется глубокая полно-связная нейронная сеть. При этом модель классификации может достигать очень высокой точности с помощью перекрестной энтропии SoftMax в качестве функции потерь [3].

В конце стоит отметить, что точность не является достаточной для оценки классификационной модели, особенно для несбалансированного набора данных. Например, несбалансированным является набор данных с 99,9% нормальных данных и 0,1% аномальных данных. Если классификатор помечает полностью входную выборку, как нормальный класс, то модель может все еще достигать точности 99,9%. Поэтому для обнаружения аномалий очень важен такой параметр, как частота обнаружения класса аномалий. Частота обнаружения – это отношение между числом правильно обнаруженных аномалий и общим числом аномалий, оно оценивает, сколько аномалий может быть обнаружено в этой классификационной модели.

Таким образом, для выявления мошеннических транзакций необходимо, во-первых, использовать передискретизацию для преобразования несбалансированного набора данных в сбалансированный. Затем использовать шумоподавляющий автокодировщик, чтобы удалить шум, образовавшийся после передискретизации. И наконец, использовать нейронную сеть для окончательной классификации.

В дальнейшем в работе предполагается программно реализовать рассмотренный алгоритм выявления мошеннических операций с кредитными картами. При этом разработка будет происходить на языке Python в виртуальной среде Google Colab.

Библиографический список

1. Танцев А.С. Применение нейросетевых технологий для решения задач банковской сферы // Новые информационные технологии в научных исследованиях: материалы XXV Юбилейной Всероссийской научно-технической конференции студентов, молодых ученых и специалистов. Рязань: ИП Коняхин А.В. (Book Jet), 2020. – С. 41-42.
2. Головинов А.О., Климова Е.Н. Преимущества нейронных сетей перед традиционными алгоритмами // Экспериментальные и теоретические исследования в современной науке: сб. ст. по матер. V междунар. науч.-практ. конф. № 5(5). – Новосибирск: СибАК, 2017. – С. 11-15.
3. Семенов А. М. Интеллектуальные системы: учеб. пособие / Семенов А.М., Соловьев Н.А., Чернопрудова Е.Н., Цыганков А.С. – Оренбург: ОГУ, 2013. – 236 с.

УДК 004.3; ГРНТИ 50.33.14

РАЗРАБОТКА ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ ДЛЯ ПОДБОРА КОМПЛЕКТУЮЩИХ ДЛЯ КОМПЬЮТЕРА НА ОСНОВЕ МНОГОКРИТЕРИАЛЬНОЙ ВЫБОРКИ

О.Г. Швечкова, В.А. Елисеев

*Рязанский государственный радиотехнический университет имени В.Ф. Уткина,
Российская Федерация, Рязань, v.a.eliseev123@mail.ru*

Аннотация. В работе рассматривается процесс разработки автоматизированной системы для подбора комплектующих для компьютера на основе многокритериальной выборки. Приводятся основные особенности существующих методов подбора, их достоинства и недостатки, а также обоснование необходимости автоматизации данного процесса.

Ключевые слова: Персональный компьютер, комплектующие, эффективность, производительность.

DEVELOPMENT OF THE SOFTWARE FOR SELECTION HARDWARE COMPONENTS FOR COMPUTER USING MULTI-CRITERION SAMPLING

O.G. Shvechkova, V.A. Eliseev

*Ryazan State Radio Engineering University named after V.F. Utkin,
Russia, Ryazan, v.a.eliseev123@mail.ru*

The summary. The paper considers the process of developing an automated system for selection hardware components for computer using multi-criterion sampling. The main features of the existing selection methods, their advantages and disadvantages, as well as the rationale for the need to automate this process are presented.

Keywords: personal computer, hardware components, efficiency, productivity.

В настоящее время подбор комплектующих для компьютера является весьма важной задачей, т.к. текущее положение дел в стране и в мире в целом вызвал бурный, можно сказать лавинообразный рост спроса на средства вычислительной техники. Увеличение числа рабочих мест и объема дистанционного обучения стимулировал спрос на персональные компьютеры (ПК) как со стороны бизнеса, так и со стороны потребителей. За 2020 год объем продаж персональных компьютеров увеличился на 13.1% и составил 301.2 млн штук. По мнению компетентных источников, к концу 2020 года спрос на ПК превышал объемы производства, а каналы поставок работали на максимуме своих возможностей. Бренды и контрактные производители столкнулись с нехваткой комплектующих, также наблюдались проблемы с логистикой. Поставщики были вынуждены прибегать к авиаперевозкам, что привело к увеличению затрат, но позволило ускорить доставку. Все эти факторы привели к острому дефициту на некоторые виды комплектующих, что привело к их подорожанию, которое в отдельных случаях составляло 150-200%.

На данном этапе развития информационного общества трудно найти ту сферу деятельности человека, которая могла бы обойтись без применения ЭВМ. Отсюда следует вывод, что объемы производства и применения средств вычислительной техники во всех сферах деятельности человека настоятельно требуют не просто универсальных вычислительных устройств, а компьютеров, ориентированных на решение определённого класса задач. Выбор вычислительного устройства при таком подходе сводится к применению ЭВМ с набором функциональных возможностей, определяемым набором комплектующих компонентов, сформированным по определённым критериям эффективности.

Эффективность заключается в максимально быстром решении той задачи, для которой требуется компьютер. Так, для одних сфер деятельности наиболее важной является производительность процессора, которая, в свою очередь, тоже имеет ряд параметров (производительность на одно ядро, объемы кеша, многоядерная производительность). Для работы с графикой мощность процессора отходит на второй план и важнейшей является производи-

тельность видеокарты (производительность ГПУ, объемы видеопамати и поддерживаемые видеоинтерфейсы). Затрагивая понятия эффективности, нельзя обойти стороной и количество потребляемой энергии, которая затрачивается для решения определенных задач. Существуют комплектующие, которые имеют схожую производительность, но количество энергии, которую они затратят на одну и ту же задачу, может существенно отличаться. Этот параметр зависит прежде всего от техпроцесса производства, а также от эффективности архитектуры, на основе которой разрабатывается чип. Для других типов задач высокая производительность не требуется, а важнейшим становится надежность и бесперебойная работа. Для повышения надежности используются более производительные системы охлаждения, которые позволяют работать комплектующим при более низкой температуре, а также качественные блоки питания, которые позволят бесперебойно работать всей системе даже при скачках напряжения или других неполадках в сети.

Учет всех требований пользователя при подборе комплектующих представляется трудоёмкой задачей. В настоящее время эта проблема решается покупкой готовых решений в магазинах бытовой электроники или специализированных компьютерных магазинах, а также путем обращения к специалистам, готовым подобрать, подходящее для ваших задач решение.

Однако, данные подходы и применяемые методы не могут считаться универсальными и эффективными, так как имеются очевидные проблемы. В случае покупки готовых сборок в магазинах электроники возникает сразу несколько проблем.

Первая – это ограниченность выбора среди того количества собранных ПК, которое есть в магазинах. Это в свою очередь приводит к тому, что ни один из предложенных вариантов не будет сбалансированным и всецело подходящим под требуемые задачи.

Вторая проблема не менее, а возможно, и более критическая. Ее суть заключается в профессионализме консультанта, который будет помогать вам определяться с выбором компьютера. Так, во многих магазинах (даже в тех, которые ориентированы непосредственно на продажу компьютеров) берут на работу специалистов без каких-либо профильных знаний, не говоря уже про образование. А человеку, который слабо разбирается в теме компьютерных комплектующих, будет тяжело отличить профессионала, от неопытного консультанта.

Третья проблема состоит в том, что консультанты, даже если они и обладают достаточным уровнем знаний, зачастую могут рекомендовать не тот продукт, который является лучшим, а тот который им выгоднее продать.

В случае обращения к специалистам по подбору ПК также возникает ряд проблем, основной из которой также является его профессионализм. Конечно, в современном мире с помощью интернета можно посмотреть отзывы и мнения о качестве работы компаний, но существует также и вероятность того, что негативные отзывы были удалены, а позитивные были накручены.

Для решения перечисленных проблем и устранения имеющихся недостатков существующих методов мною предложен алгоритм автоматизированного подбора комплектующих на основе многокритериальной выборки, который позволит учесть специфику требуемых задач и выполнить подбор комплектующих, наиболее соответствующих решению поставленных задач.

Суть предлагаемого метода состоит в том, что разрабатываемая система будет определять комплектующие, которые имеют лучшие показатели для данного вида задач, учитывая при этом бюджет, который пользователь готов потратить на покупку ПК.

В зависимости от типа решаемых задач, траты на отдельный вид комплектующих (процессор, материнская плата, видеокарта и т.д.), относительно всего компьютера, могут достаточно сильно колебаться. Путем анализа существующих цен на комплектующие и их влияние на производительность в тех или иных задачах были получены данные о процентном соотношении конкретного вида комплектующих к общей сумме трат на ПК. Получен-

ные проценты не являются фиксированными и могут незначительно отличаться в зависимости от бюджета и специфических требований.

Данные сведения представлены в таблице 1.

Таблица 1

Тип задачи	Стоимость процессора (в %, от общей суммы)	Стоимость видеокарты (в %, от общей суммы)	Стоимость материнской платы (в %, от общей суммы)	Стоимость оперативной памяти (в %, от общей суммы)	Стоимость остальных комплектующих
Офисные/ мультимедиа	35	0*	15	20	30
Игры	20	35	10	12	23
Работа с графикой/ Моделирование	17	50	10	7	16
Работа с большими объемами вычислений	35	24	14	9	18

*-Для офисных задач отдельная видеокарта обычно не требуется. Для работы с графикой используется видеоадаптер, встроенный в процессор.

Для выбора лучшего из возможных процессоров и видеокарт будут применяться их результаты в нескольких бенчмарках (программах, тестирующих производительность). Все результаты будут складываться и делиться на определенный коэффициент. По итогу получится балл, в соответствии с которым будет происходить определение лучшего. При схожей или равной производительности возможно также сравнение по непосредственным характеристикам, которые наиболее сильно влияют на результат выполнения требуемых задач. К таким характеристикам относятся:

- для процессоров – максимальная частота ядра, количество ядер, техпроцесс, тепловыделение, размер кеша и поддерживаемые инструкции.
- для видеокарт – частота видеопроцессора, объем видеопамати, тип видеопамати, ширина шины.
- для оперативной памяти – частота, объем, стандарт, тайминги.
- для материнских плат – чипсет, возможность разгона процессора, количество слотов для оперативной памяти, видеокарт и других устройств.

Программная система будет учитывать следующие параметры для выбора ПК: диапазон бюджета, назначение (которое в свою очередь будет подразделять на несколько пунктов, таких как игры, работа, мультимедиа и другие), предпочтительный производитель комплектующих, размеры корпуса, объем необходимой памяти и другие.

При внедрении разработанной программной системы предполагается достичь следующих результатов:

- 1) облегчение и автоматизация подбора аппаратных компонентов ПК;
- 2) повышение скорости выполнения задач для сотрудников предприятий и пользователей;
- 3) предоставление данных о самых выгодных ценах в различных магазинах;
- 4) рост экономической выгоды пользователям и компаниям.

После поведения этапа тестирования предполагается этап разработки инструкций пользователя, а затем размещения программной системы в открытом доступе в виде веб-сайта, где любой пользователь сможет самостоятельно выполнить подбор по множеству критериев.

Также существует возможность внедрения программы на различного рода предприятиях. Для этого могут потребоваться незначительные доработки программы под специфические профессиональные задачи фирмы.

УДК 004.415.25; ГРНТИ 20.23.21

РАЗРАБОТКА ПРОГРАММНОГО МОДУЛЯ НА ПЛАТФОРМЕ 1С ДЛЯ РАСЧЕТА КОЛИЧЕСТВА СОТРУДНИКОВ, НЕОБХОДИМОГО ДЛЯ ИСПОЛНЕНИЯ РАБОЧИХ ПРОЦЕССОВ

М.А. Соловов

*Рязанский государственный радиотехнический университет имени В.Ф. Уткина,
Российская Федерация, Рязань, solovov.m@yandex.ru*

Аннотация. В данной статье представлен метод расчета количества необходимых сотрудников для исполнения бизнес-процессов, а также рассмотрена реализация на платформе 1С.

Ключевые слова: процессы, эффективность, сотрудники, разработка, 1С.

THE DEVELOPMENT OF THE SOFTWARE MODULE FOR A CALCULATION OF THE QUANTITY OF NECESSARY EMPLOYEES FOR REALIZATION OF BUSINESS PROCESSES AND ITS REALIZATION ON THE PLATFORM 1C

M.A. Solovov

*Ryazan State Radio Engineering University named after V.F. Utkin,
Russia, Ryazan, solovov.m@yandex.ru*

Abstract. This article presents a method of the calculation of the necessary quantity of the employees for performing business processes as well as reviewed its realization on the platform 1C.

Keywords: processes, efficiency, employees, development, 1C.

Создание эффективно работающей организации или оценка уже имеющейся – это сложный и комплексный процесс, который состоит из множества различных аспектов. Одной из таких задач является расчет необходимого количества сотрудников для выполнения той или иной деятельности.

Представленный ниже способ подходит только в случае, когда деятельность организации можно описать с помощью процессного подхода. Процессный подход это одна из концепций управления, которая окончательно сформировалась в 80-х годах 20 века. В соответствии с ней вся деятельность организации рассматривается как набор процессов. Для того чтобы управлять, необходимо управлять процессами. Данный подход стал одним из ключевых элементов улучшения качества [1].

Важно отметить, что в некоторых ситуациях результаты расчета могут не давать точных результатов. Решение о количестве сотрудников в итоге принимается ответственными лицами.

Первоначальным этапом является определение трудоемкости процесса или группы процессов за период. Обычно за такой период берется 1 год.

Используя формулу (1), необходимо вычислить трудоемкость процесса за период. Предварительно необходимо узнать, сколько по времени выполняется процесс в часах и количество его повторений за период.

$$T_{\text{пп}} = T_{\text{процесса}} * K_{\text{период}}, \quad (1)$$

где $T_{\text{пп}}$ – трудоемкость процесса за период (человек*час);

$T_{\text{процесса}}$ – трудоемкость одного выполнения процесса (человек*час);

$K_{\text{период}}$ – количество повторений за период.

Если есть необходимость рассчитать количество требуемых сотрудников для выполнения блока процессов, то необходимо воспользоваться формулой (2), чтобы рассчитать общую трудоемкость. Часто в блоки объединяются схожие процессы одной области: юридический отдел, отдел кадров и др.

$$T_{\text{блока}} = \sum_{i=0}^n T_{\text{пип}} , \quad (2)$$

где $T_{\text{блока}}$ – трудоемкость процессов одного блока (человек*час);

$T_{\text{пип}}$ – трудоемкость i -го процесса за период (человек*час).

Для оценки соответствия штатной и списочной численностям трудоемкость процесса/процессов блока выражают в условных штатных единицах (УШЕ). Расчет представлен формулой (3).

$$T_{\text{УШЕ}} = \frac{T_{\text{блока}}}{\text{РЧ}_{\text{рд}} * \text{РД}_{\text{сотр}}} , \quad (3)$$

где $T_{\text{УШЕ}}$ – трудоемкость процесса/процессов блока (УШЕ);

$T_{\text{блока}}$ – трудоемкость процесса/процессов одного блока (человек*час);

$\text{РЧ}_{\text{рд}}$ – количество часов в рабочем дне (на одного сотрудника);

$\text{РД}_{\text{сотр}}$ – количество рабочих дней в году (на одного сотрудника).

При сравнении штатной численности отдела с трудоемкостью соответствующего блока следует учитывать, что сотрудники не тратят 100% своего времени на рабочие вопросы и на процессную деятельность. Здесь принимается допущение, что 15% рабочего времени тратится на нерабочие вопросы и задачи [2]. Более того, трудоемкость работы в УШЕ может иметь дробный вид, и потому при сравнении с целочисленным значением штатной численности могут быть сформулированы неоднозначные выводы. Для предотвращения таких ситуаций вводится категория «Количество сотрудников, необходимое для исполнения описанных (оцененных) процессов». Для данной категории необходимо определить долю годовой трудоемкости УШЕ, при которой возникает необходимость в дополнительном исполнителе. Практика показывает, что необходимо привлекать рабочую единицу, если набирается хотя бы 40% трудоемкости УШЕ (данный процент может изменяться в зависимости от объема зарплатного фонда или индивидуальных предпочтений руководства), то есть это 2,72 часа продуктивного рабочего времени в день на выполнение процессов [3]. Расчет производится по формуле (4).

$$K_{\text{СоиВ}} = \begin{cases} \text{ОКРУГЛВНИЗ} \left(\frac{T_{\text{УШЕ}}}{1 - \text{РВ}_{\text{НВ}}}; 0 \right), & \text{если } \left(\frac{T_{\text{УШЕ}}}{(1 - \text{РВ}_{\text{НВ}})}; 0 \right) < 0,4 \\ \text{ОКРУГЛВВЕРХ} \left(\frac{T_{\text{УШЕ}}}{1 - \text{РВ}_{\text{НВ}}}; 0 \right), & \text{если } \left(\frac{T_{\text{УШЕ}}}{(1 - \text{РВ}_{\text{НВ}})}; 0 \right) \geq 0,4 \end{cases} , \quad (4)$$

где $K_{\text{СоиВ}}$ – количество сотрудников, необходимое для исполнения описанных (оцененных) процессов блока (УШЕ);

$T_{\text{УШЕ}}$ – трудоемкость процессов блока (УШЭ);

$\text{РВ}_{\text{НВ}}$ – доля рабочего времени, которое тратится сотрудниками на нерабочие вопросы, в %. При отсутствии эмпирических данных, принимается в размере 15%.

При оценке трудоемкости процессов важно понимать соотношение различных видов трудоемкости и фактической численности сотрудников, исполняющих процессы. Стоит выделить важные аспекты.

1. Сотрудников, занятых исполнением одного из блоков процессов, может быть меньше, чем это предписано штатным расписанием, из-за чего потенциальная трудоемкость подразделения снижается.
2. Совокупный объем работ отражает реальную нагрузку. Когда реальная трудоемкость процессов выше потенциальной трудоемкости списочной численности, сотрудникам приходится работать сверх нормы.

3. Нельзя абсолютно всю деятельность сотрудников описанных (оцененных) процессов представить в виде процессов, так как иногда поступают задачи творческого или проектного характера. Это значит, что оценка трудоемкости описанных процессов будет отличаться в меньшую сторону от совокупного объема работ, исполняемого сотрудниками блока.
4. На основании оценки трудоемкости процессов рассчитывается потребность в сотрудниках, выполняющих соответствующий блок работ. Эта потребность рассчитывается допущением, что 15% рабочего времени работник тратит на нерабочие вопросы. Потребность в новом сотруднике возникает, когда набирается 40% трудоемкости УШЕ.

Выполнение расчета по данному методу может оказаться довольно трудозатратным процессом, особенно если объем данных большой. Без применения программных средств обойтись трудно.

Для решения выше описанной задачи и ей подобных на практике часто используют платформу 1С. Она зарекомендовала себя как многофункциональное надежное программное решение, в котором присутствует большое количество справочных материалов, материалов для обучения. Поэтому принято решение использовать данную платформу в качестве средства разработки.

Первоначально необходимо добавить информацию об организации и имеющимся функциональным блокам работ (рисунок 1 и 2). Это позволит в дальнейшем не вводить информацию каждый раз вручную, а выбирать из готового списка.

Наименование	Форма организ...	Адрес	Руководитель
Главное управление «Региональная энергетическая комиссия» Рязанской области	Главное управление	390025, г. Рязань, ул. Семинарская, д. 33/45	Семенов Николай Иванович
Государственная жилищная инспекция Рязанской области	Государственная инспекция	390006, г. Рязань, ул. Либкнехта, д. 18	Шмидт Геннадий Викторович
Министерство финансов Рязанской области	Министерство	390000, г. Рязань, ул. Почтовая, д. 51	Наумова Марина Александровна

Рис. 1. Справочник «Организации»

- ⊖ { } Блок Функций
 - ⊖ ∞ Значения
 - ∞ Управление Кадрами
 - ∞ Управление Финансовыми Ресурсами
 - ∞ Управление Закупками
 - ∞ Юридическая Поддержка
 - ∞ Управление ИТ

Рис. 2. Перечисление «Блок Функций»

Теперь необходимо внести информацию о процессах, для которых будут проводиться расчеты (рисунок 3). Стоит отметить, что какая-то часть процесса может выполняться в авто-

матическом режиме, поэтому в столбце «р, % (рук.)» необходимо указать процент времени, когда работник выполняет работу непосредственно сам.

Оценка трудоемкости процессов 000000002 от 06.12.2020 13:46:24

Провести и закрыть | Записать | Провести | Печать | Еще ▾

Номер: 000000002 | Дата: 06.12.2020 13:46:24

Наименование организации: Главное управление «Региональная энергетическая комисс»

Функциональный блок: Управление кадрами

Процесс: Коэффициент централизации 000000002 от 02.12.2020 5:49:0

Трудозатраты процесса, часы: 47,33 | Количество повторений в год: 100,00 | Трудозатраты за 1 год, часы: 4 733,00

Добавить | Поиск (Ctrl+F) | Еще ▾

N	Операция / Этап процесса	t, min (рук.)	p, % (рук.)	Время операции	Количество повторений шага для 1 процесса	Трудозатраты (мин)
1	Подготовительная работа	500	50	250	1	250,00
2	Обработка документов	700	20	560	4	2 240,00
3	Согласование	200	25	150	1	150,00
4	Утверждение	200	100	200	1	200,00
Сумма трудозатрат:						2 840,00

Рис. 3. Документ «Оценка Трудоемкости Процессов»

Последним этапом является заполнение документа «Расчет УШЕ» (рисунок 4).

Расчет УШЕ 000000001 от 06.12.2020 14:01:02

Провести и закрыть | Записать | Провести | Еще ▾

Номер: 000000001

Дата: 06.12.2020 14:01:02

Наименование организации: Главное управление «Региональная энергетическая комисс»

Функциональный блок: Управление кадрами

ИтогТрудоемкостей, часы: 5 189,60

Трудоемкость процессов блока (УШЕ): 3,09

Количество сотрудников, необходимое для исполнения описанных процессов блока (УШЕ): 4,00

Добавить | Поиск (Ctrl+F) | Еще ▾

N	Процесс	Трудоемкость процесса, часы
1	Оценка трудоемкости процессов 000000001 от 03.12.2020 0:28:31	456,60
2	Оценка трудоемкости процессов 000000002 от 06.12.2020 13:46:24	4 733,00
Суммарная трудоемкость:		5 189,60

Рис. 4. Документ «РасчетУШЕ»

В дальнейшем планируется, что созданный модуль станет одним из компонентов разработки, которая будет помогать производить оценку эффективности работы организации в ряде аспектов.

Библиографический список

1. Менеджмент качества [Электронный ресурс]. URL - <https://www.kpms.ru/> (дата обращения: 01.03.2021);
2. Зарплата.ру – база актуальных вакансий и резюме [Электронный ресурс]. URL - <https://www.zarplata.ru/> (дата обращения: 01.03.2021);
3. Vouchercloud.com [Электронный ресурс]. URL - <https://www.vouchercloud.com/> (дата обращения: 01.03.2021).

УДК 004.054 ГРНТИ 20.23.19

СРАВНИТЕЛЬНЫЙ АНАЛИЗ ПАРСЕРОВ ИСХОДНОГО КОДА В КОНТЕКСТЕ ВОЗМОЖНОСТЕЙ ОБНАРУЖЕНИЯ ДЕФЕКТОВ В НЕМ

И.Ю. Перехода

*Рязанский государственный радиотехнический университет имени В.Ф. Уткина,
Российская Федерация, Рязань, perehodavanya@gmail.com*

Аннотация. В работе проводится сравнительный анализ двух парсеров. Рассматриваются их достоинства и недостатки, а также возможность использования для обнаружения дефектов исходного кода.

Ключевые слова: синтаксический анализатор, рекурсивный спуск, лексемы, синтаксическое дерево, грамматика.

COMPARATIVE ANALYSIS OF SOURCE CODE PARSERS IN THE CONTEXT OF POSSIBILITIES OF DETECTING DEFECTS IN IT

I.Yu. Perekhoda

The summary. The paper provides a comparative analysis of two parsers. Their advantages and disadvantages are considered, as well as the possibility of using them for detecting defects in the source code.

Keywords: parser, recursive descent, lexemes, syntax tree, grammar.

Синтаксический анализ представляет собой процесс сопоставления линейной последовательности лексем языка с его формальной грамматикой [1]. Результатом синтаксического анализа является синтаксическое строение предложения, представленное либо в виде дерева зависимостей, либо в виде дерева составляющих, либо в виде некоторого сочетания первого и второго способов(см. рисунок 1).

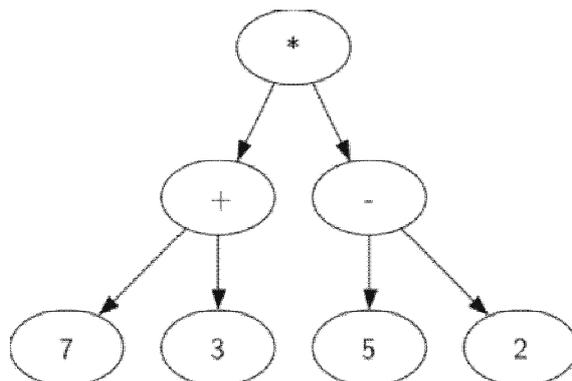


Рис. 1. Синтаксическое дерево выражения $((7 + 3) * (5 - 2))$

Синтаксический анализатор (англ. parser) – это программа или часть программы, выполняющая синтаксический анализ.

К области применения синтаксического анализа, как правило, относятся:

- Языки программирования;
- Структурированные данные (XML, HTML и другие);
- Построение индекса в поисковой системе;
- SQL-запросы;
- Математические выражения;
- Лингвистика;
- Веб-скрейпинг.

Для сравнения были выбраны ANTLR (ANother Tool for Language Recognition) и библиотека PYPARSING для языка Python. Оба инструмента используют метод рекурсивного спуска.

ANTLR представляет собой мощный синтаксический анализатор [2], который можно использовать для чтения, обработки, выполнения или перевода структурированного текста или двоичных файлов. Данный программный продукт широко используется в научных кругах и промышленности для создания языков (Groovy, Jython), инструментов (TwitterSearch) и фреймворков. Данный инструмент используется множеством крупных компаний в своих проектах. К таким компаниям относятся Oracle (SQLDeveloperIDE), LexMachina.

Рассмотрим преимущества и недостатки каждого.

Таблица 1. Преимущества и недостатки ANTLR

Преимущества	Недостатки
<ul style="list-style-type: none"> • Свободный доступ. • Единая нотация. • Применение рекурсивного спуска. • Удобство работы с абстрактным синтаксическим деревом. • Предоставление сообщений об ошибках и восстановление после них. • Наличие визуальных сред разработки. 	<ul style="list-style-type: none"> • Чрезмерная сложность освоения. • Время обработки.

Если с преимуществами данного программного обеспечения всё ясно, то стоит отметить и очень существенный недостаток, который заключается в чрезмерной сложности его освоения, чего не скажешь о библиотеке PYPARSING.

Таблица 2 – преимущества и недостатки библиотеки PYPARSING.

Преимущества	Недостатки
<ul style="list-style-type: none"> • Свободный доступ. • Применение рекурсивного спуска. • Простота установки, освоения и дальнейшей эксплуатации. • Предоставление сообщений об ошибках. 	<ul style="list-style-type: none"> • Малое количество документации и примеров.

Данный программный модуль, разумеется, не сравнится в частоте использования с инструментом, рассмотренным ранее, хотя бы потому, что она ограничивается в применении лишь языком Python, чего не скажешь о ANTLR, который широко применяется для синтаксического анализа в таких языках, как Java, C#, Python, JS и многие другие.

Однако библиотека PYPARSING также используется различными предприятиями в своих коммерческих проектах. Несмотря на свободный доступ, данный модуль предоставля-

ется компаниям для коммерческого использования в пакете подписки Tidelift. Помимо этого, разработка данной библиотеки поддерживается компанией JetBrains [3].

Но, как в случае и с ANTLR, не удалось обойтись без недостатков. Основным недостатком данной библиотеки является почти полное отсутствие документации и примеров.

В качестве примера и оценки разницы использования этих двух инструментов, можно рассмотреть самый элементарный пример синтаксического анализа фразы «Hello, World!».

Исполняющийся код ANTLR:

```
grammar Hello;
r   : 'hello' ID;
ID  : [a-z]+ ;
WS  : [ \t\r\n]+ ->skip ;
```

Однако этот код является лишь небольшим фрагментом, для работы которого необходимо либо описать ещё 2 класса вручную, либо вызвать в командной строке соответствующую команду, которая загрузит в директорию проекта все необходимые файлы.

Для сравнения, в модуле PYPARSING исполняющийся код для синтаксического анализа будет выглядеть следующим образом.

```
from pyparsing import Word, alphas

# define grammar of a greeting
greet = Word(alphas) + "," + Word(alphas) + "!"

hw = "Hello, World!"
print (greet.parseString(hw))
```

Результат работы этого фрагмента будет выглядеть следующим образом:

```
['Hello', ',', 'World', '!']
```

Как видно из примера, для работы этого фрагмента необходимо единожды вызвать библиотеку, в которую уже входит реализация всех необходимых классов, что сильно облегчает использование данного модуля для синтаксического анализа исходного кода, поскольку данный модуль позволит сконцентрировать большее внимание на вычислении различных метрик кода.

Исходя из рассмотренных выше примеров о оценки достоинств и недостатков рассмотренных программных средств, выбор пал на модуль для Python из-за простоты своего освоения, а, следовательно, это облегчит вычисление метрик для анализа программного кода и выявление закономерностей, влияющих на качество программного обеспечения.

Библиографический список

1. Wikipedia, Parsing [Электронный ресурс] // Режим доступа: <https://en.wikipedia.org/wiki/Parsing/>, дата обращения – 24.02.2021
2. AboutTheANTLRParserGenerator [Электронный ресурс] // Режим доступа: <https://www.antlr.org/about.html/>, дата обращения – 26.02.2021.
3. Pyparsing homepage [Электронный ресурс] // Режим доступа: <https://github.com/pyparsing/pyparsing/wiki/>, дата обращения – 28.02.2021.

УДК 004.91 ГРНТИ 50.41.25

МЕТРИКИ ПРОГРАММ НА ЯЗЫКЕ JAVA И ПРОГРАММНЫЕ СРЕДСТВА ДЛЯ ИХ ВЫЧИСЛЕНИЯ

А.В. Горкин

*Рязанский государственный радиотехнический университет имени В.Ф. Уткина,
Российская Федерация, Рязань, avgorkin1@gmail.com*

Аннотация. Сделан обзор на реализации программного обеспечения для вычисления метрик программного кода на языке Java. Выделены их преимущества и недостатки. Предложено собственное решение. Представлена его структура, рассмотрены направления для его дальнейшей доработки.

Ключевые слова: метрики программного кода, программное обеспечение для анализа структур программ.

METRICS OF JAVA PROGRAMS AND SOFTWARE TOOLS FOR IT CALCULATING

A.V. Gorkin

*Ryazan State Radio Engineering University named after V.F. Utkin,
Russia, Ryazan, avgorkin1@gmail.com*

The summary. A review is made on the implementation of software for calculating metrics of program code in the Java language. Their advantages and disadvantages are highlighted. A custom solution is proposed. Its structure is presented and the directions for its further improvement are analyzed.

Keywords: program code metrics, software for analyzing program structures.

Для того, чтобы сравнивать программы, выявлять их проблемные части используются метрики. Разрабатываемое программное обеспечение (ПО) предназначено для анализа метрик программного кода, а так же наглядного представления структуры проекта. Ранее был рассмотрен предполагаемый принцип работы такого приложения [1].

У этого ПО существуют аналоги. Среди них:

1. DependencyFinder [2] – проект с открытым исходным кодом, предназначенный для анализа программ, написанных на языке Java. Одна из утилит проекта позволяет вести подсчет метрик программного кода. К ее недостаткам стоит отнести отсутствие возможности у пользователя вести экспорт метрик в табличный формат, а также невозможность задания пользователем своих собственных ограничений на один из видов метрик.
2. Ohcount [3] – проект с открытым исходным кодом, предназначенный для измерения метрик программного кода на разных языках программирования. Отличительной особенностью программного обеспечения является возможность вести анализ в смешанных файлах, например, HTML документ, в котором также содержится CSS и JavaScript код. Недостатком проекта является узконаправленный подсчет метрик, ориентированный на количество строк программного кода.
3. Metrics [4] – плагин, встраиваемый в среду разработки, позволяющий измерять широкий спектр метрик программного кода на языке Java, а также вести их экспорт в xml формат. Главным недостатком проекта является поддержка только среды разработки Eclipse, популярность которой с каждым годом падает, уступая лидирующую позицию IntelliJ IDEA [5].
4. clock [6] – проект, написанный на Perl для подсчета метрик исходных текстов программ на разных языках программирования. Недостатком решения является узко направленный список метрик, к которым относятся количество пустых строк, количество физических строк кода, а также количество комментариев в тексте программы.

Как можно заметить, у каждого из перечисленных аналогов существуют значительные недостатки. Разрабатываемое решение будет лишено вышеперечисленных недочетов. Для проектирования ПО был выбран язык программирования Java. В ходе разработки получено настольное приложение для следующих ОС: Windows, Linux, MacOS. Основу парсинга исходного кода выполняет библиотека JavaParser [7], главной возможностью которой является распознавание синтаксических структур языка. В качестве сборщика проекта используется фреймворк ApacheMaven [9].

Проект состоит из 2 модулей: core и desktop. Модуль core является независимым и содержит логику анализа исходного кода программ. Такой подход позволит в дальнейшем проще масштабировать ПО. К задачам модуля core относятся:

1. Составление древовидной структуры проекта;
2. Выявление метрик программного кода.

Для удобного анализа синтаксические единицы языка (рисунок 1) и метрики программного кода (рисунок 2) организованы в иерархическую структуру. Это позволяет воспользоваться возможностями полиморфизма.

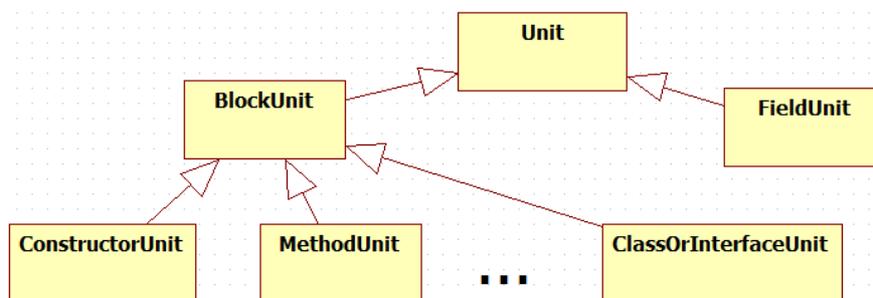


Рис. 1. Иерархия синтаксических конструкций языка

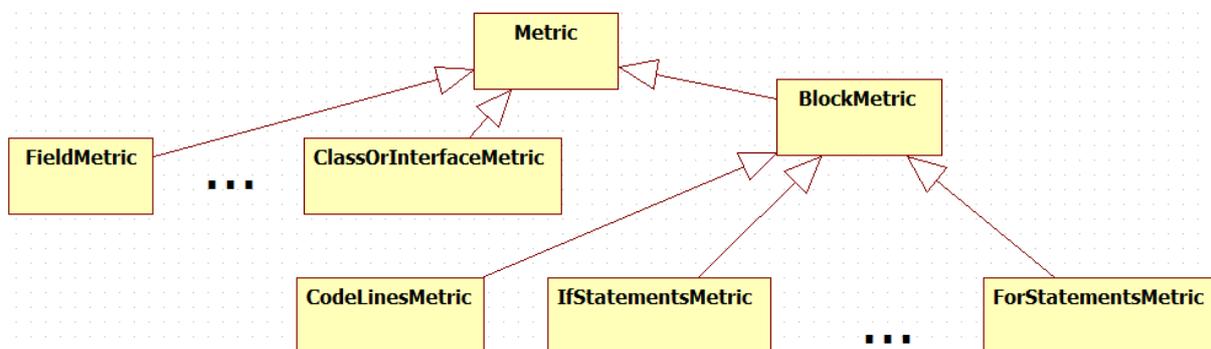


Рис. 2. Иерархия метрик программного кода

Проект представляется в виде совокупности юнитов, вложенных друг в друга. Например, объект класса SourceCodeUnit (аналог файла с исходным кодом) может иметь несколько дочерних экземпляров ClassOrInterfaceUnit-ов (аналог класса/интерфейса), а они в свою очередь – совокупность экземпляров FieldUnit/MethodUnit-ов. В каждом из юнитов определен список метрик. Благодаря такому подходу с помощью вызова всего одного метода у родительского элемента можно рассчитать метрики каждого из вложенных дочерних элементов.

Рассмотрим более подробно взаимодействие классов Unit и Metric. Как ранее говорилось, каждый Unit обладает списком метрик. На этапе построения древовидной структуры проекта каждому Unitобъекту ставится в соответствие ряд «пустых» метрик. В данном контексте словосочетание «пустые метрики» означает, что мы лишь определяем правило нахождения их величин, а сам процесс нахождения откладываем до наступления триггерного со-

бытия. Таким триггерным событием может быть, например, действие пользователя по запуску анализа своего проекта с учетом метрик. Благодаря этому, мы разделяем тяжелый процесс преобразования исходного текста на 2 независимых этапа: составление древовидной структуры проекта и выявление метрик программного кода, что в свою очередь уменьшает время ожидания пользователя при работе с приложением.

Модуль `desktop` зависит от модуля `core` и является одной из возможных реализаций пользовательского интерфейса. В данном случае пользовательский интерфейс реализован с использованием библиотеки `Swing` (Рисунок 3). Пользователю предлагается выбрать папку с проектом, внутри которой находятся файлы с исходным кодом программы (`.java`), после чего формируется список файлов для анализа, который передается модулю `core` для получения древовидной структуры проекта.

The screenshot displays the 'Order' class in an IDE. On the left, a tree view shows the class hierarchy with fields and methods. In the center, a table lists metrics for the class. On the right, the source code of the `Order` class is visible.

Метрика	Величина
Поля	4
Методы	11
Нестатические блоки инициализации	1
Статические блоки инициализации	1
switch выражения	2
if выражения	5
цикл for	1
цикл foreach	0
цикл while	1
цикл do-while	1
Количество строк кода	101

```

public abstract class Order {
    private static final int TEST_CONST = 1;
    private int id;
    private List<OrderEntry> orderEntries;
    private String cashierName;

    {
        if (a == 1) {
            System.out.println("I'm here!");
        }
    }

    static {
        if (a == 1) {
            System.out.println("I'm here!");
        }
    }

    public Order() {
    }

    public Order(int id, List<OrderEntry> orderEntries, String cashierName) {
        this.id = id;
        this.orderEntries = orderEntries;
        this.cashierName = cashierName;
    }

    public int getId() {
        return id;
    }

    public void setId(int id) {
        this.id = id;
    }
}

```

Рис. 3. Пример просмотра метрик класса

На данный момент разработана основная логика программного обеспечения. В дальнейшем планируется добавить функциональность по анализу кода с помощью пользовательских правил, экспорт метрик в табличный формат, а также доработать пользовательский интерфейс.

Стоит отметить, что одним из важных направлений использования ПО является его применение в учебном процессе для анализа программ студентов по дисциплинам, связанным с объектно-ориентированным программированием.

В ходе доклада были освещены следующие темы:

1. Существуют готовые решения по подсчету метрик программного кода, но все они не лишены серьезных недостатков, основными из которых являются ограниченный набор метрик, невозможность переноса статистики в табличный документ, а также отсутствие возможности задавать пользовательские ограничения на значения метрик+6;
2. Разрабатываемое решение состоит из 2 модулей, один из которых отвечает за анализ исходного кода, а второй за пользовательский интерфейс.
3. Основной принцип построения модели метрик и синтаксических конструкций языка – древовидная структура;
4. Графический интерфейс приложения позволяет пользователю выбрать папку с проектом. На основе выбора пользователя формируется список файлов с исходным кодом и передается модулю `core`. Впоследствии пользователю представляется наглядная структура классов, его метрики, а также исходный код.

Разработка пользовательского интерфейса выполняется с помощью библиотеки Swing.

Библиографический список

1. Новые информационные технологии в научных исследованиях. НИТ-2020. Материалы конференции. 2020
2. DependencyFinder. URL: <https://depfind.sourceforge.io/> (дата обращения: 20.02.2021).
3. OhcountgithubURL: <https://github.com/blackducksoftware/ohcount> (дата обращения: 20.02.2021).
4. Metrics. URL: <http://metrics.sourceforge.net/> (дата обращения: 20.02.2021).
5. Java Tools and Technologies Landscape Report. URL: <https://www.jrebel.com/blog/java-trends-and-historical-data> (дата обращения: 20.02.2021).
6. cloc. URL: <https://github.com/AIDanial/cloc>
7. JavaParser: HomeURL: <https://javaparser.org/> (дата обращения: 20.02.2021).
8. Smith N., vanBruggen D., Tomassetti F. JavaParser: Visited. Leanpub, 2019.
9. ApacheMaven. URL: <https://maven.apache.org/> (дата обращения: 20.02.2021)

УДК 004.02; ГРНТИ 50.41.25

ПРОГРАММНАЯ РЕАЛИЗАЦИЯ АЛГОРИТМА ЛЕВЕНБЕРГА-МАРКВАРДТА ДЛЯ ОБУЧЕНИЯ ИСКУССТВЕННЫХ НЕЙРОННЫХ СЕТЕЙ ПРИ АППРОКСИМАЦИИ ПОЛИНОМА В СРЕДЕ МАТЛАВ

А.Н. Пылькин, П.А. Корнев, Е.И. Кокорев

*Рязанский государственный радиотехнический университет имени В.Ф. Уткина,
Российская Федерация, Рязань, goodwin050898@gmail.com*

Аннотация. В машинном обучении применяется ряд математических методов нацеленных на решение задач аппроксимации, как одних из наиболее часто встречающихся классов задач в системах искусственного интеллекта, например, при нахождении экстремума целевой функции, задач приближения с помощью кривых.

Ключевые слова: алгоритм Левенберга-Марквардта, искусственная нейронная сеть (ANN), аппроксимация, метод Гаусса-Ньютона, метод градиентного спуска, метод наименьших квадратов (LS), MATLAB.

SOFTWARE IMPLEMENTATION OF THE LEVENBERG-MARQUARDT ALGORITHM FOR TRAINING ARTIFICIAL NEURAL NETWORKS WHEN APPROXIMATING A POLYNOMIAL IN MATLAB

A.N. Pylkin, P.A. Kornev, E.I. Kokorev

*Ryazan State Radio Engineering University named after V.F. Utkin,
Russia, Ryazan, goodwin050898@gmail.com*

Annotation. In machine learning, a number of mathematical methods are used aimed at solving approximation problems, as one of the most common classes of problems in artificial intelligence systems, for example, when finding the extremum of an objective function, approximation problems using curves.

Keywords: Levenberg-Marquardt algorithm, artificial neural network (ANN), approximation, Gauss-Newton method, gradient descent method, least squares (LS) method, MATLAB.

Для решения таких задач существует множество методов. Один из них метод градиентного спуска и метод Ньютона. Данные методы имеют ряд недостатков: метод градиентного спуска характеризуется малой скоростью схождения при движении по оврагу, а метод Ньютона может не сходиться при определенных условиях.

Находит применение метод Левенберга-Марквардта предназначенный для оптимизации параметров нелинейных регрессионных моделей. Он является дальнейшим развитием метода градиентного спуска и метода Гаусса-Ньютона и устраняет ряд их недостатков.

Метод Левенберга-Марквардта

Метод Левенберга-Марквардта – предназначен для оптимизации параметров нелинейных регрессионных моделей.

Пусть имеется регрессионная выборка – множество пар $D = \{(x_n, y_n)\}_{n=1}^N$ свободной переменной $x \in R$ и зависимой переменной $y \in R$. Регрессионная модель – функция $f(\omega, x_n)$ непрерывно дифференцируемая в области $W \times X$.

Требуется найти такое значение вектора параметров ω , которое бы доставляло локальный минимум функции ошибки:

$$E_D = \sum_{n=1}^N (y_n - f(\omega, x_n))^2;$$

Перед началом работы алгоритма задается начальный вектор параметров ω . На каждом шаге итерации этот вектор заменяется на вектор $\omega + \Delta\omega$. Для оценки приращения $\Delta\omega$ используется линейное приближение функции:

$$f(\omega + \Delta\omega, x) - f(\omega, x) \approx J\Delta\omega,$$

где J — якобиан функции $f(\omega, x_n)$ в точке ω .

Здесь вектор параметров $\omega = [\omega_1, \dots, \omega_R]^T$.

Приращение $\Delta\omega$ в точке ω , доставляющий минимум E_D равно нулю. Дифференцируя $\frac{\partial E_D}{\partial \omega}$, получим[3]:

$$\frac{\partial E_D}{\partial \omega} = (J^T J)\Delta\omega - J^T(y - F(\omega)) = 0.$$

Чтобы найти значение $\Delta\omega$ нужно решить систему линейных уравнений:

$$\Delta\omega = (J^T J)^{-1} J^T (y - f(\omega));$$

Так как число обусловленности матрицы $J^T J$ есть квадрат числа обусловленности матрицы J , то матрица $J^T J$ может оказаться существенно вырожденной. Поэтому Марквардт предложил ввести параметр регуляризации $\lambda \geq 0$ [4],

$$\Delta\omega = (J^T J + \lambda I)^{-1} J^T (y - f(\omega)),$$

где I — единичная матрица. Этот параметр назначается на каждой итерации алгоритма. Если значение ошибки E_D убывает быстро, малое значение λ сводит этот алгоритм к алгоритму Гаусса-Ньютона.

Алгоритм останавливается, в том случае, если приращение $\Delta\omega$ в последующей итерации меньше заданного значения, либо если параметры ω доставляют ошибку E_D меньшую заданной величины. Значение вектора ω на последней итерации считается искомым.

Недостаток алгоритма — значительное увеличение параметра λ при плохой скорости аппроксимации. При этом обращение матрицы $J^T J + \lambda I$ становится бессмысленным. Этот недостаток можно устранить, используя диагональ матрицы $J^T J$ в качестве регуляризации слагаемого:

$$\Delta\omega = (J^T J + \lambda \text{diag}(J^T J))^{-1} J^T (y - f(\omega)).$$

Программная реализация алгоритма Левенберга-Маркварда

Для применения алгоритма Левенберга-Маркварда была построена искусственная нейронная сеть со следующими слоями [5,6]:

1. Один входной слой.
2. Один скрытый слой, содержащий 10 нейронов.

Один выходной слой.

Для получения зависимости вида $T=f(X)$, в качестве входных и выходных данных выступают матрицы X размерностью 2×94 и T размерностью 1×94 .

Для получения данных выборки обучения, был написан простой программный код в среде Matlab (Листинг 1), создавший 2 матрицы, входную $X[1 \times 94]$ и выходную $T[1 \times 94]$. Матрица X заполнена случайными числами от 0 до 99. $T=f(X)$ имеет вид $T = \sqrt{X}$.

```

datain = randi([1 99],1,94);
dataout = zeros(1,94);
for i=1:94
    dataout(i)=roundn(sqrt(datain(i)),-2);
end

```

Листинг 1 «Заполнение входной и выходной матрицы $X[1 \times 94]$ и $T[1 \times 94]$ соответственно»

Для создания ИНС использовалась стандартный пакет расширений среды Matlab – «neural network toolbox». С помощью данного пакета, был сгенерирован программный код (Листинг 2), используя окно настройки искусственной нейронной сети (Рисунок 1).

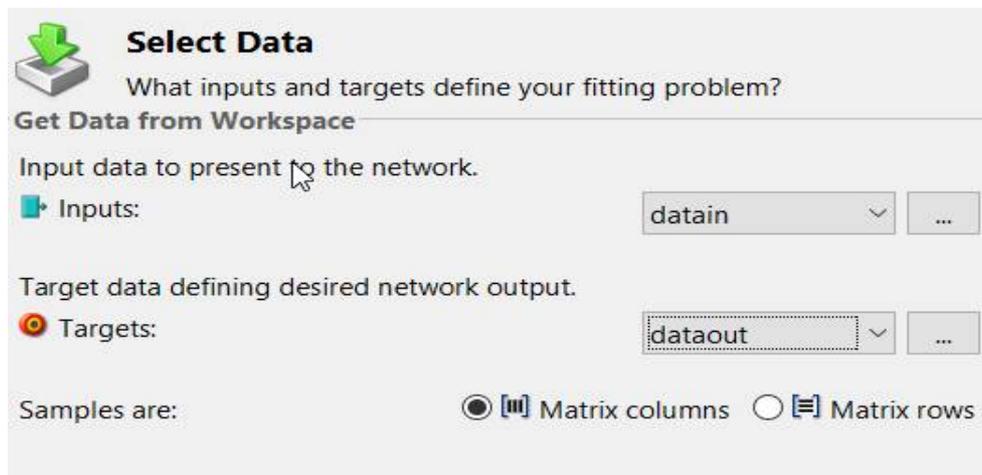


Рис. 1. Окно настройки ИНС, заполнение входными и выходными данными выборку обучения

```
x = datain;
t = dataout;

trainFcn = 'trainlm'; % Levenberg-Marquardt backpropagation.

% Create a Fitting Network
hiddenLayerSize = 10;
net = fitnet(hiddenLayerSize,trainFcn);

% Setup Division of Data for Training, Validation, Testing
net.divideParam.trainRatio = 70/100;
net.divideParam.valRatio = 15/100;
net.divideParam.testRatio = 15/100;

% Train the Network
[net,tr] = train(net,x,t);

% Test the Network
y = net(x);
e = gsubtract(t,y);
performance = perform(net,t,y)

% View the Network
view(net)
```

Листинг 2«Программный код, сгенерированный стандартным пакетом Matlab»

Обучение было завершено за 19 итераций (рисунок 2). Временем обучения можно пренебречь. Функционал ошибки (ФО) брался в виде среднеквадратической ошибки ($\text{performance} = \text{mse}$) и составил $2,90 \times 10^{-6}$. Значение градиента (gradient) составило 0,000929. Значение ФО не улучшалось на вариационной выборке в течение последних 6 эпох обучения, то есть наиболее выгодные значения коэффициентов \mathbf{W} , \mathbf{b} были получены после 13-ой эпохи обучения. Данный элемент обучения можно внимательно исследовать по специальному графику (рисунок 4). Для изучения распределения ошибки по характерным значениям можно воспользоваться гистограммой ошибок (рисунок 3). При построении гистограммы выбираются 20 наиболее часто встречающихся значения разности $e_i = t_i - y_i$. Для приближенного определения зависимости $\mathbf{Y} = \lambda(\mathbf{T})$ применяют анализ графиков линейной регрессии (рисунок 3).

Для правильной оценки вклада всех точек в построение аппроксимационной кривой применяют построение результирующего графика (рисунок 3). Для проверки работы пакета и анализа данных, полученных в результате обучения, был реализован программный код проверки (листинг 3). Данный код сравнивает числа, пропущенные через ИНС, обученную для извлечения квадратного корня, и встроенной в Matlab математической функции по извлечению квадратного корня. Для сравнения был взят интервал целых чисел от 1 до 20. Результат средней ошибки составил 0.0155 (рисунок 4).

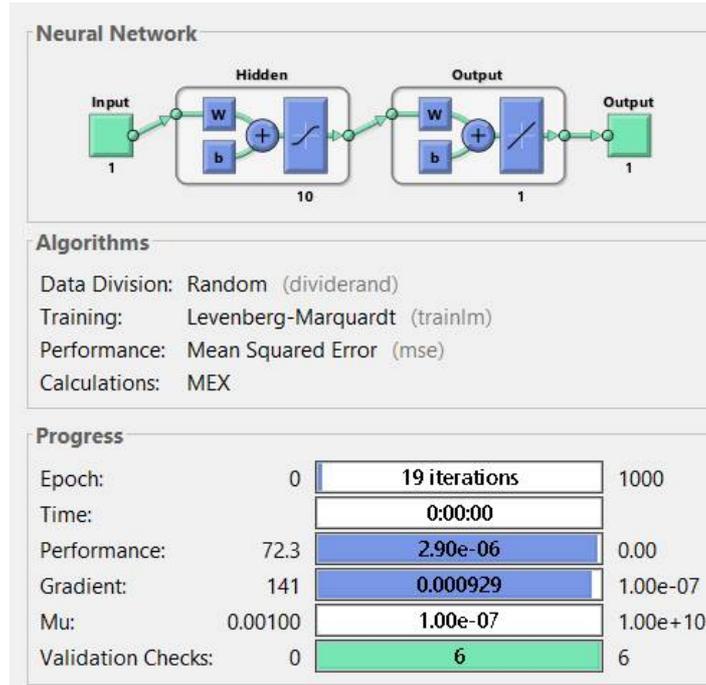


Рис. 2. Результаты алгоритма Левенберга-Марквардта»

```

datavarif = zeros(1,20);
datasqrt = zeros(1,20);
error = zeros(1,20);
average=0;
for i=1:20
    datavarif(i) = sim(net,[i]);
    datasqrt(i) = sqrt(i);
    error(i)=datavarif(i)-datasqrt(i);
    average = average+ error(i);
end
average=average/20;
i=1:1:20;
plot(i,datavarif,i,datasqrt)

```

Листинг 3 «Программный код проверки данных ИНС»

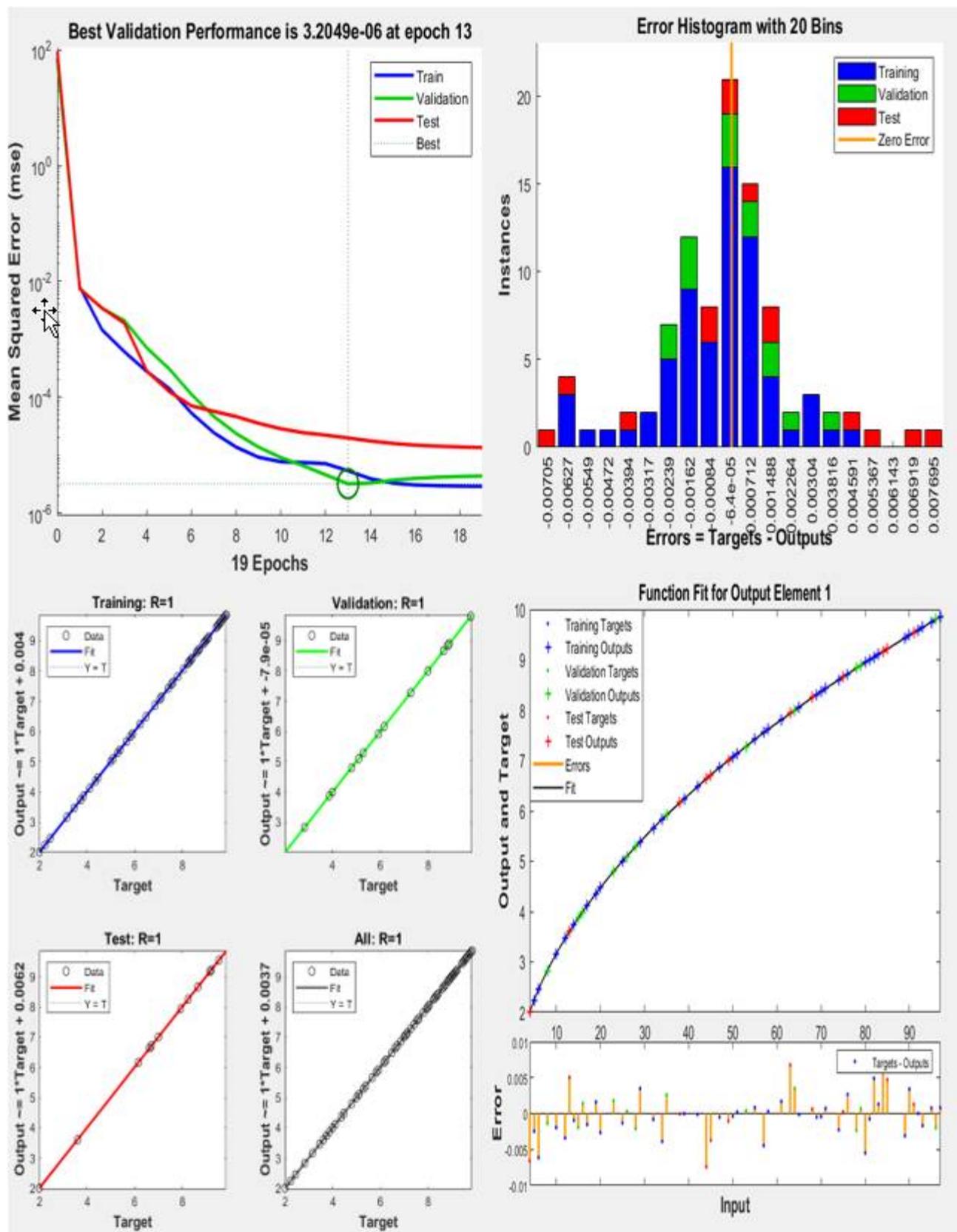


Рис. 3. Построение результирующего графика: 1-Определение наилучшего по валидации ФО, 2- Гистограмма ошибок ИНС, 3-График линейной регрессии, 4- График аппроксимации зависимости

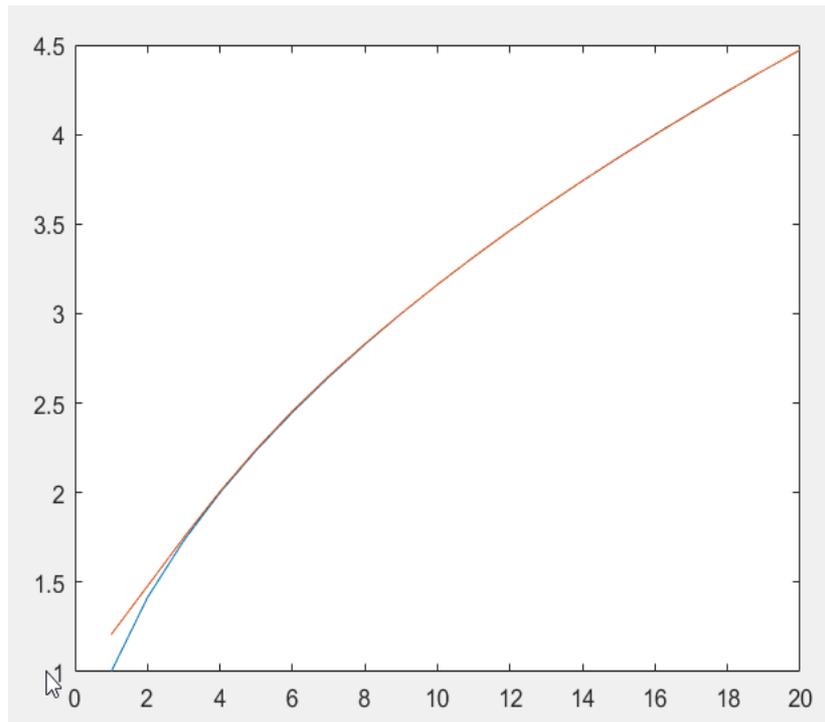


Рис. 4. Сравнение выходных данных ИНС и встроенной функции извлечения квадратного корня Matlab

Библиографический список

1. Н.И.Глебов, Ю.А.Кочетов, А.В.Плясунов. Методы оптимизации. Учебное пособие Новосибирск: Новосибирский государственный университет, 2000. 105 с.
2. Hastie T., Tibshirani R., Friedman J. The Elements of Statistical Learning. Springer, 2014. — 739 p
3. Демиденко, Е.З. Оптимизация и регрессия. Издательство: М.: Наука: 1989 г. 296с.
4. Пархоменко С.С., Леднёва Т.М. Обучение нейронных сетей методом Левенберга-Марквардта в условиях большого количества данных. // Вестник ВГУ, Серия: системный анализ и информационные технологии, 2014, №2. С. 98-106.
5. Хайкин Саймон. Нейронные сети М.: ООО «И.Д. Вильямс», 2019. 1104с.
6. Осовский С. Нейронные сети для обработки информации / Пер. с польского И.Д. Рудинского. - М.: Финансы и статистика, 2002. - 344 с.
7. Корнев П.А. Автоматизация поиска оптимальных параметров искусственных нейронных сетей // Научные разработки: евразийский регион: материалы международной научной конференции теоретических и прикладных разработок (г. Москва, 29 декабря 2018 г.). / отв. ред. Д.Р. Хисматуллин. Москва: Издательство Инфинити, 2018. С. 182-187.
8. Корнев П.А. Основы поиска и настройки оптимальных параметров искусственных нейронных сетей // НАУЧНО-ПРАКТИЧЕСКИЕ ИССЛЕДОВАНИЯ. 2019. № 8.7 (23). С. 34-37.
9. Корнев П.А. Исследование параметров оптимальности нейросетевого решения в задачах классификации // Системы управления и информационные технологии, №1(79), 2020. С. 36-39.
10. P.A. Kornev, A.N. Pylkin. Research of the Methods of Optimization of Artificial Neural Networks Training in the Tasks of Regression Analysis // 2nd International Conference on Control Systems, Mathematical Modeling, Automation and Energy Efficiency (SUMMA), Lipetsk, Russia, 2020, pp. 365-369, doi: 10.1109/SUMMA50634.2020.9280625.
11. P.A. Kornev, A.N. Pylkin. Fundamentals of optimization of training algorithms for artificial neural networks // E3S Web Conf. 224 01022 (2020). DOI: 10.1051/e3sconf/202022401022.
12. А.В. Шолохов, П.А. Корнев, А.Н. Пылькин Обзор метода сопряженных градиентов и его основных модификаций для решения нейросетевых задач // Современные технологии в науке и образовании – СТНО-2020: сб. тр. III междунар. науч.-техн. форума: в 10 т. Т.4./ под общ. ред. О.В. Миловзорова. – Рязань: Рязан. гос. радиотехн. ун-т, 2020; Рязань. С.191-194
13. А.Н. Пылькин, Е.И. Кокорев, В.Г. Макаров Использование алгоритма Левенберга-Марквардта при обучении искусственных нейронных сетей для решения задач аппроксимации // Новые информационные технологии в научных исследованиях: материалы XXV Юбилейной Всероссийской научно-технической конференции студентов, молодых ученых и специалистов; Рязань: ИП Коняхин А.В. (Book Jet), 2020.С. 40-43

УДК 004.04; ГРНТИ 50.41.25

РОЛЬ ВЫЧИСЛИТЕЛЬНЫХ СИСТЕМ В ОПРЕДЕЛЕНИИ ПОТРЕБНОСТИ ЗАКАЗА АПТЕЧНОЙ СЕТИ

Д.А. Фролов

*Рязанский государственный радиотехнический университет имени В.Ф. Уткина,
Российская Федерация, Рязань, Denfroal@yandex.ru*

Аннотация. В работе рассматривается роль, которую играют современные вычислительные системы для улучшения аптечного бизнеса. В том числе рассматривается конкретный алгоритм и ключевые аспекты, необходимые для расчета оптимального ассортимента лекарственных препаратов в аптеке.

Ключевые слова: ассортиментный план, система поддержки принятия решения, параметр расчета потребности автозаказа.

THE ROLE OF COMPUTER SYSTEMS IN DETERMINING THE NEED OF ORDERING A PHARMACY NETWORK

D.A. Frolov

*Ryazan State Radio Engineering University named after V.F. Utkin,
Russia, Ryazan, Denfroal@yandex.ru*

The summary. The paper examines the role that modern computing systems play to improve the pharmacy business. This includes considering a specific algorithm and key aspects necessary to calculate the optimal range of drugs in a pharmacy.

Keywords: assortment plan, decision support system, parameter for calculating ADR requirement.

Управление – в широком, кибернетическом смысле — это обобщение приемов и методов, накопленных разными науками об управлении искусственными объектами и живыми организмами. Язык управления – это использование понятий «объект», «среда», «обратная связь», «алгоритм» и т. д. [1].

Каждой организации необходимо упростить процесс обработки информации, а также стремиться уменьшить затраты времени, материальных и трудовых ресурсов в ходе своей деятельности. Решения данных задач можно достичь с помощью автоматизированных вычислительных систем, которые играют важную роль в современных бизнес-процессах. Наличие положительного эффекта от грамотного применения автоматизированных систем управления в аптечных организациях доказано уже не раз, в условиях же нарастающего экономического кризиса информационные технологии могут стать серьезным инструментом для оптимизации управления и сокращения расходов в секторе фармацевтического ритейла [2].

Аптечный ассортимент - набор разновидностей аптечных товаров по наименованиям, маркам, размерам и типам. Оптимальным ассортиментом считается набор лекарственных препаратов и других аптечных товаров, удовлетворяющих реальным потребностям с максимально полезным эффектом для потребителей при минимальных затратах на их разработку производства и доведения до потребителей. Комплексная система мер по определению набора товарных групп, наиболее предпочтительных для успешной работы на рынке, называется ассортиментной политикой аптечной сети.

Основными задачами ассортиментной политики являются:

- увеличение количества новых покупателей;
- удовлетворение запросов потребителей;
- оптимизация экономических и финансовых результатов предприятия.

Определение потребности в лекарственных препаратах и предметах медицинского назначения непосредственно связано с их товарными характеристиками, с фактическим их потреблением и выявлением закономерностей изменения спроса на них. При определении потребности решаются два основных вопроса:

1. В каком объёме и что именно должно быть предложено из большого множества различных товаров и услуг.
2. Для кого необходимы эти товары.
3. Насколько они будут востребованы. Основными факторами, влияющими на формирование спроса и потребления на лекарственные препараты, являются следующие:
 - рост численности населения и повышение их материального благосостояния;
 - расширение сети аптечных организаций и связанное с этим увеличение численности аптечных кадров;
 - развитие фармацевтической науки, использование новых методов ранней диагностики, а также внедрение более эффективных лекарственных препаратов.

Описание предметной области

Управляющий аптечной сети каждую неделю закупает лекарственные препараты в свои аптеки для дальнейшей их продажи. Для этого формируется автозаказ, в рамках которого, определяется количество закупаемого товара. Данная система позволяет определить в каком количестве необходимо поставлять лекарственные препараты в аптеки для получения максимальной прибыли управляющему. Пользователю необходимо ввести данные по продажам конкретной аптеки за прошедшую неделю, проанализировав которые, система будет производить расчет потребности автозаказа и предлагать оптимальный вариант изменения параметра расчета потребности для закупки товаров у поставщика.

Основные проблемы принятия решений

Человек (управляющий аптечной сетью) не всегда может точно определить в каком количестве закупать лекарственные препараты в аптеку, так как товар может не пользоваться спросом в несезонный период, залеживаться и терять срок годности, или же наоборот, продаваться активно. Может происходить нехватка товара и отказы на кассе, что в конечном счете пагубно влияет на развитие аптечной сети и получение максимальной прибыли. В результате анализа система должна вычислять и выводить параметр расчета потребности (РП), который показывает на сколько нужно изменить товарный запас аптеки, чтобы в оптимальном количестве закупить товар у поставщика.

Описание процесса принятия решения

На первоначальном этапе происходит анализ ситуации, т. е. анализ продаж лекарственных препаратов за предыдущую неделю. Пользователь вводит данные, и, основываясь на них, система выявляет наиболее проблемные места продаж аптечной сети и производит расчет потребности автозаказа, после чего предлагает оптимальный вариант закупки товара у поставщика. Далее происходит согласование решения данной системы и управляющего аптечной сетью. Получив согласие руководителя, сотрудник вводит полученные от системы результаты в специальную программу («Трейд Фарм» или др.), связанную с аптечной деятельностью, которая в конечном счете формирует количество и конкретные названия лекарственных препаратов на следующий срок. Данные доходят до поставщика, после чего товар доставляется в аптеку, где происходит его дальнейшая продажа. Позже проводится проверка фактических результатов с теми, которые руководитель надеялся получить.

Входные параметры расчетно-логической системы:

1. Количество позиций (лекарственных препаратов).
2. Количество отказов ассортиментного плана на кассе в аптеке.
3. Стойкая дефектура – количество позиций, которых не было в наличии весь период продажи.
4. Количество позиций, которые уходили в дефектуру.

5. Количество дней дефектуры.
6. Оборачиваемость товара.

Также в ходе анализа СППР будет использовать следующие вычисляемые параметры:

- Процент дефектуры (ПД), который равен отношению количества позиций, которые уходили в дефектуру к количеству позиций, находящихся в АП аптеки:

$$\text{ПД} = \text{Кол-во ушло в дефект} / \text{АП} * 100\%.$$

- Коэффициент дефектуры (КД), который равен отношению количества дней дефектуры к позициям, которые уходили в дефектуру:

$$\text{КД} = \text{Кол-во дней дефект} / \text{Кол-во ушло в дефект}.$$

Данные входные параметры будут использоваться для вычисления параметра расчета потребности, который покажет на сколько нужно изменить размер автозаказа для оптимальной закупки лекарственных препаратов у поставщика для дальнейшей их продажи в аптеке.

Выходным параметром для данной СППР будет являться параметр расчета потребности (РП) - дробное число, которое покажет на сколько дней необходимо изменить товарный запас, для закупки товара у поставщика для получения максимальной прибыли. Схема принятия решения представлена на рисунке 1.

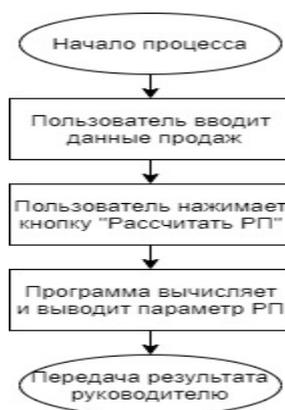


Рис. 1. Схема принятия решения

Тестирование работы на примере аптеки с входными параметрами:

- Кол-во позиций АП – 567
- Кол-во отказов на кассе – 23
- Стойкая дефектура – 9
- Кол-во дней дефектуры – 15
- Позиций ушло в дефектуру – 98
- Оборачиваемость – 51

После ввода входных параметров пользователь нажимает кнопку «Рассчитать РП». После этого программа производит вычисление еще двух параметров: процента дефектуры (ПД) и коэффициента дефектуры (КД) данной аптеки и анализирует данные продаж путем сравнения результатов работы на прошлой неделе с текущими, по данным параметрам. Расчет производится в два этапа, сначала анализируется числитель, затем знаменатель параметра расчета потребности (РП). Каждый параметр имеет свой приоритет важности, который будет влиять на расчет числителя и знаменателя, а именно его увеличение или уменьшения. Если расчет оказался положительным числитель или знаменатель параметра РП увеличивается на 1, отрицательным – уменьшается на 1, иначе при расчёте 0, он остается неизменным.

Приоритеты важности параметров:

- кол-во отказов на кассе – 0,5;
- процент дефектуры – 1;
- стойкая дефектура– 0,3;
- кол-во дней дефектуры – 1;
- оборачиваемость АП – 1,5;
- позиций уходило в дефектуру – 0,3;
- коэффициент дефектуры – 1.

В нашем случае аптека имела следующие показатели за предыдущую неделю:

- кол-во позиций АП – 540;
- кол-во отказов на кассе – 5;
- стойкая дефектура – 7;
- кол-во дней дефектуры – 18;
- позиций уходило в дефектуру – 115;
- оборачиваемость – 78;
- запас аптеки для автозаказа – 7/15.

Рассчитанные параметры:

$$КД= 15/98=0,15$$

$$ПД=134/567=23\%$$

В данной ситуации произошло увеличение показателя стойкой дефектуры, тогда как позиции, которые уходили в дефектуру уменьшились, следовательно расчет 0. Но при расчете КД оказался равным 0,15, это меньше нормы (1,5-1,7), поэтому расчет +1, следовательно в совокупности расчет оказался равным $1+0=1$, положительным, поэтому произошло увеличение числителя параметра РП на 1. Далее происходит анализ знаменателя. Показатель количество отказов на кассе существенно увеличился, поэтому расчет +0,5. Далее, произошло увеличение количества дней дефектуры, поэтому расчет -1. Процент дефектуры > 10, следовательно расчет -1. Оборачиваемость больше нормы (>40), значит расчет -1,5. Итоговый расчет знаменателя равен $0,5-1-1-1,5=-3$, следовательно знаменатель параметра РП уменьшится на 1.

Таким образом, при заданных параметрах, коэффициент запаса рассматриваемой аптеки должен равняться 8/12.

Библиографический список

1. Меньшов А.В. Теоретические основы автоматизированного управления: Учеб. пособие. -- М.: МГУП, 2002.
2. Кольчугин А., Музылева М. Автоматизация аптечного бизнеса: что, когда и зачем? // «Российские аптеки», №1-2, 2009.

УДК 004.05; ГРНТИ 50.49.37

ПРИНЦИПЫ ПОСТРОЕНИЯ АВТОМАТИЗИРОВАННОЙ ИНФОРМАЦИОННОЙ СИСТЕМЫ КОНТРОЛЯ И УПРАВЛЕНИЯ ДОСТУПОМ НА ПРЕДПРИЯТИИ

Е.А. Тараненко

*Рязанский государственный радиотехнический университет имени В.Ф. Уткина,
Российская Федерация, Рязань, taranka1991@gmail.com*

Аннотация. В работе рассматриваются виды системы контроля и управления доступом (СКУД). Общие понятия о системе. Приводятся их особенности, плюсы и минусы системы.

Ключевые слова: система контроля и управления доступом(СКУД), автоматизированная система управления.

PRINCIPLES OF BUILDING AN AUTOMATED INFORMATION SYSTEM FOR CONTROL AND ACCESS CONTROL AT THE ENTERPRISE

Е.А.Тараненко

*Ryazan State Radio Engineering University named after V.F. Utkin,
Russia, Ryazan, taranka1991@gmail.com*

The summary. The paper discusses the types of access control systems (ACS). General concept of the system. Their features, pros and cons of the system are given.

Keywords: access control system (ACS), automated control system.

В настоящее время все больше актуальности набирает вопрос обеспечения безопасного функционирования жилых домов, производства и т.п. К основным требованиям, которые предъявляются к системам безопасности и контроля доступа, следует относить возможно высокую эффективность и независимость от человеческого фактора, что ориентирует проектировщиков, производителей, разработчиков и инсталляторов на повышение качества систем.

В данной статье будут рассмотрены основные типы систем, их преимущества и недостатки, а также приведен пример внедрения системы.

Понятие системы контроля доступом

Система контроля и управление доступом (СКУД) – это объединённые в комплексы механические, электронные, аппаратно-программные и другие средства, обеспечивающие доступ людей в определённые зоны [1].

СКУД обычно состоит из серверов СКУД – обычных компьютеров, которые управляют подключенными к ним контроллерами СКУД. Контроллер – это центр системы, именно в нем хранится информация о конфигурации, режимах работы системы, список людей, у которых есть доступ к этому ресурсу, а также особые привилегии доступа к этому ресурсу. Когда система проста, то контроллер может быть встроен в считыватель, замок, турникет, дверь и др.

Следующей важной частью СКУД является устройство – считыватель, который можно подключить к контроллеру. Он представляет собой устройство, которое помогает считывать информацию, записанную на пропуске, карточке или другом оборудовании. Эту информацию считыватель передает контроллеру, которые принимает решение о допуске человека. Есть техническая возможность, чтобы считыватель спрашивал подтверждения решения у контроллера.

У каждого считывателя есть ответная часть – идентификатор, который содержит информацию, с помощью которой происходит идентификация человека. Каждому идентификатору прописан какой-то уровень доступа, в соответствии с которым пользователь имеет возможность получить доступ к тому или иному ресурсу на определенное время.

Типы систем контроля и управления доступом

Автономный СКУД управляет одним или несколькими точками доступа, при этом нет передачи информации на центральный пульт и контроля со стороны оператора. Управляющий компьютер при этом не используется. Чаще всего это электронные замки для ограничения доступа к помещению.

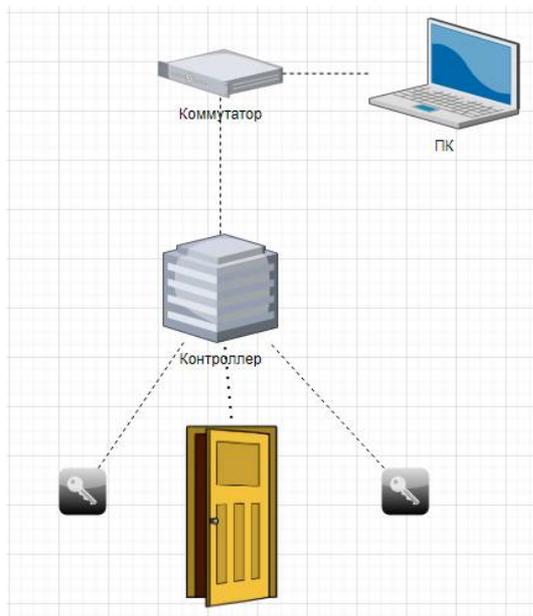


Рис. 1. Пример автономной СКУД

К достоинствам таких систем можно отнести быстрое и легкое удаление ключа из базы данных при потере пропуска или увольнении сотрудника и т.п.

Недостатки: процесс очень неудобен для программирования, когда количество дверей от четырех и более, а пользователей более сорока. Также при этом отсутствует возможность оперативного воздействия на процесс прохода.

Сетевые (централизованные) СКУД имеют центральный контроллер в отличие от автономной СКУД. Для построения сетевой СКУД требуется прокладка кабельных линий, обеспечивающих информационную связь контроллеров. Если использовать для связи контроллеров Ethernet, то будет возможность задействовать для всех цепей на объекте компьютерную сеть. На рисунке 1 показан пример контроллеров доступа с дверьми и серверами управления СКУД.

Это большой класс СКУД, основная особенность которых заключается в том, что имеется возможность конфигурирования аппаратуры и управления процессом доступа с компьютерных терминалов. Разные сетевые СКУД имеют свои индивидуальные особенности и различаются по архитектуре, масштабу, возможностям, количеству управляющих компьютеров, типу применяемых считывателей, уровням устойчивости к взлому и электромагнитным воздействиям.

Несмотря на совершенную архитектуру, имеются следующие существенные недостатки этих систем:

- 1) высокая цена оборудования (необходима покупка серверов, у которых должна быть большая вычислительная мощность);
- 2) дорогостоящее обслуживание, необходимое для оборудования;
- 3) отсутствие возможности работы с системой, находясь вне той же организации;
- 4) необходимость установки лицензированной системы, которая так же дорогостояща.

Разработка архитектуры является главной частью этапа разработки системы. Разрабатываемая система представляет собой службу для нормального количества пользователей

(примерно 200). Разрабатываемая СКУД относится к среднему типу, имеет много точек доступа и около сотни пользователей.

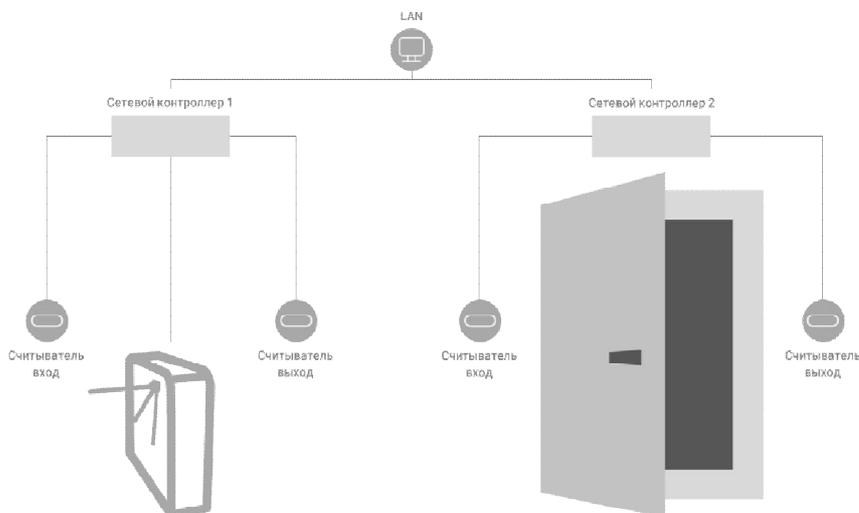


Рис. 1. Пример сетевой СКУД

<https://securityevolution.com.ua/sistemy-kontrolya-dostupa/>

Универсальные СКУД совмещает в себе оба типа – сетевой и автономный. Могут работать в сетевом режиме под контролем центрального устройства управления, переходящим в автономный режим, когда возникают поломки сетевого оборудования или обрыв связи.

Пример внедрения СКУД на предприятие

Характеристика СКУД предприятия на данный момент. Имеется двухэтажное здание с цокольным этажом. На данный момент на предприятии пропускной режим реализуется на основе пластиковых магнитных карт, которые прикладываются к считывателю [2].

При существующей организации СКУД есть риски утечки или утраты информации при несанкционированном доступе в помещение, а также потере магнитной карты. Человеческий фактор в данной системе может повлиять на работу охранника. Возможное дублирование информации в системе увеличивает нагрузку на нее. Отсюда вытекает основная цель - улучшение системы контроля и управления доступом с помощью внедрения биометрии и устранения дублирования информации.

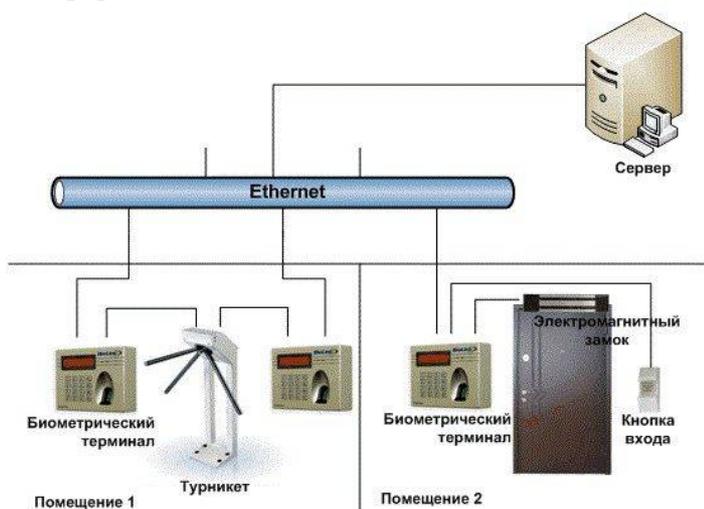


Рис. 2. Простейшая схема биометрической системы

Будущие СКУД

Архитектура СКУД стала строиться на основе «умных» контроллеров, они будут сами взаимодействовать между собой и иметь несколько мастеров-контроллеров, выполняющих роль сервера. Они также могут объединяться на базе единого сервера.

Библиографический список

1. Крахмалев А.К. Средства и системы контроля и управления доступом. Учебное пособие. М.: НИЦ «Охрана» ГУВО МВД России. 2003.
2. РД Гостехкомиссии России «Автоматизированные системы. Защита от несанкционированного доступа к информации. Классификация автоматизированных систем и требования по защите информации». – М.: ГТК РФ, 1992.

ИСПОЛЬЗОВАНИЕ DOCKER ДЛЯ РАЗВЕРТЫВАНИЯ WEB ПРИЛОЖЕНИЯ

Н.В. Всемиров, А.Н. Пылькин

*Рязанский государственный радиотехнический университет имени В.Ф. Уткина,
Российская Федерация, Рязань, vsemir4ik@mail.ru*

Аннотация. В работе рассматриваются действия, достаточные для развертывания web приложения.

Ключевые слова: Docker, web приложения.

USING DOCKER TO DEPLOY A WEB APPLICATION

N.V. Vsemirov, A.N. Pylkin

*Ryazan State Radio Engineering University named after V.F. Utkin,
Russia, Ryazan, vsemir4ik@mail.ru*

The summary. The work considers the steps sufficient for deploying a web application.

Keywords: Docker, web applications.

Актуальность данной технологии заключается в потребности специалистов в удобном инструменте для ввода в эксплуатацию web приложений и их поддержке.

Docker — программное обеспечение с открытым исходным кодом, применяемое для разработки, тестирования, доставки и запуска веб-приложений в средах с поддержкой контейнеризации. Он нужен для более эффективного использования системы и ресурсов, быстрого развертывания готовых программных продуктов, а также для их масштабирования и переноса в другие среды с гарантированным сохранением стабильной работы.

Docker упаковывает ПО в стандартизированные блоки, которые называются контейнерами. Каждый контейнер включает все необходимое для работы приложения: библиотеки, системные инструменты, код и среду исполнения.

Используемые термины:

Images (образы) - схемы приложения, которые являются основой контейнеров. Они состоят из слоев, которые Docker комбинирует в один образ при помощи вспомогательной файловой системы UnionFS. Так решается проблема нерационального использования дисковой памяти. Параметры образа определяются в Docker-file.

Docker-file - это простой текстовый файл, в котором содержится список команд Docker-клиента. Эти команды идентичны командам Linux.

Containers (контейнеры) - создаются на основе образа и запускают само приложение. Список запущенных контейнеров можно увидеть с помощью команды *docker ps*. Контейнеры запускаются напрямую из ядра операционной системы Linux. Благодаря этому, они потребляют гораздо меньше ресурсов, чем при аппаратной виртуализации. Изоляция рабочей среды осуществляется при помощи технологии **namespace**. Для каждого изолированного

пространства (контейнера) создается уникальное пространство имен, которое и обеспечивает к нему доступ. Любой процесс, выполняемый внутри контейнера, ограничивается namespace.

Docker Daemon - фоновый сервис, запущенный на хост-машине, который отвечает за создание, запуск и уничтожение Докер-контейнеров. Демон — это процесс, который запущен на операционной системе, с которой взаимодействует клиент.

Docker Client - утилита командной строки, которая позволяет пользователю взаимодействовать с демоном. Существуют другие формы клиента, например, Kitematic, с графическим интерфейсом.

Docker Hub - регистр Докер-образов. То есть архив всех доступных образов. Если нужно, то можно содержать собственный регистр и использовать его для получения образов.

Рассмотрим, как настроить и запустить web приложение с помощью утилиты Docker compose. Он применяется не во всех случаях. Если проект является простым приложением, не требующим использования сторонних сервисов, то для его развертывания можно ограничиться только Docker. Docker Compose рекомендуется использовать при проектировании сложных программных продуктов, включающих в себя множество процессов и сервисов. В файле `docker-compose.yml` настраиваются контейнеры, которые надо создать для проекта, потом они будут созданы автоматически с помощью `docker-compose`. Файл использует синтаксис YAML и должен содержать такие данные

version: 'версия'

networks:

сет

volumes:

хранилища

services:

контейнеры

Создадим папку `docker-project` и создадим в ней файл `docker-compose.yml`:

`mkdir docker-project`

`vim docker-project/docker-compose.yml`

Например:

`version:`

'3.5'

`services:`

Рассмотрим содержимое настройки контейнера:

имя_контейнера:

image: образ:версия

Чтобы запустить группу контейнеров, настроенную в `docker-compose.yml` необходимо перейти в папку, где находится этот файл конфигурации и выполнить там команду

docker-compose up

Чтобы остановить контейнеры достаточно нажать `Ctrl+C`.

Команда `down` не просто останавливает все запущенные контейнеры, но и удаляет их:

`docker-compose down`

С помощью `docker` можно пробросить порт контейнера в основную операционную систему и получить к нему доступ. Для этого используется директива `ports`. Синтаксис такой:

ports:

- внешний_порт:внутренний_порт

Например:

`ports: - "8095:80"`

Для монтирования хранилищ или внешних папок хоста используется пункт `volumes`. Синтаксис очень похож на работу с портами:

volumes:

- /путь/к/внешней/папке:/путь/к/внутренней/папке

К контейнеру можно монтировать хранилища, создаваемые в docker.

volumes: project-vl:

Например :

docker-mysql:

image: mysql

volumes: - project-vl:/var/lib/mysql

environment:

- MYSQL_ROOT_PASSWORD=password

- MYSQL_DATABASE=database

- MYSQL_USER=user

- MYSQL_PASSWORD=password

Для взаимодействия между контейнерами используются виртуальные сети, они добавляются похожим образом на хранилища.

Сначала необходимо добавить сеть в глобальную секцию networks

networks: project-network:

Затем для каждого контейнера, которые будут иметь доступ к этой сети, надо добавить сеть в раздел networks.

Docker-phpmyadmin:

image: phpmyadmin/phpmyadmin:latest

ports: - "8095:80"

environment: - PMA_HOST=docker-mysql

networks: - project-network

Таким образом данная технология позволяет автоматизировать процесс ввода в эксплуатацию сложных web приложений, а также снизить трудозатраты на поддержку работоспособности инфраструктуры приложения. Docker является важным инструментом для каждого современного разработчика, как основа аппаратной виртуализации приложений. Эта технология обладает широким функционалом и возможностями для контроля процессов. Docker позволяет не только развертывать контейнеры, но и оперативно масштабировать их экземпляры, работать с многоконтейнерными приложениями, а также объединять несколько Docker-хостов в единый кластер. Docker характеризуется достаточно простым синтаксисом. Поэтому он довольно прост в освоении. Программное обеспечение совместимо со всеми версиями операционных систем Linux и Windows, поэтому область применения Docker практически не ограничена.

Библиографический список

- 1 Docker.com URL: <https://www.docker.com/>
- 2 Docker Hub URL: <https://hub.docker.com/>

УДК 519.171.1; ГРНТИ 50.41.25

ПРИМЕНЕНИЕ ГРАФА ДЛЯ РЕШЕНИЯ ЗАДАЧ ПОИСКА НАИБОЛЕЕ ПОПУЛЯРНЫХ АВТОРОВ КНИГ В СОЦИАЛЬНОЙ СЕТИ

Д.В. Новинская

*Рязанский государственный радиотехнический университет имени В.Ф. Уткина,
Российская Федерация, Рязань, vampi-reska@mail.ru*

Аннотация. В работе рассматриваются социальные сети, теория графов применительно к социальным сетям. Приводятся некоторые виды графов и решаемые ими задачи.

Ключевые слова: социальная сеть, актор, теория графов, социальный граф, граф интересов.

APPLICATION OF THE GRAPH TO SOLVE THE PROBLEMS OF SEARCHING THE MOST POPULAR AUTHORS OF BOOKS ON THE SOCIAL NETWORK

D.V. Novinskaya

*Ryazan State Radio Engineering University named after V.F. Utkin,
Russia, Ryazan, vampi-reska@mail.ru*

The summary. The work deals with social networks, graph theory as applied to social networks. Some types of graphs and the problems they solve are presented.

Keywords: social network, actor, graph theory, social graph, graph of interests.

Интеллектуальный анализ данных (data mining) - это исследование данных, в котором используются методы искусственного интеллекта для принятия решений в различных сферах деятельности человека.

Стоит отметить, что в data mining существует отдельное направление - social mining, в котором применяют алгоритмы и методы интеллектуального анализа данных для поиска и обнаружения зависимостей и знаний в социальных сетях или в тех областях, где данные можно представить в виде графов или сетей. Области применения social mining достаточно обширны.

Социальные сети являются отличным источником данных и важно уметь правильно с ними работать. Наиболее часто для анализа и визуализации в social mining используются графы, в которых узлами (актерами) являются люди или группы людей, а ребрами - взаимоотношения (связи) или потоки информации между актерами.

Актор - это социальная единица, которая является субъектом социальных отношений. Актерами могут быть отдельные люди, но также и семьи, сообщества, организации или государства, если исследователя интересует не только внутренняя структура, но и отношения актеров с окружающим миром.

Сеть - это некоторый набор актеров и отношений между ними. Если говорят о социальных отношениях, то это называют социальной сетью. С помощью теории графов осуществляется анализ сетей, а используя свойства графов, можно узнать свойства социальных структур.

Выделяются различные категории сетевых организаций: внутренние сети, уменьшающие иерархию и делающие предприятие более открытым для рынка, вертикальные сети, повышающие производительность периодически зависимых звеньев путем создания различных форм сотрудничества независимых специализированных предприятий, межрыночные сети, укрепляющие горизонтальные связи между различными отраслями, потребительские сети, создаваемые на базе потребностей конечных потребителей. Большое внимание уделяется изучению межличностных отношений, которые складываются между сетями и внутри них. Для сетевых структур является важной способностью поддерживать и творчески использовать неформальные человеческие отношения. Если люди достаточно долго взаимодействуют между собой, например, в рамках одного предприятия, то образуются небольшие группы людей, например, совместно обедающих проводящих свободное время, помогающих друг другу воспитывать детей, которые имеют тенденцию становиться сетевой группой с преоб-

ладанием горизонтальных связей. Происходит неформальное структурирование коллектива на основе взаимной симпатии, интереса. Формируется как бы невидимая сеть внутри предприятия или организации.

Таким образом, под сетевой организацией понимается идеальный тип, характеризующийся структурой сети принципиально равноправных партнеров. Вследствие ограниченности собственных ресурсов для достижения общих целей члены сети прибегают к коллективному самоограничению [1].

Анализ социальных сетей в сети Интернет ведется не первое десятилетие и поскольку это некая интерпретация социума, есть смысл перенять инструменты изучения из социальных в виртуальные, но с учетом специфики последних.

Теория графов

Отслеживание связей — инструмент первичного установления наличия либо отсутствия отношений. Анализ сетей позволяет объяснить, каким образом отношения между людьми влияют на результаты деятельности, оценить потенциальную эффективность тех или иных реформ и нововведений, рассчитать оптимальный дизайн организации, определять, какие пользователи являются центральными для отношений, а какие — периферийными.

Теория графов является одним из разделов дискретной математики, который занимается изучением свойств графов.

Теория графов, как и большинство точных наук, возникла из решения и, вернее будет сказать, из формулировки известных математических проблем. В данном случае это задачи о Кенигсбергских мостах, разрезании пиццы и раскраски карты. Все вышеуказанные проблемы связаны, в первую очередь, с некими объектами, имеющими характеристики, представляемые графически, и поэтому естественно применить графы для исследования таких объектов как социальные сети [2].

В общем смысле графом называют совокупность двух конечных множеств, состоящих из множества точек, которые называют вершинами и множества пар вершин - рёбер. Он изображается точками на плоскости, прямоугольниками, овалами и другими фигурами, в которых раскрывается смысл вершины. При наличии между вершинами рёбер, соответствующие точки или фигуры соединяются отрезком или дугой. Если граф ориентированный, отрезки заменяют стрелками и явно указывают направление дуги.

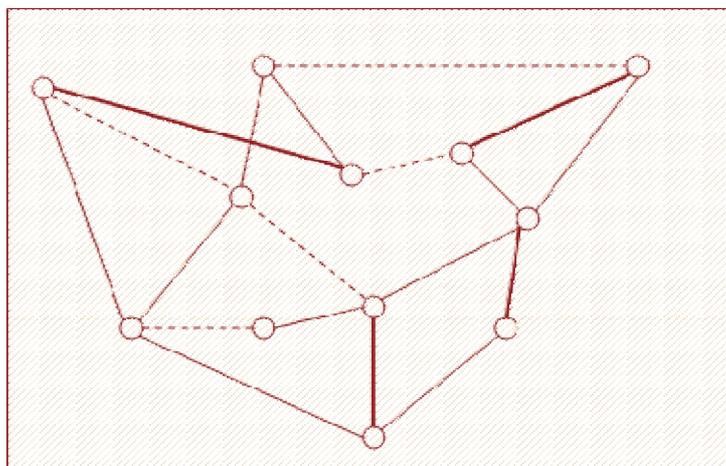


Рис. 1. Изображение графа

Поскольку одному графу могут быть сопоставлены различные графические представления, следует разделять изображение графа и собственно граф (абстрактную структуру). Изображение призвано лишь показать, какие вершины соединены рёбрами.

Социальным графом (social graph) называют граф, узлы которого представлены социальными объектами, например, пользовательские профили с их атрибутами, сообщества и различный медиа-контент, а рёбра - социальными связями между ними.

Неявным социальным графом (implicit social graph) является граф, который, при отсутствии явных социальных связей можно сформировать, исходя из взаимодействия пользователя с другими людьми и группами пользователей социальной сети.

Благодаря социальным графам решаются задачи идентификации пользователей, социального поиска, генерации рекомендации по выбору медиа-контента, выделяются "реальные" связи или идёт сбор открытой информации для моделирования графа. При поиске и обработке информации социальных графов приходится сталкиваться с некоторыми проблемами, например, с различием социальных сетей или закрытостью социальных данных.

Решаемые задачи

Рассмотрим подробнее задачи, решаемые с помощью социальных графов:

1. *Идентификация пользователей* заключается в том, чтобы обнаруживать профили, которые принадлежат одному и тому же человеку в разных социальных сетях. Решение этой задачи позволяет получить более совершенный социальный граф, что, в свою очередь, может быть полезно в решении других задач.

2. *Социальный поиск* или поиск социальных объектов (пользователей, их данных и записей и т.п.) основан на анализе некоторого количества связей, в которых находятся искомые объекты.

3. *Генерация рекомендаций* часто используется при создании графа интересов на основе социального графа. Она заключается в поиске наиболее точных алгоритмов генерации рекомендаций и предложений для пользователей. В генерации рекомендаций можно выделить:

1) Рекомендации друзей.

2) Рекомендации контента.

Существуют устоявшиеся подходы для рекомендательных систем:

1) Коллаборативная фильтрация - формируется список объектов, рекомендуемых исходя из мнений пользователей, которые ведут себя идентично.

2) Фильтрация содержимого - формируется на базе характеристик объекта и известной о нём информации.

3) Социальные подходы - основываются на социальных связях пользователей.

4. *Выявление "настоящих" связей* состоит в применении "разведки на основе открытых источников" для раскрытия реальных связей между пользователями.

Граф интересов

Граф интересов - это онлайн представление интересов определённого человека, которое было получено на основе его действий в социальной сети.

Вершинами графа могут быть интересы пользователя или его профиль в социальной сети, а рёбра показывают отношения между его вершинами. С помощью графа интересов можно доступно отобразить информацию о том, что конкретный пользователь хочет купить, что сделать или куда пойти.

Социальный граф и граф интересов имеют тесную связь, однако следует всё же различать эти понятия. Граф интересов проектируется для создания сети интересов человека, тогда как социальные сети построены на взаимодействии людей друг с другом и организованы на основе социального графа.

Таким образом, при анализе увлечений человека строится граф интересов, который позволяет адаптировать окружающую пользователя веб-среду, которая основана на пересе-

чении графа интересов с содержанием веб-пространства. Иногда граф интересов может происходить из социальной сети и может сохранять и обновлять данные на основе информации из этой социальной сети. Социальные сети нередко могут служить источником материала о явно выраженных интересах пользователя.

Граф интересов имеет несколько вариантов применения как для непосредственных пользователей, так и для людей, использующих его в бизнесе.

Сочетание графа интересов и социального графа могут быть использованы при установлении зависимостей между пользователями социальных сетей, предлагать схожие увлечения незнакомым друг другу людям. Также он применим в маркетинговых исследованиях, помогает проанализировать интересы аудитории для последующей реализации товаров или услуг, способствует более корректному изучению настроений текстов для контекстной рекламы, делает её настроенной на увлечения конкретного пользователя. Граф интересов в том числе помогает реализовывать продукт на основе пожеланий целевой аудитории производителя, с его помощью становится ясно, какие качества продукта стоит оставить, а какие улучшить в последующих версиях.

Например, создание графа можно предложить для поиска наиболее популярных авторов книг среди пользователей сети. В дальнейшем такую информацию можно использовать в рекламе книжного интернет-магазина на различных сайтах. Поиск будет осуществляться по ключевым словам, которыми будут фамилии писателей, упоминающихся пользователями социальной сети. Далее следует фрагмент кода на языке Python:

```
for site in set(sites): item = Author ()
item['Comm_text'] = unicode(" ".join(site.xpath('div/div/div').extract()))
if u'Джордж Оруэлл' or u'Оруэлл' or u'Orwell' in item['comm_text']:
item['Name'] = unicode('Orwell')
item['Label'] = unicode(" ".join(site.xpath('h3/a').extract())) #comment_title
item['Source'] = (" ".join(site.xpath('ul/li[2]/a/@href').extract())).split(" ", 1)[0]
item['Type'] = "Directed"
item['Weight'] = "1"
item['Comm_blog'] = " ".join(site.xpath('ul/li[3]/text()').extract())
item['Comm_path'] = " ".join(site.xpath('h3/a/@href').extract())
items.append(item)
```

Программа структурирует полученную информацию и сохраняет следующие поля:

- Заголовок комментария;
- Ссылка на комментарий;
- Текст комментария;
- Автор комментария;
- Блог, где размещен комментарий.

Для формирования графа в программном коде используются служебные поля:

- Label – подпись к узлу;
- Source – автор комментария или поста;
- Name – автор книги, которого упомянул в тексте автор комментария;
- Weight – вес ребра;
- Type – тип графа.

В результате анализа комментариев социальной сети можно увидеть, что популярность авторов книг в сети распределена следующим образом (рисунок 2).

Name	Weight
Orwell	7332
Bradbury	4978
Dostoevskiy	4911
Bulgakov	3756
Remarque	2658
Saint-Exupery	1658
Salinger	975

Рис. 2. Распределение количества упоминаний авторов

Таким образом, становится очевидно, каких авторов можно использовать для рекламы продукции книжного интернет-магазина в конкретной социальной сети.

Библиографический список

1. Социальные сети: теория и практика [Текст] : учебное пособие /Ю. Н. Фролов, Л. К. Габышева. — Тюмень : ТюмГНГУ, 2012. — 140 с.
2. Пастухова Г.В., Казаков В.В. АНАЛИЗ СОЦИАЛЬНЫХ СЕТЕЙ МЕТОДАМИ ТЕОРИИ ГРАФОВ С ПОМОЩЬЮ ПАКЕТА МАТНЕМАТИКА 10 // Материалы VII Международной студенческой научной конференции «Студенческий научный форум» URL: <https://scienceforum.ru/2015/article/2015012507>.

СЕКЦИЯ «ЭВМ И СИСТЕМЫ»

УДК 004.932; ГРНТИ 28.23.15

ИССЛЕДОВАНИЕ ГИСТОГРАММ РАСПРЕДЕЛЕНИЯ СПЕКТРАЛЬНЫХ СОСТАВЛЯЮЩИХ В СПЕКТРЕ ЛАНДШАФТНЫХ ИЗОБРАЖЕНИЙ

Б.В. Костров, Н.Н. Гринченко, А.А. Вьюгина, С.Н. Баранова

*Рязанский государственный радиотехнический университет имени В.Ф. Уткина,
Российская Федерация, Рязань, vyugina.an.a@gmail.com*

Аннотация. Работа с ландшафтными изображениями подразумевает их различную обработку, в том числе и анализ. Одной из наиболее интересных характеристик представляет частотный спектр изображения, полученный с помощью преобразования Уолша. Наглядно изобразив данный спектр в виде гистограммы, можно провести анализ распределения спектральных составляющих.

Ключевые слова: преобразование Уолша, ландшафтные изображения, гистограмма, частотный спектр.

INVESTIGATION OF HISTOGRAMS OF THE DISTRIBUTION OF SPECTRAL COMPONENTS IN THE SPECTRUM OF LANDSCAPE IMAGES

B.V. Kostrov, N.N. Grinchenko, A.A. Vyugina, S.N. Baranova

*Ryazan State Radio Engineering University named after V.F. Utkin,
Russia, Ryazan, vyugina.an.a@gmail.com*

The summary. Working with landscape images involves their various processing, including analysis. One of the most interesting characteristics is the frequency spectrum of the image obtained using the Walsh transform. By visualizing this spectrum in the form of a histogram, you can analyze the distribution of spectral components.

Keywords: Walsh transform, landscape images, histogram, frequency spectrum.

Некоторые сферы деятельности работают с большими участками земель, кроме того, огромные просторы занимают водные поверхности: реки, озера, моря и океаны. В связи с этим может возникнуть необходимость провести различного рода анализ данных территорий на предмет половодья весной, успешности посевов осенью и других аналитических работ. Для исследования таких территорий приходится использовать аппаратуру, позволяющую снимать ландшафтные изображения с высоты. Такие снимки могут иметь разное качество и информативность. Для определения этих характеристик возможно провести исследование спектральных составляющих, отображенных в виде гистограммы.

Первым шагом для анализа ландшафтного изображения является его перевод к частотному спектру. Для этого исходное изображение подвергается быстрому преобразованию Уолша-Адамара, упорядоченное по Уолшу. В нашей работе будет использовано одномерное преобразование по столбцам, что позволит исследовать исходное изображение на предмет изменения частот в линейном пространстве.

С математической точки зрения функции Уолша – это семейство функций, которые составляют ортогональную систему и могут принимать из всех значений только 1 и -1 на всей области определения. При дискретном спектральном анализе сигналов с N отсчетами число спектральных составляющих должно быть равно N , поэтому число функций, включаемых в систему, равно числу отсчетов каждой функции [1].

В системе функций Уолша-Адамара элементарная матрица Адамара при $N = 1$ имеет вид $H_1 = [1]$. При $N=2$ матрица примет вид:

$$H_2 = \begin{bmatrix} +1 & +1 \\ +1 & -1 \end{bmatrix}. \quad (1)$$

Для $N=4$:

$$\mathbf{H}_4 = \begin{bmatrix} \mathbf{H}_2 & \mathbf{H}_2 \\ \mathbf{H}_2 & -\mathbf{H}_2 \end{bmatrix}. \quad (2)$$

Тогда для $N = 2^n$ матрица имеет вид:

$$\mathbf{H}_n = \begin{bmatrix} \mathbf{H}_{2^{n-1}} & \mathbf{H}_{2^{n-1}} \\ \mathbf{H}_{2^{n-1}} & -\mathbf{H}_{2^{n-1}} \end{bmatrix}. \quad (3)$$

В системе Уолша [2] $\{wal(i,j)\}$ функции расположены в порядке увеличения их числа знакоперемен на интервале определения. Эта система:

$$\mathbf{H}_{w(3)} = \begin{bmatrix} +1 & +1 & +1 & +1 & +1 & +1 & +1 & +1 \\ +1 & +1 & +1 & +1 & -1 & -1 & -1 & -1 \\ +1 & +1 & -1 & -1 & -1 & -1 & +1 & +1 \\ +1 & +1 & -1 & -1 & +1 & +1 & -1 & -1 \\ +1 & -1 & -1 & +1 & +1 & -1 & -1 & +1 \\ +1 & -1 & -1 & +1 & -1 & +1 & +1 & -1 \\ +1 & -1 & +1 & -1 & -1 & +1 & -1 & +1 \\ +1 & -1 & +1 & -1 & +1 & -1 & +1 & -1 \end{bmatrix}. \quad (4)$$

Множество функций Уолша можно обозначить по формуле:

$$S_w = \{wal_w(i, j), i = 0, 1, \dots, N-1\}, \quad (5)$$

где $N = 2^n, n = 1, 2, 3, \dots; w$ – индекс, обозначающий упорядочение по Уолшу.

Преобразование \mathbf{H}_w для некоторого блока $(n \times n)$, где $\mathbf{X}_{(n)} = \{X(0), X(1), \dots, X(N-1)\}$ определяется матричным уравнением вида:

$$\mathbf{X}_{(n)} = \frac{1}{N} \mathbf{H}_{w(n)} \cdot \mathbf{V}_{x(n)}, \quad (6)$$

где $\mathbf{X}_{(n)}$ – матрица исходного изображения, а $\mathbf{V}_{x(n)}$ – матрица коэффициентов прямого преобразования Уолша.

Исходя из свойств преобразования Уолша [3]: $\mathbf{H}_{w(n)}$ – ортогональная и симметрическая, тогда обратное преобразование записывается в следующем виде:

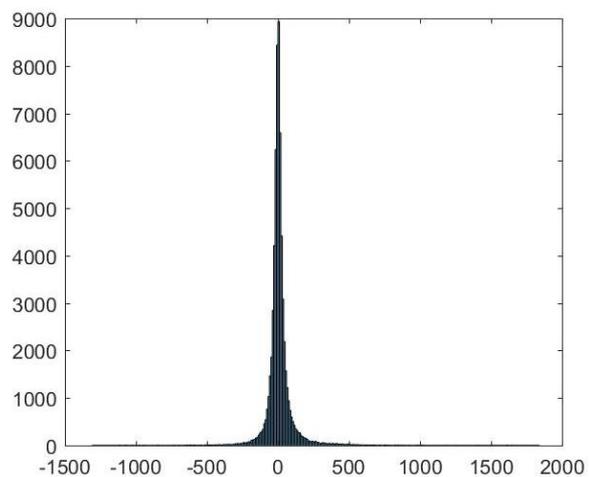
$$\mathbf{V}_{x(n)} = \mathbf{X}_{(n)} \cdot \mathbf{H}_{w(n)}. \quad (7)$$

На рисунках 1-3 представлены исходные данные и построенные гистограммы по частотному спектру на трех разных типах ландшафтных изображений: земной и водной глади, лесных массивов, горных пустынных просторов. Все изображения были взяты размером

256x256. Гистограммы построены средствами Matlab [4], которые автоматически определили, диапазоны, равные 30.

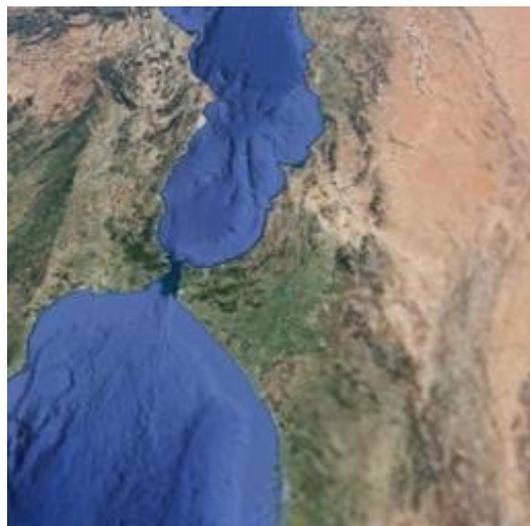


а

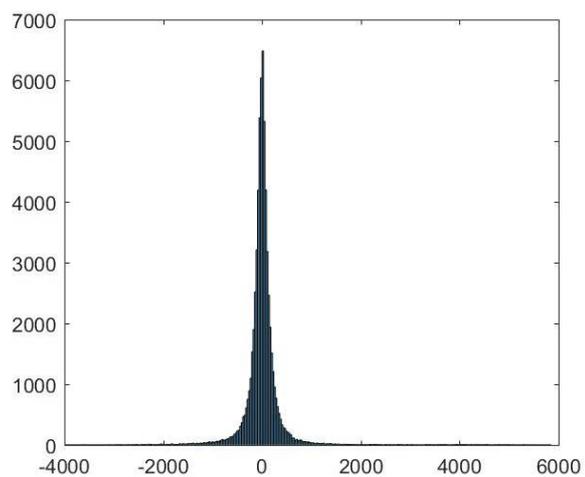


б

Рис. 1. Пример для изображения «Лесной массив»: а – исходное изображение, б – гистограмма



а



б

Рис. 2. Пример для изображения «Водоем»: а – исходное изображение, б – гистограмма

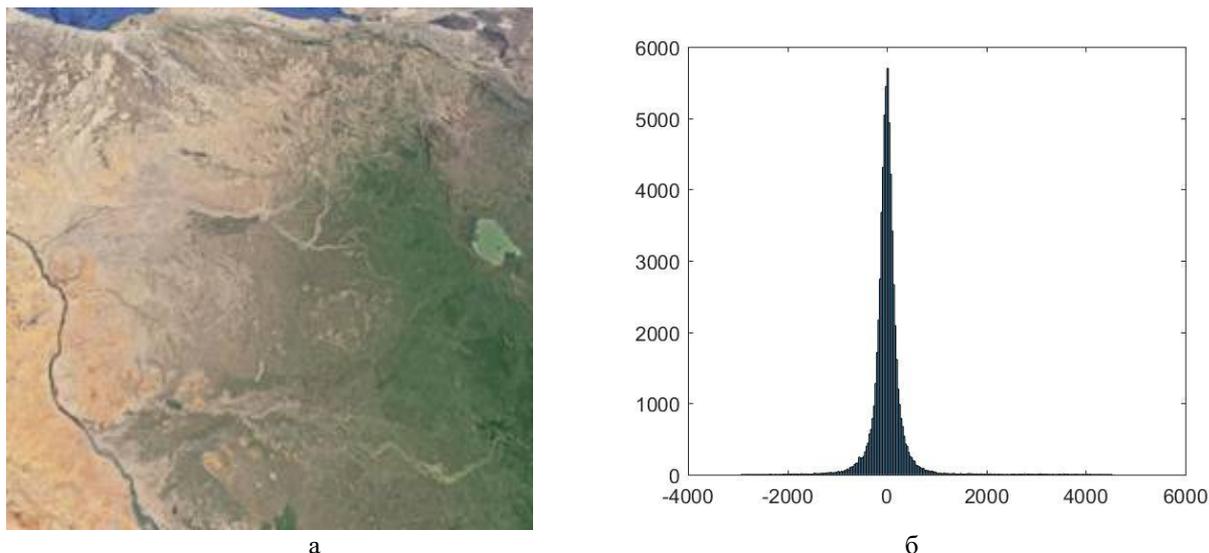


Рис. 3. Пример для изображения «Пустыня»: а – исходное изображение, б – гистограмма

При анализе гистограммы одним из наиболее интересующих параметров является размах, который вычисляется как разница между максимальным и минимальным значением.

В качестве примеров были взяты изображения разных ландшафтов, которые также имеют неоднородную структуру. Так, лесной массив наиболее гладкий и не имеет резких перепадов яркости, что дает значения спектра, наиболее приближенные к нулевым значениям. Размах для данного набора спектра $1840 - (-1310) = 3150$.

Наиболее интересными для исследования представляется пара изображений, представленных на рисунках 2 и 3. Здесь имеются различные значения яркости исходных пикселей, что дает больший разброс полученных после преобразования значений спектра. В обоих случаях размах гистограммы достигает:

$5850 - (-3990) = 9840$ для изображения «Водоем», $4530 - (-2940) = 7470$ – «Пустыня».

И при визуальном сравнении с ландшафтом лесных массивов можно выделить, что у изображения «Водоем» имеются резкие границы между сушей и водой, которые соответствуют высокочастотной составляющей спектра. Однако за счет сглаженности самого водоема и прилегающей земли сохраняется большое количество значений спектра, приближенного к нулю.

Важной информацией, которую можно извлечь из гистограммы изображений, является количество диапазонов в разрезе содержащихся в них элементов. Так в таблице 1 представлена числовая статистика в зависимости от различного числа элементов, а также общее количество полученных диапазонов.

Таблица 1. Статистика количества диапазонов частот для изображений

Изображение	Количество диапазонов	< 6000 и >= 4000	< 4000 и >= 2000	< 2000 и >= 500	< 500 и >= 0
Лесной массив	315	2	3	11	295
Водоем	364	4	4	13	305
Пустыня	249	6	6	12	225

Из анализа таблицы можно выявить зависимость от количества различного количества яркостей на изображении с большим значением общего значения диапазонов, что особенно заметно в сравнении «Водоема», где количество диапазонов больше по сравнению с другими изображениями.

Стоит обратить внимание на соотношение общего числа высокочастотных диапазонов, которые имеют небольшое количество элементов спектра (менее 500). Из всего диапазона 93,7%, 83,8%, 90,3% занимают диапазоны, стремящиеся к граничным значениям. В рассматриваемые промежутки не вошли некоторые частоты, количество элементов в которых превышает 6000. Так, для изображения «Водоем» таких диапазонов составляет 38, что составляет 10,4%.

Вид гистограммы стремится к нормальному распределению, поэтому еще одной исследуемой характеристикой является количество элементов в спектре разных знаков. Данные параметры представлены в таблице 2 и отражают, что количество положительных и отрицательных точек спектра примерно равно.

Таблица 2. Статистика положительных и отрицательных элементов в спектре

Изображение	+ элементов в спектре	-элементов в спектре
Лесной массив	32412	33124
Водоем	34936	30600
Пустыня	32941	32595

На основе различных частотных диапазонов можно построить изображения, отображающие структуру спектральных составляющих (рисунки 4 и 5).

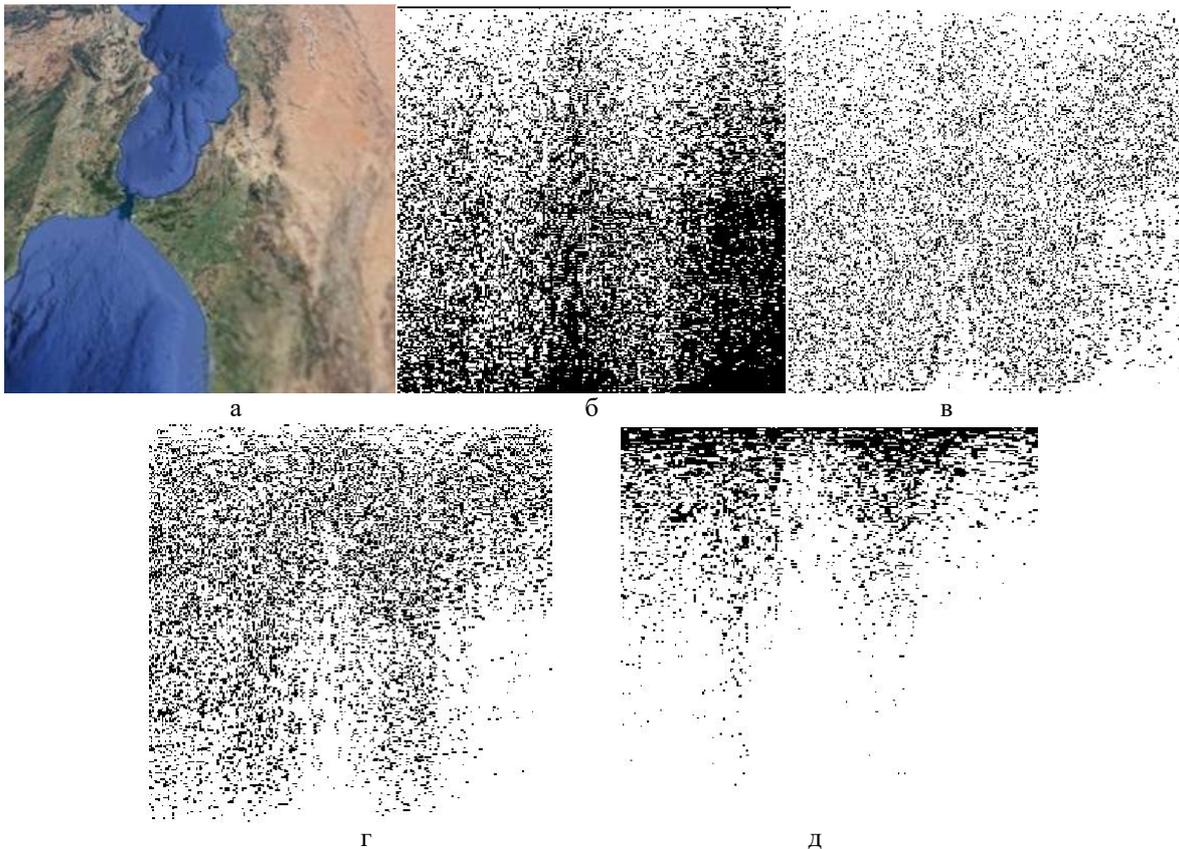


Рис. 4. Структура спектральных составляющих изображения «Водоем»: а – исходное изображение, б – 0-90, в – 90-150, г – 150-360, д – 360 и более

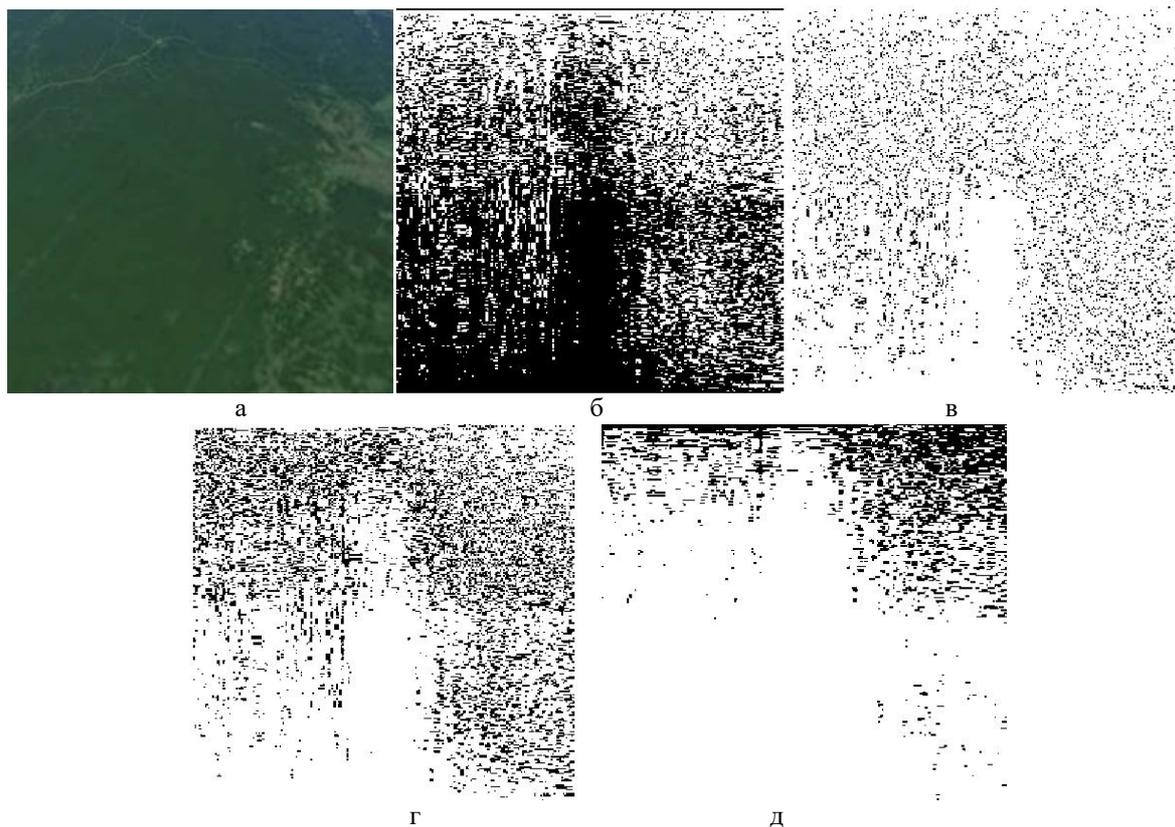


Рис. 5. Структура спектральных составляющих изображения «Лесной массив»: а – исходное изображение, б – 0-30, в – 30-40, г – 40-100, д – 100 и более

Исходя из проделанного анализа, можно сделать вывод, что гистограмма частотного спектра позволяет сказать о сглаженности изображения, так как все значения спектра сгруппированы и отображены в соответствии с их категорией. Благодаря значению, полученным при анализе гистограммы, можно увидеть, что различные ландшафтные элементы в частотном представлении отличаются большим размахом в диапазоне и также имеют заметное смещение значений. Частотный спектр позволяет выделить места перепада яркости, которые совпадают с высокими значениями, которые хорошо выделяются на построенной гистограмме.

Изображение структуры спектра позволяет выявить энергетически малозначимые элементы на исходном изображении, что является предпосылкой для эффективного сжатия. Также полученная информация из данного анализа может найти практическое применение для дешифрирования ландшафтных изображений или каталогизации снимков.

Библиографический список

1. Злобин В.К., Костров Б.В., Свирина А.Г. Спектральный анализ изображений в конечных базисах. М.: КУРС: ИНФРА 2016. 172 с.
2. Моделирование канала передачи аэрокосмических изображений с использованием каскадного кода / Б.В. Костров, Н.Н. Гринченко, Е.С. Герашенко, В.Ю. Потапова, А.С. Тарасов, А.В. Токарев // Известия Тульского государственного университета. Технические науки. 2016. Вып. 2. С.124 – 129.
3. Ахмед Н., Рао К. Р. Ортогональные преобразования при обработке цифровых сигналов: Пер. с англ./Под ред. И. Б. Фоменко. М.: Связь, 1980. 248 с.
4. Гонсалес Р., Вудс Р., Эддинс С. Цифровая обработка изображений в среде Matlab. М.: Техносфера, 2006. 616 с.

УДК 621.324; ГРНТИ 50.09.47

МОДЕЛЬ ПЛАНИРОВЩИКА ЗАДАЧ В ВИДЕ ФУНКЦИОНАЛЬНЫХ БЛОКОВ IEC 61499

Р.А. Бикташев, Д.М. Жданов, Д.С. Журкин

Пензенский государственный университет,

Российская Федерация, Пенза, dmitry.zhdanov01@yandex.ru

Аннотация. Предложена модель планировщика задач с глобальной очередью для управляющих многопроцессорных систем реального времени в виде функциональных блоков с конкретным примером реализации.

Ключевые слова: функциональный блок, IEC 61499, многопроцессорные системы, nxtStudio.

THE MODEL OF TASK SCHEDULER AS FUNCTION BLOCKS IEC 61499

R.A. Biktashev, D.M. Zhdanov, D.S. Zhurkin

Penza State University,

Russia, Penza, dmitry.zhdanov01@yandex.ru

The summary. A model of a task scheduler with a global queue for real-time control multiprocessor systems is proposed in the form of function blocks with a specific implementation example.

Keywords: function block, IEC 61499, multiprocessor systems, nxtStudio.

Необходимость повышения производительности и надежности управляющих систем промышленной автоматики, работающих в реальном времени, всегда было актуальной задачей. Известно, что решение этой проблемы лежит на пути создания распределенных и многопроцессорных систем управления. Однако их производительность и надежность в большой степени зависит от характеристик ядра распределенной операционной системы. В работе [1] предлагается планировщик задач с аппаратной поддержкой функции диспетчеризации многопроцессорных систем с глобальной очередью задач, который дает возможность реализации ядра операционных систем с высокой реактивностью и надежностью. Одной из проблем, которые встают при разработке подобных устройств, в том числе и планировщиков задач для многопроцессорных систем, является проверка работоспособности на стадии эскизного проектирования.

Алгоритм планирования с разделением времени и глобальной очередью задач представляет наиболее простым и одновременно эффективным способом реализации планировщика задач. Он обеспечивает равномерную загрузку процессоров управляющей системы, что способствует обеспечению высокой производительности многопроцессорной системы.

Алгоритм работы устройства планировщика основан на использовании классической задачи «работа парикмахерской», содержащий клиентскую и серверную части. Клиентская часть осуществляет постановку задач в очередь, серверная - выбирает задачу из очереди по правилам FIFO, LIFO или приоритетному, и запускает её на обслуживание в выбранный процессор. Действия клиента и сервера синхронизируются по схемам «писатели-читатели» и «рандеву», причем последняя предполагает наличие выбранной задачи и готовность процессора к обслуживанию этой задачи. В качестве клиента выступает планировщик задач, в качестве сервера – диспетчер задач совместно с процессорный блоком.

В самом общем виде функционирование планировщика/диспетчера задач выглядит следующим образом. В начальном состоянии каждый включающийся в работу процессор выполняет операцию «готов» и переходит в спящий режим, в котором находится до тех пор, пока на обслуживание не поступит задача из очереди. Если все процессоры заняты обслуживанием, то вновь поступившая задача помещается в конец очереди, число мест в которой ограничено. Если вновь поступившая задача обнаружит, что очередь заполнена, она на обслуживание не принимается и покидает систему. Принятая на обслуживание задача находится в очереди до тех пор, пока не поступит на выполнение в процессор, при этом в очереди осво-

бождается одно место. После выполнения очередной задачи просматривается очередь, и если в ней имеются хотя бы одна ожидающая задача, то диспетчер по описанному выше правилу назначает её на выполнение в один из свободных процессоров. Если очередь пуста, то свободные процессоры переходят в режим ожидания. Если в процессорном пуле имеется несколько свободных процессоров, то производится приоритетный или иной способ выбора одного из них для обслуживания очередной задачи.

Одним из основных трендов в разработке распределенных систем промышленной автоматизации является использование международного стандарта IEC 61499, основанного на функционально-блочной модели [2]. Цель данной работы – показать применимость и полезность использования представления аппаратных моделей в видефункциональных блоков (ФБ) IEC 61499, которые призваны инкапсулировать в себе реализацию тех или иных функций и предоставлять интерфейсы для взаимодействия с другими блоками. Таким образом, происходит переход на более высокий уровень абстракции – интеллектуальные устройства будут обеспечивать часть функциональности, из которой будет создаваться общая схема взаимодействия.

Для реализации планировщика задач на функциональных блоках была выбрана демонстрационная версия среды разработки nxtSTUDIO 3.0 SP1, разработанная компанией nxtControl. Она легко интегрирует все задачи автоматизации в одном инструменте, предлагает готовые программные библиотеки, обеспечивает аппаратно-независимую разработку и использует IEC 61499, а также IEC 61131 в качестве парадигмы управления.

Были реализованы следующие ФБ, изображённые в схеме соединения на рисунке 1:

E_control_queue – блок управления очередью задач предназначен для приема идентификаторов задач в систему [1]. Он производит анализ на наличие свободного места в очереди задач. Если оно есть, помещает идентификатор новой задачи в блок очереди. Также осуществляет выборку идентификатора задачи из очереди и передаёт его в E_assign_task.

E_queue_task – блок очереди задач хранит поступающие идентификаторы задач [1]. По запросу от блока управления очередью задач, заносит идентификатор новой задачи в список или извлекает идентификатор запускаемой задачи из списка для передачи его в процессор. Внутренней реализацией является очередь LIFO на 10 элементов.

E_choosing_proc – блок выбора процессора принимает все сигналы «Свободен» на событийные входы (PF1, PF2, и PF3) от всех процессоров в системе (3 процессора) [1]. Если в наличии имеется хотя бы один процессор, не занятый обслуживанием текущей задачи, то этот блок устанавливает активный уровень сигнала «Есть свободные процессоры» на событийном выходе (PFO) и таким образом извещает блок управления назначением задач о том, что имеются «спящие» процессоры. Выбор процессоров осуществляется схемой взаимоисключения и схемой приоритетов. Для сообщения блоку диспетчеризации о том, какой из процессоров выбран для обслуживания, служат информационные выходы (PS1, PS2, PS3).

E_assign_task – блок синхронизации и назначения задач анализирует информацию о наличии ожидающих задач в очереди и свободных процессоров, назначаемых для обработки этих задач [1]. Данный блок осуществляет взаимодействие очереди задач с процессорами. Блок выбора процессора формирует информацию о том, какой из свободных процессоров выбран для обслуживания текущей задачи. Идентификатор принятой задачи передается блоком очереди задач по шине V_task_id.

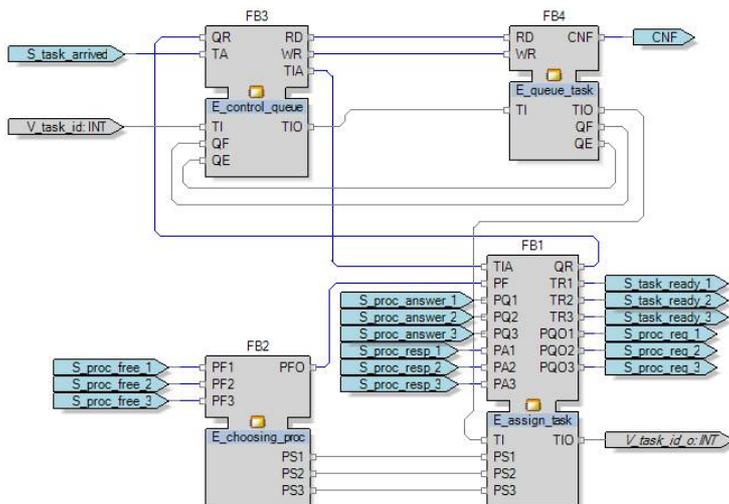


Рис. 1. Планировщик задач в виде ФБ

На рисунке 2 изображены ЕСС диаграмм созданных ФБ, которые работают по поступающим событиям. При инициализации ФБ, его состоянием является **Start** (двойной прямоугольник). Обычный прямоугольник – это состояния ФБ, которые соединены переходами (стрелочки, имеющие условие для осуществления перехода), а также за состоянием может быть закреплен блок действий (в левой части записан алгоритм или ссылка на него; в правой части какие выходные события активируются в данном состоянии).

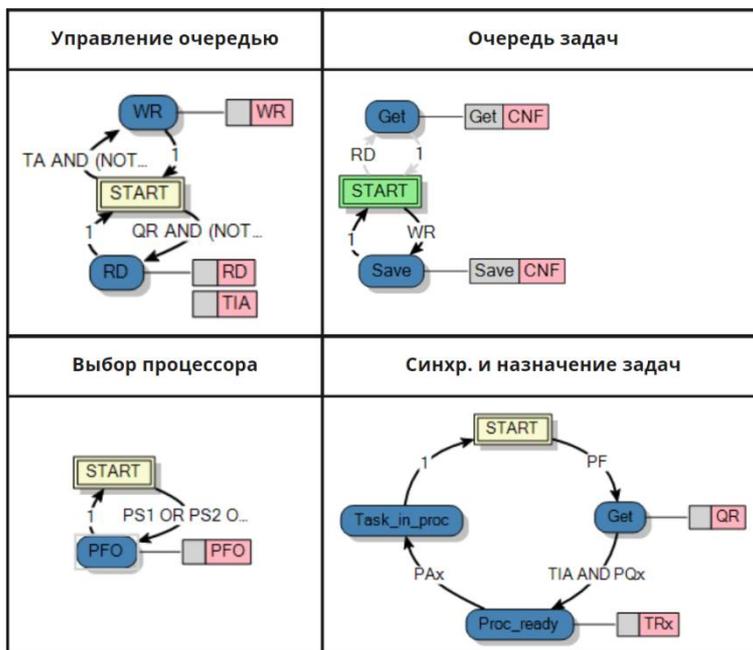


Рис. 2. ЕСС диаграммы базовых ФБ планировщика задач

В блоке управления очередью задач видно, что при поступлении события **TA** (TaskArrived) и на информационном входе **QF** (Queuefull) сигнал False, выполняется переход в состояние **WR** (write), и связанные с ним действия подают сигнал на событийный выход **WR**, после чего возвращается в стартовое состояние. Событийный выход **WR** связан WITH-связью с информационным выходом **TIO** (taskidoutput), что определяет интерфейс подключения к следующему за ним ФБ. При выполнении перехода **QR & (NOTQE)** состояние ФБ

меняется на **RD** (read), и связанные с ним действия подают сигнал на событийные выходы **RD** и **TIA**.

Интерфейсы разных ФБ выполняются корректно. Например, ФБ управления очередью задач имеет событийный выход **WR** с WITH-связью с информационным выходом **TIO**, а ФБ очереди задач имеет событийный вход **WR** с WITH-связью с информационным входом **TI**. Таким же образом, ФБ выбора процессора имеет выход **PFO** с WITH-связью с **PS1**, **PS2** и **PS3**, а ФБ синхронизации и назначения задач имеет вход **PF** с WITH-связью с **PS1**, **PS2** и **PS3**.

ФБ очереди задач имеет некоторые алгоритмы, что должны выполняться в определённых состояниях. Ниже приведены листинги алгоритмов:

Get:

```
ALGORITHM Get IN ST:
  IF (Cursor > 0) THEN
    Cursor:=Cursor-1;
    TIO := Queue[Cursor];
    QF:=FALSE;
  END_IF;
  IF (Cursor = 0) THEN
    QE := TRUE;
  ELSE
    QE := FALSE;
  END_IF;
END_ALGORITHM
```

Save:

```
ALGORITHM Save IN ST:
  IF (Cursor < 10) THEN
    Queue[Cursor] := TI;
    Cursor:=Cursor+1;
    QE:=FALSE;
  END_IF;
  IF (Cursor = 10) THEN
    QF := TRUE;
  ELSE
    QF := FALSE;
  END_IF;
END_ALGORITHM
```

Среда разработки nxtSTUDIO позволяет проводить отладку отдельно взятых базовых ФБ. На рисунке 3 представлен процесс отладки базового ФБ очереди задач. В разделе интерфейса ФБ можно подавать данные на информационный вход **TI** (TaskID), на событийные входы **RD** (Read) и **WR** (Write). В нижнем разделе отслеживаемых переменных представлено состояние внутреннего массива **Queue** на 10 элементов и значение **Cursor** (курсор указывает на пустой элемент, доступный для записи).

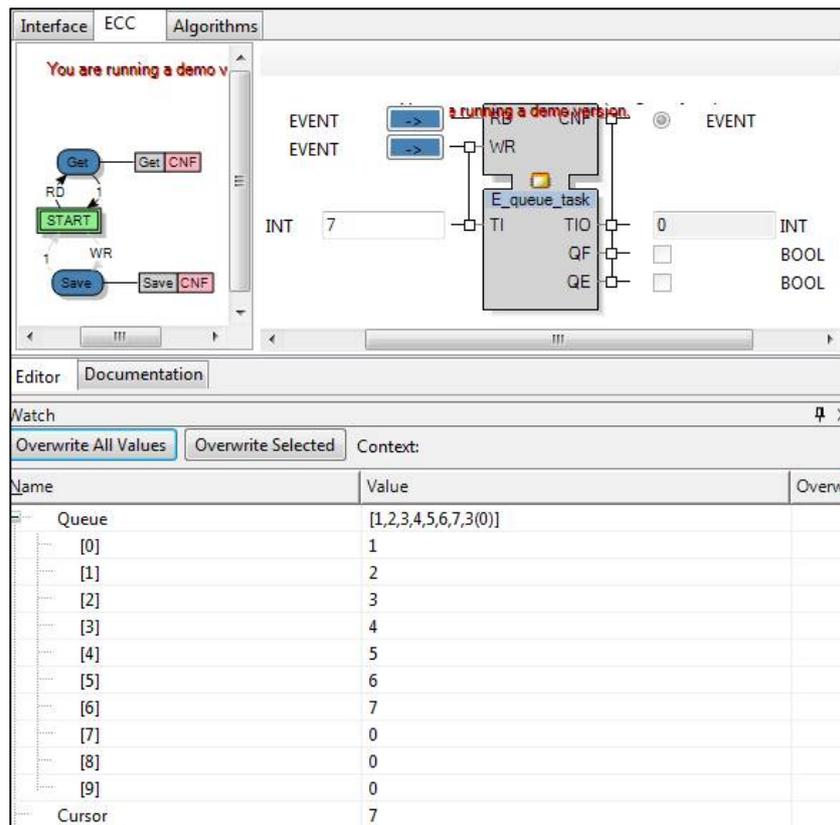


Рис. 3. Процесс отладки базового ФБ очереди задач

На ФБ наглядно видно интерфейс, от каких сигналов зависит переход состояния, от каких сигналов зависят те или иные данные (WITH-связь). По ECC диаграмме можно посмотреть поведение блока и как на него влияют те или сигналы, проследить путь алгоритма поведения ФБ. Инкапсулируя интерфейс и поведение, это становится больше похоже на объектно-ориентированную модель проектирования, которая получила широкое признание в области программирования. Структурируя таким образом интерфейсы, можно добиться большей совместимости различных устройств от различных поставщиков.

В ходе выполнения данной работы была разработана модель планировщика задач в виде ФБ IEC 61499 в среде разработки nxtSTUDIO, а также протестирована работа его очереди задач. В дальнейшем необходимо написать клиентское приложение, позволяющее запустить общую систему.

Библиографический список

1. Вашкевич, Н. П. Аппаратная поддержка диспетчера задач с глобальной очередью в многопроцессорных системах / Н. П. Вашкевич, Р. А. Бикташев, А. И. Меркурьев // Известия высших учебных заведений. Поволжский регион. Технические науки. – 2011. – № 3. – С. 3–14.
2. Дубинин В.Н., Вяткин В.В. Модели функциональных блоков IEC 61499, их проверка и трансформации в проектировании распределенных систем управления: монография. – Пенза: Изд-во ПГУ, 2012. – 348 с.

УДК 004.627; ГРНТИ 28.23.15

ВЫБОР РАЗМЕРНОСТИ ОРТОГОНАЛЬНОГО БАЗИСА ДЛЯ ПЕРЕДАЧИ ИЗОБРАЖЕНИЙ

А.С. Бастрычкин

*Рязанский государственный радиотехнический университет имени В.Ф. Уткина,
Российская Федерация, Рязань, a_s_bas@mail.ru*

Аннотация. В работе рассматриваются различные ортогональные преобразования на предмет выбора оптимальной размерности базисных функций для передачи изображений. Приводятся основные особенности преобразований, достоинства и недостатки, а также результаты замера времени выполнения преобразований в зависимости от размерности базиса.

Ключевые слова: ортогональные преобразования, передача изображений, сжатие изображений, преобразование Уолша, квазидвумерный спектр.

SELECTION OF THE ORTHOGONAL BASIS DIMENSION FOR IMAGE TRANSMISSION

A.S. Bastrychkin

*Ryazan State Radio Engineering University named after V.F. Utkin,
Russia, Ryazan, a_s_bas@mail.ru*

The summary. The paper describes different orthogonal transformations to select optimal basis functions dimension for image transmission. Given their main transformations features, advantages and disadvantages, and results of transformation time measurement depending on basis dimension.

Keywords: orthogonal transformations, image transmission, image compression, Walsh transform, quasi-two-dimensional spectrum.

В настоящее время существует множество методов и алгоритмов устранения информационной избыточности. Однако на практике и в литературе недостаточное влияние уделяется методам, учитывающим информационные ограничения канального оборудования и назначение систем. Поиск подобных методов позволил бы существенно сократить расходы на разработку и реализацию систем передачи информации.

Отдельное место занимает задача поиска оптимального преобразования для перехода в спектральное пространство, так как известно, что подобные операции могут занимать существенное время и тратить много ресурсов [1]. Также следует учитывать не только вид этого преобразования, но и размерность базиса, которая может значительно повлиять на время перехода от исходного сигнала к его частотному представлению.

В данной работе рассмотрено несколько преобразований, позволяющих перейти в спектральные пространства.

Дискретное преобразование Фурье

Дискретное преобразование Фурье (ДПФ) двумерного изображения может быть выражено формулой (1):

$$F(u, v) = \frac{1}{MN} \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} f(x, y) e^{-i2\pi\left(\frac{ux}{M} + \frac{vy}{N}\right)}, \quad (1)$$

где u, v – спектральные координаты;

x, y – координаты изображения;

M, N – размер изображения f .

Данное преобразование переводит изображение в частотную область, но элементы спектра являются комплексными числами, и из-за этого они мало подходят для применения в методах устранения информационной избыточности.

Дискретное косинусное преобразование

Дискретное косинусное преобразование (ДКП) является преобразованием, которое переводит изображение в пространство косинусных компонент, при этом все операции происходят с действительными числами (2):

$$F(u, v) = \frac{2}{N} \alpha(u)\alpha(v) \sum_{x=0}^M \sum_{y=0}^N f(x, y) \cos \left[\frac{(2x+1)u\pi}{2M} \right] \cos \left[\frac{(2y+1)v\pi}{2N} \right], \quad (2)$$

где $\alpha(u), \alpha(v) = \frac{1}{\sqrt{2}}$ при $u, v = 0$ и 1 в остальных случаях.

Данное преобразование нашло широкое применение в алгоритмах сжатия сигналов, таких как JPEG, MPEG, MP3. Причиной распространённости данного преобразования является высокая плотность упаковки информации в малое число коэффициентов.

Преобразование Хаара

Простейшее вейвлет преобразование (преобразование Хаара) заключается в разложении исходного сигнала по функциям, изображённым на рисунке 1.

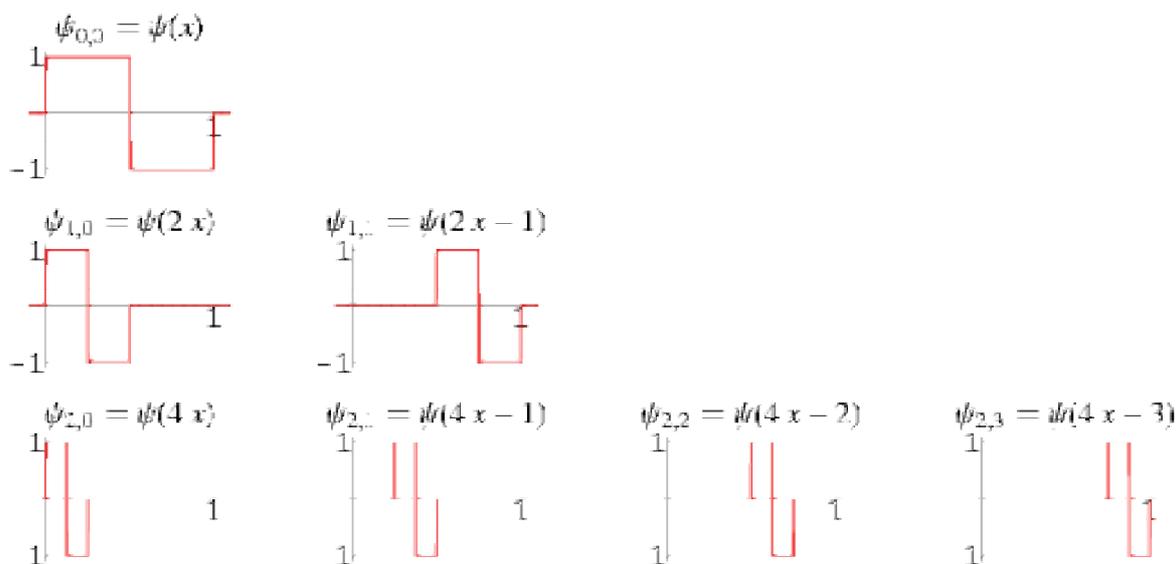


Рис. 1. Функции системы Хаара

Для двумерного цифрового сигнала преобразование Хаара будет заключаться в нахождении полусумм (аппроксимирующих компонент) и полуразностей (детализирующих компонент) соседних элементов исходного сигнала [2].

Вейвлет функции, как и ДКП, способны упаковывать большое количество информации в малое число коэффициентов разложения. Благодаря этим свойствам вейвлеты также нашли применение в алгоритмах сжатия, например, в JPEG2000.

Преобразование Уолша

Преобразование Уолша можно представить в следующем виде(3):

$$w(u) = \frac{1}{N} \sum_{x=0}^{N-1} f(x) (-1)^{\langle x, u \rangle}, \quad (3)$$

где $\langle x, u \rangle = \sum_{s=0}^{n-1} x_s u_s$;

$n = \log_2 N, x_s$ и u_s - коэффициенты двоичного представления чисел x и u .

Преобразование Уолша имеет также быстрый алгоритм, реализация которого схожа с реализацией быстрого преобразования Фурье методом Кули-Тьюки [3]. Также если производить преобразование Уолша только по строкам, можно получить квазидвумерный спектр изображения [1].

Деление изображений на блоки

Деление изображений на блоки является неотъемлемой частью многих алгоритмов сжатия изображений, использующих переход к спектральным пространствам, таких, как JPEG, JPEG2000, WebP.

Преимущества данной операции состоит в том, что алгоритм становится более универсальным, то есть подходит для большого числа форматов изображений. Также существенным преимуществом является то, что деление на блоки даёт гораздо меньшие спектральные коэффициенты, чем преобразование всего изображения.

Среди недостатков данной операции можно отметить существенное увеличение времени перехода к спектральному представлению.

Размер блока изображения напрямую определяет размерность базиса ортогональных функций, по которому будет происходить разложение сигнала и переход к спектрам.

В данной работе исследовано влияние размера блока на время получения спектрального представления для перечисленных выше преобразований.

Описание эксперимента

Для каждого из преобразований были выполнены следующие действия:

1. Сгенерировано 500 матриц размером 512x512, заполненных значениями от 0 до 255 с равномерным распределением;
2. Каждая матрица разбивалась на блоки размерами от 2x2 до 256x256;
3. Над каждым блоком выполнялось исследуемое преобразование;
4. Производился замер среднего времени перехода от изображения к спектрам в блоках в зависимости от размера блока.

Данный эксперимент позволит выявить зависимость времени преобразования от его вида и от размера блока.

Эксперимент был реализован в системе Matlab, в качестве исследуемых были выбраны стандартные реализации перечисленных выше преобразований.

Результаты и выводы

Результат представлен в виде графика на рисунке 2. На данном графике FFT2—это быстрое преобразование Фурье, DST2 —дискретное косинусное преобразование, HAAR2 – преобразование Хаара, FWHT—преобразование Уолша-Адамара, квазидвумерный спектр.

Рисунок 2 позволяет сделать представленные далее выводы.

Сокращение размерности базиса (уменьшение размера блока) может приводить к существенному увеличению времени перехода к спектральному пространству. Соответственно, для сокращения времени перехода целесообразно выбирать наибольший возможный размер блока.

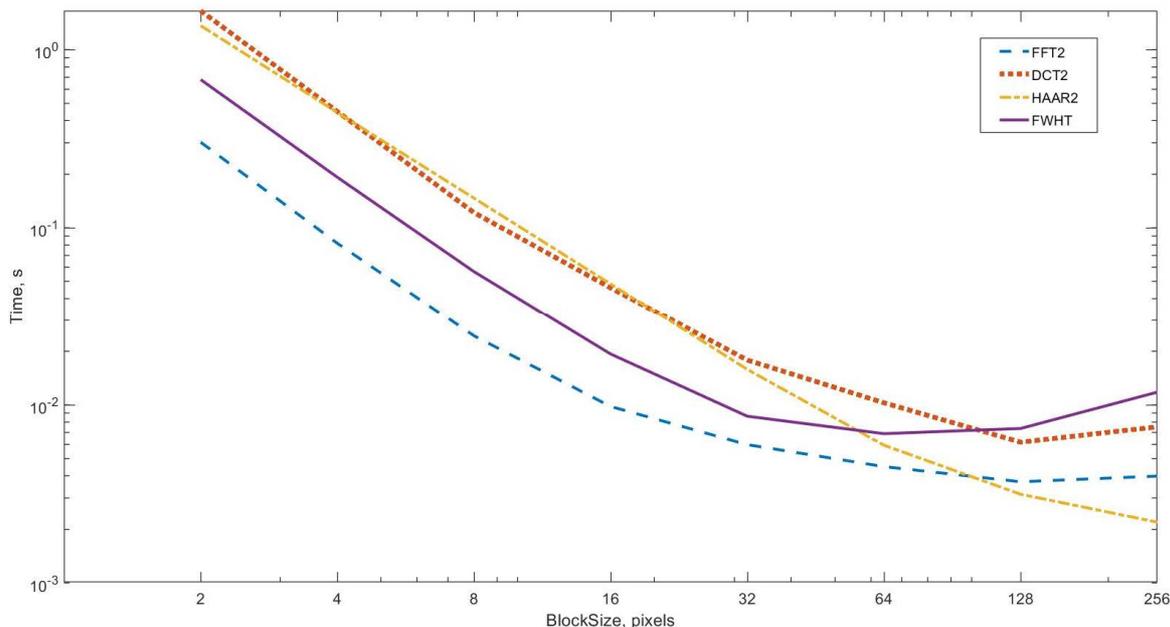


Рис. 2. Зависимость времени перехода к спектрам от размера блока для различных преобразований

Преобразование Фурье имеет наименьшее время в большинстве случаев из-за высокой оптимизации стандартной реализации. Функция преобразования Фурье реализована с помощью сторонней библиотеки FFTW, написанной на языке C, тогда как остальные функции реализованы без использования сторонних библиотек. К тому же дополнительным неудобством представляется работа с комплексными числами.

При использовании квазидвумерного преобразования Уолша и размера блока от 8x8 до 64x64 можно получить выигрыш во времени в 1.5 – 2 раза по сравнению с ДКП и преобразованием Хаара.

Таким образом, для разработки алгоритмов сжатия двумерных сигналов, ориентированных на существенное упрощение и сокращение времени работы целесообразно использовать квазидвумерное представление сигналов.

Библиографический список

1. Злобин В. К., Костров Б. В., Свирина А. Г. Спектральный анализ изображений в конечных базисах. М.: КУРС; ИНФРА-М, 2016. 172 с.
2. Гонзалес Р., Вудс Р., Эддинс С. Цифровая обработка изображений в среде MATLAB. М.: «Техносфера», 2006. 616 с.
3. Beauchamp, Kenneth G. Applications of Walsh and Related Functions: With an Introduction to Sequency Theory. London: Academic Press, 1984.

УДК 004.932.2, 159.942.33; ГРНТИ 28.23.15, 15.01.77

ПОИСК АНТРОПОМЕТРИЧЕСКИХ ТОЧЕК ЛИЦА С ПОМОЩЬЮ OPENFACE И MEDIAPIPE

В.А. Саблина, А.В. Савин

*Рязанский государственный радиотехнический университет имени В.Ф. Уткина,
Россия, Рязань, sablina.v.a@evm.rsreu.ru*

Аннотация. При решении задачи распознавания микровыражений одним из важных этапов является этап поиска антропометрических точек лица. Методы, основанные на машинном обучении, в настоящее время набирают все большую популярность. В данной работе рассмотрены два метода обнаружения антропометрических точек: с использованием OpenFace и MediaPipe. Представлено описание этих подходов, и проведен сравнительный анализ полученных результатов на основе среднего отклонения полученных точек от точек на эталонном изображении.

Ключевые слова: микровыражение, распознавание эмоций, наборы данных микровыражений лица, антропометрические точки, OpenFace, MediaPipe, TensorFlow.

FACIAL LANDMARK DETECTION BASED ON OPENFACE AND MEDIAPIPE

V.A. Sablina, A.V. Savin

*Ryazan State Radio Engineering University named after V.F. Utkin,
Russia, Ryazan, sablina.v.a@evm.rsreu.ru*

Abstract. One of the most important stages of the micro-expression recognition is the facial landmark detection. Nowadays it is more and more interesting to apply machine learning methods in this field. In this paper the methods based on OpenFace and MediaPipe frameworks are investigated. The description of these approaches is represented. And the comparative analysis of the obtained results is performed on the basis of the average deviation of detected facial landmarks from the points on the reference image.

Keywords: Micro-expressions, emotion recognition, facial micro-expression datasets, facial landmarks, OpenFace, MediaPipe, TensorFlow.

1. Введение

Современное развитие вычислительной техники позволяет решать множество задач анализа изображений или видеопотока в реальном времени. Одними из таких задач являются задачи определения и анализа микровыражений на лице человека. В настоящее время эта задача остается актуальной, так как микровыражения по своей природе – это очень короткие непроизвольные выражения лица, которые обычно возникают, когда человек сознательно или бессознательно скрывает свои подлинные эмоции. По сравнению с обычными выражениями лица, или макровыражениями, микровыражения обычно длятся очень короткое время, которое составляет менее половины секунды [1]. С учетом этого требуются сложные комплексы с высокочувствительными и быстрыми камерами. Кроме того, на текущий момент очень мало наборов данных (datasets), которые можно было бы использовать для решения задач определения и анализа микровыражений с помощью машинного обучения.

Автоматизированную систему распознавания микровыражений можно представить в виде конвейера, состоящего из следующих основных этапов: предобработка изображения, поиск элементов лица, поиск антропометрических точек, определение эмоций с использованием классификатора [2].

Одним из важных этапов обнаружения микровыражений является этап поиска антропометрических точек. Задача поиска антропометрических точек представляет собой поиск ключевых точек лица, определяющих части лица, такие как нос, рот, глаза подбородок, после этого можно выполнять отслеживание изменений положения этих точек.

Существующие методы поиска антропометрических точек можно классифицировать на несколько категорий. Основными из них являются: целостные методы, методы ограниченной локальной модели ConstrainedLocalModel (CLM) и методы, основанные на регрессии.

Различия между ними заключаются в способе использования информации о внешнем виде и форме лица [2].

2. Методы поиска антропометрических точек

В данной работе для поиска антропометрических точек использовались методы, основанные на OpenFace и MediaPipe. Для решения задачи поиска антропометрических точек в них используются методы машинного обучения (Machine Learning, ML).

MediaPipe – это набор библиотек, методов, предобученных моделей и готовых конвейеров обработки изображений (фреймворк), которые позволяют решать множество задач, связанных с анализом изображений. Фреймворк разрабатывается и поддерживается исследователями и инженерами компании Google. В число задач, которые можно решать с помощью данного фреймворка, входят задачи поиска и определения объектов на изображении, определение положения тела человека на изображении и отслеживание движений в потоке изображений, поиск и распознавание лиц на изображении и другие. Для решения подобных задач в MediaPipe используется библиотека TensorFlow, позволяющая решать задачи с помощью машинного обучения и глубокого обучения (Deep Learning). MediaPipe предоставляет вычислительные блоки (блоки конвейера), из которых можно собирать необходимые конвейеры обработки изображений. На рисунке 1 представлен пример конвейера обработки видеопотока с камеры. Кроме того, существует масса готовых конвейеров, в частности, одним из таких конвейеров обработки изображений является конвейер поиска антропометрических точек. Конвейер для поиска антропометрических точек, который непосредственно использовался в данной работе, позволяет найти 468 точек. Для этих точек после расчета доступны координаты x , y и z [3].

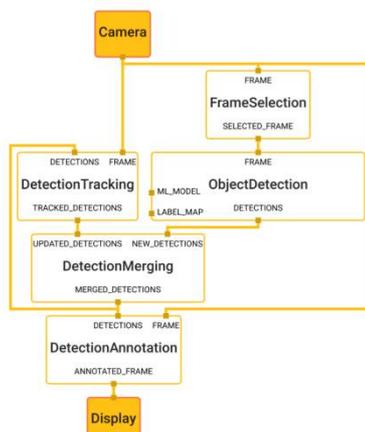


Рис. 1. Пример конвейера MediaPipe для обработки видеопотока

OpenFace представляет собой набор библиотек, методов и предобученных моделей (фреймворк) для анализа и распознавания лиц с использованием глубоких нейронных сетей (deep neural networks). Для решения задачи поиска антропометрических точек в OpenFace используются несколько сторонних библиотек. С помощью фреймворка OpenFace можно найти 68 точек, представленных на рисунке 2. С помощью библиотеки OpenCV на исходном изображении определяется положение лица, а далее с помощью библиотеки dlib и предобученной модели, предоставляемой в рамках фреймворка OpenFace, производится поиск антропометрических точек. Для поиска антропометрических точек лица в dlib используется каскадная регрессия [4, 5].

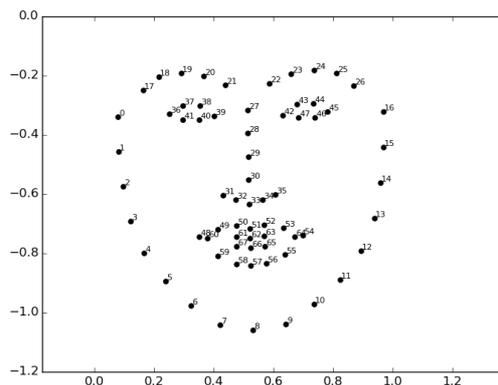


Рис. 2. Набор антропометрических точек, которые может найти OpenFace

3. Набор данных

В данной работе для проведения эксперимента использовался набор изображений (dataset) под названием SAMM (SpontaneousActionsandMicro-Movements). Набор содержит 159 изображений со спонтанными эмоциями и зафиксированными микровыражениями, характеризующими эти спонтанные эмоции. В наборе присутствуют лица мужчин и женщин разного возраста, рас и этнических групп. Данный набор подготовлен группой исследователей из университета Манчестер Метрополитан (ManchesterMetropolitanUniversity). Для создания набора использовалась высокоскоростная и чувствительная камера. Участникам для создания набора данных демонстрировались различные изображения, которые должны были вызывать в них те или иные эмоции, а камера фиксировала изменения в выражении лиц с высокой частотой кадров [6].

4. Описание и результаты эксперимента

Описание структуры эксперимента можно отобразить в виде диаграммы, которая представлена на рисунке 3. Для проведения эксперимента был разработан программный стенд.

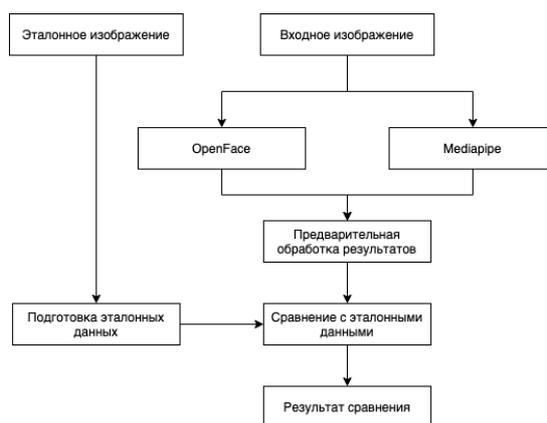


Рис. 3. Диаграмма описания эксперимента

На вход программного стенда передается необработанное входное изображение. Это изображение передается в функции поиска антропометрических точек, одна из которых реализована с использованием OpenFace фреймворка, а другая – с использованием MediaPipe. Результатами работы функций являются массивы точек с координатами (x, y) для OpenFace и с координатами (x, y, z) для MediaPipe. Это координаты антропометрических точек в дву-

мерном или трехмерном пространстве соответственно. Результаты для данного этапа представлены на рисунке 4.



Рис. 4. Результаты поиска антропометрических точек (слева для OpenFace, справа для MediaPipe)

Полученные точки поступают на вход следующего блока конвейера для нормализации, в котором, во-первых, точки приводятся к единой системе координат, а во-вторых, из найденных точек MediaPipe выбираются только те, что соответствуют точкам, найденным OpenFace.

Далее в программный стенд загружается эталонное изображение, которое проходит процесс подготовки данных, а именно происходит выделение эталонных точек. Пример эталонного изображения представлен на рисунке 5. Эталонное изображение подготавливается вручную, путем расстановки точек на элементах лица, на которые должны попадать антропометрические точки. Расстояние между точками настолько мало, что внешне они выглядят как единая линия.



Рис. 5. Пример эталонного изображения

Все полученные данные передаются в блок сравнения, который выдает оценку отклонения найденных с помощью OpenFace и MediaPipe точек от эталонных линий. Расчет этого отклонения производится следующим образом. Выбирается очередная точка, полученная с помощью OpenFace. Ее координаты сравниваются с точками эталонного изображения. Для всех точек p_r эталонного (reference) изображения с координатами (x_r, y_r) и выбранной точки p_i с координатами (x_i, y_i) рассчитывается расстояние по формуле (1). Из полученных расстояний выбирается минимальное (2). Это минимальное расстояние принимается за показатель отклонения найденных с помощью OpenFace точек от точек эталонного изображения. После нахождения отклонений для всех точек, найденных на входном изображении, рассчитывается среднее отклонение (3).

$$S_{ir} = \sqrt{(x_r - x_i)^2 + (y_r - y_i)^2}; \quad (1)$$

$$S_{imin} = \Delta_i = \min_r(S_{ir}); \quad (2)$$

$$\Delta = \frac{\sum_{i=0}^n \Delta_i}{n}. \quad (3)$$

Аналогичный процесс повторяется также для точек, найденных с помощью MediaPipe.

В результате проведения эксперимента было обработано несколько изображений из набора SAMM для получения показателя среднего отклонения по каждому из изображений. Результаты эксперимента для нескольких изображений представлены в таблице 1.

Таблица 1. Средние отклонения антропометрических точек лица от эталонных линий для различных изображений

Номер изображения	OpenFace, Δ	MediaPipe, Δ
1	4,3257	3,1109
2	5,0118	3,8744
3	3,1051	2,9115
4	3,5296	4,1308

5. Выводы

В статье представлен сравнительный анализ двух подходов к обнаружению антропометрических точек лица.

При сравнении полученных результатов видно, что отклонение точек, полученных с помощью MediaPipe, в большинстве случаев меньше, чем отклонение точек, полученных с помощью OpenFace. Кроме того, MediaPipe предоставляет намного больше точек, а также определяет координату точки по оси Z, что позволяет использовать эти данные при распознавании микровыражений.

Кроме того, необходимо отметить, что фреймворк OpenFace в настоящее время практически не развивается в отличие от фреймворка MediaPipe, над которым работают исследователи и инженеры из Google.

Библиографический список

1. Paul Ekman, *Emotion in the Human Face*, 2nd Edition, Malor Books, 2013, 456 p.
2. Мельник О.В., Саблина В.А., Савин А.В., Борщев А.Б. Обнаружение антропометрических точек лица на основе методов глубокого обучения с целью распознавания эмоций // Биомедицинская радиотехника, том 23, № 3. – М.: Издательство «Радиотехника», 2020. – С. 45-52.
3. Camillo Lugaresi, Jiuqiang Tang, Hadon Nash, Chris McClanahan, EshaUboweja, Michael Hays, Fan Zhang, Chuo-Ling Chang, Ming Guang Yong, Juhyun Lee, Wan-Teh Chang, Wei Hua, Manfred Georg, and Matthias Grundmann "MediaPipe: A Framework for Building Perception Pipelines" in arXiv:1906.08172, 2019, 9 p. [accessed 2021 March 27], <http://arxiv.org/abs/1906.08172>.
4. Brandon Amos, BartoszLudwiczuk, and MahadevSatyanarayanan, "Openface: A General-Purpose Face Recognition Library with Mobile Applications", CMU-CS-16-118, CMU School of Computer Science, Tech. Rep., 2016, 18 p.
5. TadasBaltrušaitis, Peter Robinson, and Louis-PhilippeMorency, "OpenFace: An Open Source Facial Behavior Analysis Toolkit,"2016 IEEE Winter Conference on Applications of Computer Vision (WACV), Lake Placid, NY, USA, 2016, pp. 1-10.
6. Adrian K. Davison, Cliff Lansley, Nicholas Costen, Kevin Tan, and MoiHoon Yap, "SAMM: A Spontaneous Micro-Facial Movement Dataset," in IEEE Transactions on Affective Computing, Volume 9, No. 1, 2018, pp. 116-129.

УДК 004.8; ГРНТИ 28.23.15

ГЕНЕРАТИВНЫЙ ДИЗАЙН С ИСПОЛЬЗОВАНИЕМ ИНТЕЛЛЕКТУАЛЬНЫХ ТЕХНОЛОГИЙ

М.А. Титова, А.Ю. Громов

*Рязанский государственный радиотехнический университет имени В.Ф. Уткина,
Российская Федерация, Рязань, gromovrm@gmail.ru*

Аннотация. В работе рассматриваются интеллектуальные технологии для генеративного проектирования, выявлены недостатки предлагаемых решений и предложен алгоритм, обеспечивающий качество, разнообразие и эстетику генерируемых трёхмерных моделей.

Ключевые слова: генеративный дизайн (GD), сверточная нейронная сеть (CNN), генеративная состязательная сеть (GAN), оптимизация топологии.

GENERATIVE DESIGN USING INTELLIGENT TECHNOLOGIES

M.A. Titova, A.Y. Gromov

*Ryazan State Radio Engineering University named after V.F. Utkin,
Russia, Ryazan, gromovrm@gmail.ru*

The summary. The paper deals with intelligent technologies for generative design, identified the shortcomings of the proposed solutions and the algorithm that provides quality, variety and aesthetics generated three-dimensional models was proposed

Keywords: generative design (GD), convolutional neural network (CNN), generative adversarial network (GAN), topology optimization.

Генеративный дизайн, порождающий дизайн, или генеративное проектирование (GD) — подход к проектированию и дизайну цифрового или физического продукта, при котором человек делегирует часть процессов компьютерным технологиям и платформам [1].

GD в технических системах состоит в генерации большого количества начальных дизайнов с целью рационального использования материалов, придания оптимальной формы физическим объектам с учётом требуемых технических характеристик [2]. Он позволяет значительно увеличить эффективность производства, сократить трудозатраты инженеров на этапе технического проектирования, повысить надёжность продукции и понизить стоимость изменения прототипов конструкций.

Исследовательская компания Market Data Forecast прогнозирует, что рынок генеративного дизайна увеличится со 141 млн. долл. США в 2019 году до 315 млн долл. в 2025 году, а среднегодовой темп роста составит выше 19,9%. Ожидается, что рынок будет набирать скорость как за счет внедрения технологии в уже традиционных для нее отраслях (промышленность, автомобильная и аэрокосмическая отрасль, медицина, архитектура), так и масштабного расширения в другие индустрии [3].

В настоящее время установилась тенденция использования интеллектуальных технологий в области генеративного дизайна. Появились подходы глубокого обучения, включающие сверточную нейронную сеть (CNN) и генеративную состязательную сеть (GAN), а также их разновидности, которые успешно применяются в комплексе с топологической оптимизацией. Однако не все подходы гарантируют создание большого набора сгенерированных вариантов дизайна, которые не только соответствуют техническим требованиям проектируемой модели, но и являются эстетичными.

В другом подходе, который включал этапы создания эскиза дизайна и его инверсии, объединяющем GAN и глубокую нейронную сеть, в качестве результата пользователь имел лишь концептуальные дизайны [4]. Механический анализ соответствия модели требованиям не был включён в алгоритм генеративного дизайна и технические характеристики невозможно гарантировать.

Кроме того, полученные генеративные конструкции часто имеют органические формы, которые нецелесообразно производить традиционными методами, поэтому аддитивное производство является важной разработкой для GD [5], однако только 2 из 37 систем генера-

тивного дизайна, рассмотренные в работе [6], учитывают критерии, необходимые для создания модели, адаптированной к 3D-печати.

Таким образом, в данной статье будет разработан алгоритм, который обеспечит соответствие эстетических характеристик и технических ограничений модели, сгенерированной в трёхмерном формате.

Архитектура предлагаемого решения состоит из повторяющихся блоков исследования проекта и оценки проекта, представленная на рисунке.

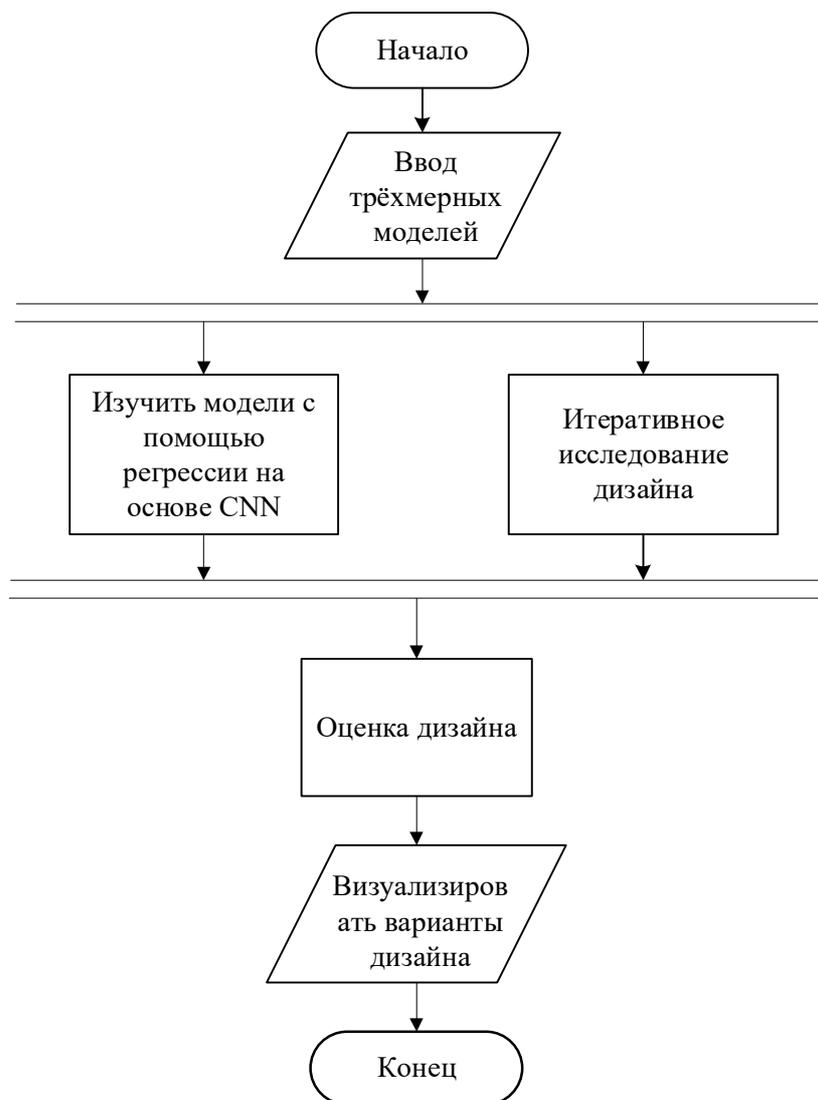


Рис. Алгоритм реализации итеративного подхода к генеративному дизайну

Итеративное исследование дизайна включает в себя создание большого количества новых проектов итеративно с использованием небольшого количества предыдущих проектов. Оценка дизайна включает количественную оценку новизны сгенерированных проектов по сравнению с предыдущими проектами и визуализацию вариантов дизайна [7,8].

3D-проекты необходимо преобразовать в 1D, прежде чем их можно будет использовать в алгоритмах машинного обучения. Трёхмерные конструкции подаются на вход в GAN, включающую трёхмерную CNN, и тем самым преобразуются в одномерные векторы.

CNN — это глубокие искусственные нейронные сети, которые в основном используются для классификации изображений, кластеризации их в классы или выполнения данных

регрессионного отображения. Эти сети запоминают соотношение между пикселями или вокселями в структуре изображения или трёхмерной модели и используют эти знания для классификации новых данных [9].

Следующий этап включает изучение моделей для оценки новизны дизайна по сравнению с предыдущими моделями. Регрессия на основе CNN может предсказать сходство, которое противоречит новизне. Данный этап может протекать параллельно предыдущим этапам. После чего варианты проектирования, полученные из итеративного исследования проекта, должны быть оценены на основе различных атрибутов дизайна, которые необходимы для инженеров. Определяется не только новизна сгенерированных моделей, но и другие физические и технические характеристики.

На последнем этапе результаты проектирования визуализируются, представляя пользователю набор оптимизированных моделей.

Таким образом, предлагаемая структура может контролировать сходство с эталонными конструкциями и техническими характеристиками как многоцелевая функция. Кроме того, описанный итеративный процесс, в котором проводится оптимизация топологии для создания обучающих данных для генеративных моделей, а выходные данные из генеративных моделей снова используются в качестве эталонных для оптимизации топологии, может повысить качество полученных моделей. На этапе оценки дизайна могут быть реализованы не только основные технические характеристики, но и расчёт стоимости аддитивного производства, расчёт стоимости материала, затрат на обработку и удаление опор при 3D-печати.

Однако качество дизайнов сильно зависит от обучающих данных, которые трудно получить и которые требуют много вычислительного пространства, поэтому в дальнейшем планируется найти способ решения выявленной проблемы и усовершенствовать алгоритм генерации.

Библиографический список

1. Княгинин В.Н. Промышленный дизайн Российской Федерации: возможность преодоления "дизайн-барьера": учебное пособие для студентов высших учебных заведений, обучающихся по направлению подготовки магистров "Инноватика" / В. Н. Княгинин. - Санкт-Петербург : Изд-во Политехнического ун-та, 2012. - 79 с.
2. The Next Wave Of Intelligent Design Automation: Briefing Paper. / Harvard Business Review Analytic Services. – 2018. – 8 p
3. Global Generative Design Market Analysis – Segmentation by Application (Product Design & Development and Cost Optimization), Component (Software and Services), End-User (Automotive, Aerospace & Defense, Industrial Manufacturing), and Region - Forecast to 2025: market report : [Market and Markets]. – 2020. – TC 9845. – 150 p.
4. Yong D. New Product Design with Automatic Scheme Generation /Yong Dai¹, Yi Li², Li Jun Liu//Sensing and Imaging – 2019. – 16 p.
5. Schafstall H. Bridging the Gap Between Design and Additive Manufacturing Using Smart Generative Design / H. Schafstall, R. Dua // E-book: Additive Manufacturing 2020. – 2020. – p. 5-8.
6. Singh V. Towards an integrated generative design framework / Vishal Singh, Ning Gu // Design studies, 33(2) – 2012. - 185–207 p.
7. Oh S. Deep Generative Design: Integration of Topology Optimization and Generative Models / Sangeun Oh, Yongsu Jung, Ikjin Lee, Namwoo Kang // Journal of Mechanical Design. – 2019. – Vol. 141, iss. 11, №111405. – 13 p.
8. Bhat P. Automated Learning of User Preferences for Selection of High Quality 3D Designs / Prashant Bhat, Iaroslav Shcherbatyi, Wolfgang Maass // 29th CIRP Design Conference 2019, Póvoa de Varzim, Portugal, 08-10 May 2019 / ed. G. D. Putnik. – 2019. – p. 814-819.
9. Rawat S. A Novel Topology Optimization Approach using Conditional Deep Learning / Sharad Rawat, M.-H. Herman Shen // Free service and archive for articles «arXiv» of Cornell University. – Publ. 2019-01-14. – 8 p.

УДК 004.65; ГРНТИ 20.23.17

ОБЗОР И АНАЛИЗ ПРИМЕНЕНИЯ ПОЛНОТЕКСТОВЫХ ИНДЕКСОВ В MSSQL

Е.А. Благодаров, В.Ю. Тарасова

*Рязанский государственный радиотехнический университет имени В.Ф. Уткина,
Российская Федерация, Рязань, egorblagodarov30@gmail.com, _valentina2008.91@mail.ru*

Аннотация. В работе рассматривается применение полнотекстовых индексов, а также сравнение быстродействия поиска текстовой информации в базе данных без индексов и с их использованием. Разработка ведется в системе управления базами данных MSSQL.

Ключевые слова: система управления базами данных (СУБД), базы данных (БД), индексы, полнотекстовые индексы.

OVERVIEW AND ANALYSIS OF THE USE OF FULL-TEXT INDEXES IN MS SQL

Е.А. Blagodarov, V.Yu. Tarasova

*Ryazan State Radio Engineering University named after V.F. Utkin,
Russia, Ryazan, egorblagodarov30@gmail.com, _valentina2008.91@mail.ru*

The summary. The paper considers the use of full-text indexes, as well as a comparison of the speed of searching for text information in a database without indexes and with their use. The development is carried out in the MS SQL database management system.

Keywords: Database management System(DBMS), databases(DB), indexes, full-text indexes.

Поиск текстовой информации – одна из основных проблем в современном мире. Появление большого числа информационных ресурсов, а также перевод бумажных носителей в электронный вид приводят к созданию различных способов поиска информации как в сети интернет, так и в СУБД. Для ускоренного поиска в СУБД используются специальные объекты – индексы, которые представляют собой структуру данных, состоящую из ключей, построенных на основе одного или нескольких столбцов таблицы или представлений, и указателей, которые сопоставляются с местом хранения заданных данных [1]. В СУБД Microsoft SQL Server для поиска текстовой информации применяются полнотекстовые индексы (Full-text).

Полнотекстовый поиск – это поиск слов или целых фраз в текстовых данных, путём их обработки по правилам заданного языка [2]. Наиболее часто он применяется при поиске в больших объёмах данных, например, если таблица содержит миллион строк и более. Такой запрос будет возвращать все строки, содержащие хотя бы одно совпадение. Совпадение формируется, при наличии в исследуемой строке всех терминов, которые были указаны в запросе, а также при соответствии другим параметрам поиска.

Возможности полнотекстового поиска в СУБД MS SQL Server:

- поиск по префиксным выражениям, то есть по началу заданного слова или фразы;
- поиск с учётом расположения слов;
- поиск синонимичных слов;
- поиск с помощью словоформ;
- поиск с учётом «стоп-слов», поиск по словам, которые указаны в таком списке, производиться не будет;
- поиск с учётом рейтинга слов [1].

Под словоформой понимаются формы глаголов, формы единственного или множественного числа существительного. Поиск с учетом рейтинга слов означает, что при наличии в запросе нескольких слов, у них возможно указать позицию важности от 0,0 до 1,0 (где 0,0 – это наименее необходимое слово при поиске, 1,0 – это самое важное слово во фразе).

Порядок создания полнотекстового индекса

1. Создадим новую БД Library (Библиотека), в которую внесём таблицу Books (Книги), в ней два столбца ID_Books (идентификатор) Paragraph (обозначает абзацы книги). В таблице Books хранится определенное литературное произведение, разделённое на абзацы. На рисунке 1 приведена заполненная таблица Books.

ID_Books	Paragraph
1	- Что, Петр, не видать еще? – спрашивал 20?го мая 1859 года, выходя без шапки на низкое крылечко постоялого двора на "" шоссе, бар...
2	Слуга, в котором все: и бирюзовая сережка в ухе, и напояженные разноцветные волосы, и учтивые телодвижения, словом, все изобли...
3	- Не видать? – повторил барин.
4	- Не видать, – вторично ответил слуга.
5	Барин вздохнул и присел на скамеечку. Познакомим с ним читателя, пока он сидит, подогнувши под себя ножки и задумчиво поглядыва...
6	Зовут его Николаем Петровичем Кирсановым. У него в пятнадцати верстах от постоялого дворика хорошее имение в двести душ, или, к...
7	Слуга, из чувства приличия, а может быть, и не желая остаться под барским глазом, зашел под ворота и закурил трубку. Николай Петро...
8	- Никак они едут?с. – доложил слуга, вынырнув из?под ворот.
9	Николай Петрович вскочил и устремил глаза вдоль дороги. Показался тарантас, запряженный тройкой ямских лошадей; в тарантасе ме...
10	- Аркаша! Аркаша! – закричал Кирсанов, и побежал, и замаяхал руками.. Несколько мгновений спустя его губы уже прильнули к безбород...
11	- Дай же отряхнуться, папаша. – говорил несколько сиплым от дороги, но звонким юношеским голосом Аркадий, весело отвечая на отд...
12	- Ничего, ничего, – твердил, умиленно улыбаясь. Николай Петрович и раза два ударил рукою по воротнику сыновней шинели и по собств...
13	Николай Петрович казался гораздо встревоженнее своего сына; он словно потерялся немного, словно робел. Аркадий остановил его.
14	- Папаша, – сказал он, – позволь познакомить тебя с моим добрым приятелем, Базаровым, о котором я тебе так часто писал. Он так лю...
15	Николай Петрович быстро обернулся и, подойдя к человеку высокого роста в длинном балахоне с кистями, только что вылезшему из та...
16	- Душевно рад, – начал он, – и благодарен за доброе намерение посетить нас; надеюсь... позвольте узнать ваше имя и отчество?
17	- Евгений Васильев, – отвечал Базаров ленивым, но мужественным голосом и, отвернув воротник балахона, показал Николаю Петрович...
18	- Надеюсь, любезнейший Евгений Васильич, что вы не соскучитесь у нас, – продолжал Николай Петрович.
19	Тонкие губы Базарова чуть тронулись: но он ничего не отвечал и только приподнял фуражку. Его темно?белокорые волосы, длинные и гу...

Рис. 1. Заполненная таблица Books

2. Создадим полнотекстовый каталог для таблицы Books. Перед тем как создать полнотекстовый индекс необходимо создать хранилище, то есть полнотекстовый каталог. Данный каталог будет являться виртуальным объектом, поэтому не будет принадлежать никакой файловой группе. На рисунке 2 изображено создание полнотекстового каталога.

```

CREATE FULLTEXT CATALOG TestCatalog -- команда создания полнотекстового каталога
WITH ACCENT_SENSITIVITY = ON -- указание на использование или неиспользование
--диакритических знаков при поиске, по умолчанию – ON
AS DEFAULT -- указание, что данный каталог является каталогом по умолчанию
AUTHORIZATION dbo -- устанавливает владельца каталога (БД или пользователь)
GO

```

Рис. 2. Создание полнотекстового каталога

3. Создадим полнотекстовый индекс для таблицы Books (рисунок 3). Индекс создается либо по таблице, либо по представлению. Представление – это объект БД, представляющий собой виртуальную таблицу, созданной на основе одной (или нескольких) базовых таблиц [3]. Требования, предъявляемые к таблице или представлению:

- уникальный адрес;
- хотя бы один столбец, содержащий ограничение notnull, например первичный ключ;
- только один полнотекстовый индекс;
- типы данных для столбцов, на которых создаются индексы: char, varchar, nchar, nvarchar, text, ntext, image, xml, varbinary или varbinary(max).

```

--СОЗДАНИЕ ПОЛНОТЕКСТОВОГО ИНДЕКСА
CREATE FULLTEXT INDEX ON Books(Paragraph) -- команда создания полнотекстового индекса,
-- ON указывает на то, для какой таблицы мы будем создавать индекс
KEY INDEX PK_Books_A66EBB997F60ED59 ON (TestCatalog) -- указание уникального индекса таблицы,
-- для которой создаётся полнотекстовый индекс. ON указывает на каталог,
-- в котором мы создаём полнотекстовый индекс
WITH (CHANGE_TRACKING AUTO) -- указание на то, что при любом изменении в таблице Terms,
-- изменения отобразятся в индексе автоматически (CHANGE_TRACKING MANUAL - ручное изменение).
GO

```

Рис. 3. Создание полнотекстового индекса

Запросы на полнотекстовый поиск

1. CONTAINS и FREETEXT– предикаты, которые используются в предложении WHERE (или HAVING) в инструкции SELECT для осуществления полнотекстового поиска (точного и неточного) в MSSQL. Они осуществляют поиск по столбцам с символьным типом данных. Данные предикаты производят поиск по:

- целым словам и фразам;
- префиксам, как слов, так и фраз;
- словам в окрестностях других слов;
- однокоренным словам;
- синонимичным словам.

Предикаты объединяются с другими предикатами, к примеру, LIKE или BETWEEN. Для реализации поиска с этими предикатами используются функции с различными аргументами: WEIGHT, FORMSOF, NEAR.

Пример 1. Найти и вывести все строки, где присутствует какая-либо форма слова «добрый». Результат выполнения запроса – рисунок 4.

```

SELECT Paragraph
FROM Books
WHERE CONTAINS (Paragraph, 'FORMSOF (INFLECTIONAL, добрый)')

```

Paragraph
1 - Папаша, - сказал он, - позволь познакомиться тебе с моим добрым приятелем, Базаровым, о котором я тебе так часто писал. Он так любезен, что согласился погостить у нас.
2 - Душевно рад, - начал он, - и благодарен за доброе намерение посетить нас; надеюсь... позвольте узнать ваше имя и отчество?
3 - Добрый папаша!
4 - Полно, папаша, полно, сделай одолжение! - Аркадий ласково улыбнулся. «В чем извиняется!» - подумал он про себя, и чувство снисходительной нежности к добродушному и мягкому отцу, смешанное с ош...
5 Павел Петрович вынул из кармана панталон свою красивую руку с длинными розовыми ногтями, - руку, казавшуюся еще красивей от снежной белизны рукавчика, застегнутого одиноким крупным оп...
6 Базаров ушел, а Аркадием овладело радостное чувство. Сладко засыпать в родимом доме, на знакомой постеле, под одеялом, над которым трудились любимые руки, быть может руки няношки, те ласк...
7 - Мы познакомились, отец! - воскликнул он с выражением какого-то ласкового и доброго торжества на лице. - Федосья Николаевна точно сегодня не совсем здорова и придет попозже. Но как же ты...
8 Понемногу она стала привыкать к нему, но все еще робела в его присутствии, как вдруг ее мать Арина умерла от холеры. Куда было деваться Фенечке? Она наследовала от своей матери любовь к по...
9 - Надо серебряных тополей побольше здесь сажать, да елок, да, пожалуй, липок, подбавивши чернозему. Вон беседка принялась хорошо, - прибавил он, - потому что акация да сирень - ребята добры...
10 - И добры мужички надуют твоего отца всенепременно. Знаешь поговорку: «Русский мужик бога сполает».
11 - Твой отец добрый маленький, - промолвил Базаров, - но он человек отставной, его песенка спета.
12 И он посмотрел кругом, как бы желая понять, как можно не сочувствовать природе. Уже вечерело; солнце скрылось за небольшую осиновою рощу, лежавшую в полверсте от сада; тень от нее без кон...
13 - Плетка дело доброе, - заметил Базаров, - только мы вот добрались до последней капли...
14 - Деревья в лесу, - повторила она. - Стало быть, по твоему, нет разницы между глупым и умным человеком, между добрым и злым?
15 - По крайней мере, при правильном устройстве общества совершенно будет равно, глуп ли человек или умен, зол или добр...
16 Как все женщины, которым не удалось полюбить, она хотела чего-то, сама не зная, чего именно. Собственно, ей ничего не хотелось, хотя ей казалось, что ей хотелось всего. Покойного Одинцова она...
17 «Странный человек этот лекарь?» - думала она, лежа в своей великолепной постеле, на кружевных подушках, под легким шелковым одеялом... Анна Сергеевна наследовала от отца часть его наклон...
18 Время (дело известное) летит иногда птицей, иногда ползет червяком; но человеку бывает особенно хорошо тогда, когда он даже не замечает - скоро ли, тихо ли оно проходит. Аркадий и Базаров имен...
19 Анна Сергеевна тихонько пошла вслед за ним и, подозвав Катю, взяла ее под руку. Она не расставалась с ней до самого вечера. В карты она играть не стала и все больше посмеивалась, что вовсе не

Рис. 4. Результат запроса

Пример 2. Найти и вывести все строки, где присутствует слова «Николай» и «улыбнулся» на некотором удалении. Результат выполнения запроса – рисунок 5.

```

SELECT Paragraph

```

FROMBooks
WHERECONTAINS (Paragraph, 'Николай NEAR улыбнулся')

Paragraph	
1	Николай Петрович представил его Базарову: Павел Петрович слегка наклонил свой гибкий стан и слегка улыбнулся, но руки не подал и даже положил ее...

Рис. 5. Результат запроса

2. CONTAINSTABLE и FREETEXTTABLE– функции инструкции SELECT для осуществления полнотекстового поиска (точного и неточного) в MSSQLServer. Такие функции возвращают таблицу из неопределенного количества строк (возможно нулевое число строк), при этом возвращается определенное значение, а не просто слово или фраза. Как правило, данные функции применимы для FROM, они могут участвовать в соединении таблиц. Эти функции реализуют такой же поиск, что и предикаты CONTAINS и FREETEXT, но отличаются тем, что функции возвращают строки, сортируя их по релевантности (наличие ранга релевантности).

Пример 3. Найти и вывести не более пяти строк, содержащих фразу «по его словам», отсортировать их по рангу релевантности. Результат выполнения запроса – рисунок 6.

```
SELECT Books.Paragraph,
       NewTable.[KEY],
       NewTable.[RANK]
FROM Books JOIN CONTAINSTABLE(Books, Paragraph, "'по его словам'", 5)
AS NewTable
ON Books.ID_Books=NewTable.[KEY]
```

Paragraph			KEY	RANK
1	- Приятель Аркаши, очень, по его словам, умный человек.	123	160	
2	Слуга вошел и доложил о приезде председателя казенной палаты, сладког...	577	53	

Рис. 6. Результат запроса

Принципиальное отличие CONTAINSTABLE от FREETEXTTABLE и CONTAINS от FREETEXT заключается в более сложном и более универсальном принципе работы FREETEXT и FREETEXTTABLE. Они проводят поиск текста с анализом его смыслового содержания, за счёт чего результатом будут наиболее подходящие строки.

Пример 4. Возьмём условие из примера 2 и решим его через предикат FREETEXT, результат выполнения представлен на рисунке 7.

```
SELECT Paragraph
FROM Books
WHERE FREETEXT (Paragraph, 'Николай NEAR улыбнулся')
```

Paragraph	
1	Зовут его Николаем Петровичем Кирсановым. У него в пятнадцать верстах от постоялого двора хорошее имение в двести душ, или, как он выража...
2	Слуга, из чувства приличия, а может быть, и не желая остаться под барским глазом, зашел под ворота и закурил трубку. Николай Петрович поник гол...
3	Николай Петрович вскочил и устремил глаза вдоль дороги. Показался тарантас, запряженный тройкой ямских лошадей; в тарантасе мелькнул околы...
4	– Ничего, ничего, – твердил, умиленно улыбаясь, Николай Петрович и раза два ударил рукою по воротнику сыновней шинели и по собственному пальт...
5	Николай Петрович казался гораздо встревоженнее своего сына; он словно потерялся немного, словно робел. Аркадий остановил его.
6	Николай Петрович быстро обернулся и, подойдя к человеку высокого роста в длинном балахоне с кистями, только что вылезшему из тарантаса, креп...
7	– Евгений Васильев, – отвечал Базаров ленивым, но мужественным голосом и, отвернув воротник балахона, показал Николаю Петровичу все свое ли...
8	– Надеюсь, любезнейший Евгений Васильич, что вы не соскучитесь у нас, – продолжал Николай Петрович.
9	– Так как же, Аркадий, – заговорил опять Николай Петрович, оборачиваясь к сыну, – сейчас закладывать лошадей, что ли? Или вы отдохнуть хотите?
10	– Я здесь с коляской, но и для твоего тарантаса есть тройка, – хлопотливо говорил Николай Петрович, между тем как Аркадий пил воду из железного...
11	Кучер Николая Петровича вывел лошадей.
12	– Живей, живей, ребята, подсобляйте, – воскликнул Николай Петрович, – на водку будет!
13	– Так вот как, наконец ты кандидат и домой приехал, – говорил Николай Петрович, потрогивая Аркадия то по плечу, то по колену. – Наконец!
14	Аркадий живо повернулся к отцу и звонко поцеловал его в щеку. Николай Петрович тихонько засмеялся.
15	– А! он по медицинскому факультету, – заметил Николай Петрович и помолчал. – Петр, – прибавил он и протянул руку, – это никак наши мужики едут?
16	– Ухораты у меня большие с мужиками в нынешнем году, – продолжал Николай Петрович, обращаясь к сыну. – На протат оброка. Что ты билась, лала...

Рис. 7. Результат запроса

Этот запрос вывел строки, которые содержат не только конкретные слова, указанные в запросе, но и строки, подходящие по общему смыслу, например, рисунок 8.

– Ничего, ничего, – твердил, умиленно улыбаясь, Николай Петрович и раза два ударил рукою по воротнику сыновней шинели и по собственному пальто. – Покажи?ка себя, покажи?ка, – прибавил он, отодвигаясь, и тотчас же пошел торопливыми шагами к постоялому двору, приговаривая: «Вот сюда, сюда, да лошадей поскорее».

Рис. 8. Текст строки 4 полученной таблицы

Сравнение с оператором LIKE

Поиск слов в тексте можно реализовать с помощью оператора LIKE. Он позволяет находить заданные слова и фразы в строке, однако его возможности ограничены. Данный оператор реализует поиск по вхождению конкретного символа, то есть он не производит анализ текста для нахождения словоформ или синонимов. Для реализации поиска нескольких слов на удалении необходимо использовать логические связки, что сказывается на быстродействии. Кроме того оператор LIKE просматривает каждый символ, не имея определенный банк слов (индекс формируют свой словарь, в котором хранятся все данные из строк), что занимает больше времени, особенно при большом числе строк (> 1000). Фактически, можно сказать, что LIKE позволяет просто найти слово или фразу по их полному или частичному указанию, но не реализует какие-либо модификации с ними. В то же время стоит сказать, что полнотекстовые индексы имеют свои недостатки, например, невозможность сортировки результатов и большая затрата ресурсов на выполнение запросов (при более маленьком числе строк).

На основе вышесказанного можно сказать, что полнотекстовые индексы в MSSQL позволяют производить поиск слов и целых фраз в больших по объёму текстах. Некоторые функции и предикаты, реализованные для данного поиска, позволяют анализировать текст по смысловой нагрузки и находить не только конкретные поисковые слова, но и синонимичные пары, похожие по смыслу фразы и различные формы конкретных слов.

Библиографический список

1. Основы индексов в MicrosoftSQLServer [Электронный ресурс]: блог о компьютерах и программировании – режим доступа: <https://info-comp.ru/programirovanie/575-index-basics-in-ms-sql-server.html>.
2. Microsoft SQL Server. Основы полнотекстового поиска (Full-TextSearch) [Электронный ресурс]: блог о Microsoft SQL Server, T-SQL, DBA, Developer – режим доступа: <http://sqlcom.ru/helpful-and-interesting/full-text-search-main/>.
3. Компонент Full-textSearch [Электронный ресурс]: документация по SQL – режим доступа: <https://docs.microsoft.com/ru-ru/sql/relational-databases/search/full-text-search?view=sql-server-ver15>.

УДК 65.011.56; ГРНТИ 50.49.37

ИНТЕГРАЦИЯ ПРОДУКТОВ 1С С КОНТРОЛЬНО-КАССОВОЙ ТЕХНИКОЙ

В.О. Кривоzubов

*Рязанский государственный радиотехнический университет имени В.Ф. Уткина,
Российская Федерация, Рязань, krivozubov_sl@mail.ru*

Аннотация. В работе рассматриваются возможности интеграции продуктов 1С с контрольно-кассовой техникой, существующие аналоги, пример разработки собственной товаро-учетной системы с подключением фискального регистратора Атол 30Ф.

Ключевые слова: контрольно-кассовая техника (ККТ), онлайн-касса, фискальный регистратор, технология 1С.

INTEGRATION OF 1C PRODUCTS WITH CASH REGISTER EQUIPMENT

V.O. Krivozubov

*Ryazan State Radio Engineering University named after V.F. Utkin,
Russia, Ryazan, krivozubov_sl@mail.ru*

Annotation. The paper considers the possibilities of integrating 1C products with cash register equipment, existing analogues, an example of developing a proprietary commodity accounting system with the connection of the Atol 30F fiscal registrar.

Keywords: cash register equipment, online sales register, fiscal registrar, 1C technology.

В настоящее время все индивидуальные предприниматели и юридические лица, осуществляющие продажу товаров обязаны применять контрольно-кассовую технику (54-ФЗ «О применении контрольно-кассовой техники»[3]).

ККТ. Основные положения

Контрольно-кассовая техника (ККТ) – это торговое оборудование, которое применяется для проведения расчетов наличными или электронными средствами (платежными картами) с покупателями.

К кассовым аппаратам предъявляются следующие требования:

- мобильность;
- многофункциональность;
- надежность.

Применяемая контрольно-кассовая техника должна:

- быть зарегистрирована в налоговых органах по месту учета;
- иметь фискальную память;
- обеспечивать формирование и передачу фискальных документов.

Согласно закону №290-ФЗ от 03.07.2016 [4], индивидуальные предприниматели и юридические лица при продаже товаров должны обеспечивать:

- выдачу покупателю кассового чека после совершения покупки, а также возможность отправки его электронной версии на e-mail;
- передачу информации о расчетах через операторов фискальных данных (ОФД) в Федеральную налоговую службу (ФНС) России;
- применение фискального накопителя с возможностью его самостоятельной замены.

Изменения порядка применения ККТ вводились постепенно. С 1 февраля 2021 года для всех юридических лиц и индивидуальных предпринимателей действует одинаковый набор обязательных реквизитов чека ККТ (рис.1).

При отсутствии хотя бы одного обязательного реквизита чек признается полностью недействительным.

ККТ, удовлетворяющие вышеперечисленным требованиям, называют также онлайн-кассами.

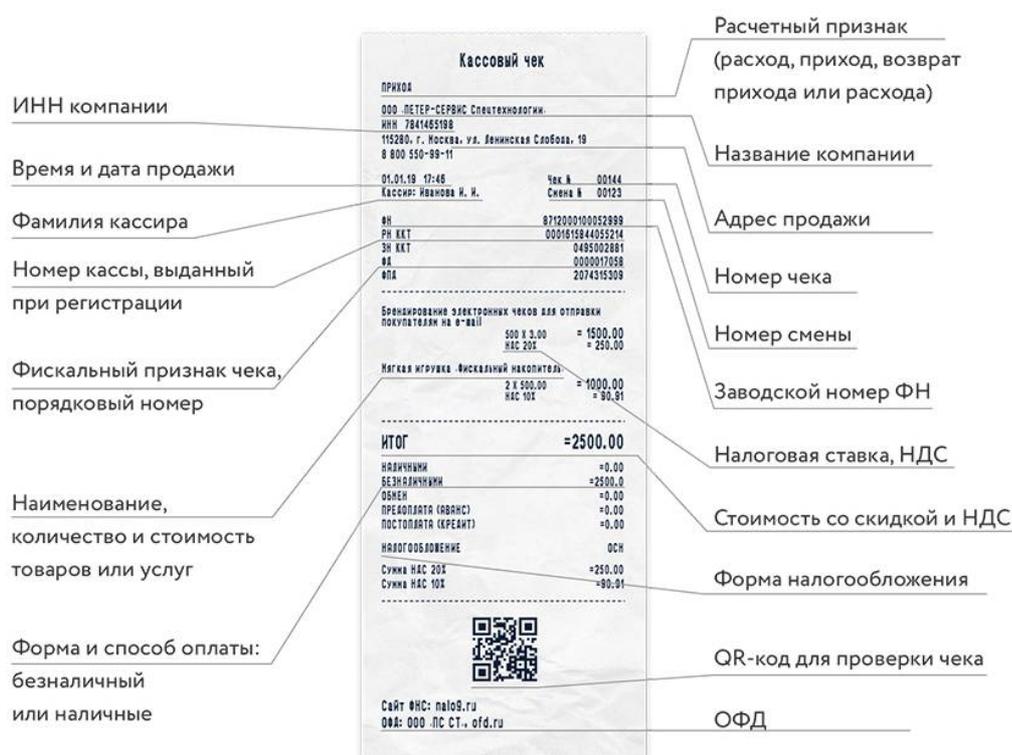


Рис. 1. Пример печатного чека с набором обязательных реквизитов

Технология 1С

В связи с необходимостью наличия ККТ на предприятии торговли возникает необходимость в ее интеграции с системами ведения учета. В настоящее время большой популярностью пользуется программное обеспечение на платформе 1С.

Основными преимуществами использования технологии 1С являются [2]:

- автоматизация предприятий («1С:Предприятие» является универсальной системой автоматизации деятельности предприятия. Конфигурация создается и сопровождается штатными средствами системы, но может быть изменена, дополнена пользователем системы, а также разработана заново);
- онлайн-касса и 1С (совмещение онлайн-кассы с системой 1С позволяет формировать электронные чеки, документы, управлять ассортиментом магазина);
- расчет и уплата налогов - залог успеха бизнеса.

Внедрение системы 1С в организации обеспечивает повышение эффективности ведения товарного и бухгалтерского учета. Система обладает отличным функционалом. Повышается не только скорость выполнения работы специалистами организации, но и качество результатов.

Анализ существующих решений

Существует уже немало готовых решений для интеграции ККТ с 1С. Все эти решения условно можно поделить на:

- системы товарного учета;
- системы бухгалтерского учета;
- смешанные системы.

Однако все они имеют существенные недостатки. Рассмотрим их на примере двух

систем: товароучетной «1С:Розница 8 ПРОФ» и бухгалтерной «1С:Бухгалтерия 8 ПРОФ».

Таблица 1. Сравнительная характеристика продуктов 1С

«1С:Розница 8 ПРОФ»	«1С:Бухгалтерия 8 ПРОФ»
<ul style="list-style-type: none"> • объемна и многофункциональна 	<ul style="list-style-type: none"> • объемна и многофункциональна
<ul style="list-style-type: none"> • для полноценного пользования системой приложенного руководства недостаточно 	<ul style="list-style-type: none"> • для работы необходимо пройти обучение
<ul style="list-style-type: none"> • не предусмотрен ряд функций 	<ul style="list-style-type: none"> • для эффективной работы программы требуются индивидуальные решения по автоматизации некоторых процессов
<ul style="list-style-type: none"> • высокая стоимость 	<ul style="list-style-type: none"> • высокая стоимость

Базовые версии перечисленных программ не имеют:

- многопользовательского режима в работе (в базовой версии работать с программой может только один пользователь и только на одном ПК);
- не поддерживают возможности конфигурирования (невозможность изменить функционал под нужды конкретного предприятия).

Таким образом, коробочные решения зачастую не эффективны для малых предприятий. Причинами этому служит высокая стоимость приобретения и сопровождения данных программ, затраты по времени запуска (в связи с наличием функционала, который нам не нужен). Для стабильной работы таких программ также требуется хороший компьютер или ноутбук с оперативной памятью не менее 8Гб и процессор с тактовой частотой от 3GHz.

Отсюда для малых предприятий вытекает необходимость разработки собственной товароучетной системы.

Собственная товароучетная система

Рассмотрим пример разработки собственной товароучетной системы для малого предприятия. Он включает в себя минимальный набор основных функций, которые необходимы для учета продаж.

Для этого достаточно разработать программу, в которой необходимо реализовать следующие задачи: поступление, реализация, возврат, списание, переоценка, подключение ККТ.

Товароучетная система должна обеспечивать следующие основные функции:

- общие сведения об организации;
- настройка и подключение оборудования;
- учет прихода товаров на склад;
- установка цен на товары;
- продажа товаров с учетом остатка на складе через ККТ;
- отчеты (мониторинг продаж).

Предполагается, что конфигурация будет состоять из 5 подсистем, которые в свою очередь состоят из различных объектов конфигурации.

Разрабатываемая система включает в себя следующие основные объекты:

Подсистема «Администрирование»:

- справочник «Сведения об организации»;
- справочник «Физические лица»;
- справочник «Магазины»;
- справочник «Склады»;
- справочник «Кассы»;
- справочник «Кассы ККТ».

Подсистема «Нормативно-справочная информация»:

- справочник «Единица измерения»;
- справочник «Номенклатура»;
- справочник «Контрагенты»;
- справочник «Сотрудники».

Подсистема «Закупки»:

- документ «Поступление товаров»;
- документ «Установка цен»;
- отчет «Анализ заказов»;
- отчет «Взаиморасчеты с поставщиками».

Подсистема «Продажи»:

- РМК (рабочее место кассира);
- документ «Реализация товаров»;
- документ «Открытие кассовой смены»;
- документ «Закрытие кассовой смены»;
- документ «Чеки коррекции»;
- документ «Настройка РМК»;
- отчет «Розничные продажи»;
- отчет «Остатки товаров на складе».

Подсистема «Подключаемое оборудование» (реализация подключения кассы).

На рисунке 2 представлена рабочая область конфигурации, включающая переход между подсистемами, а также пример документа «Поступление товаров».

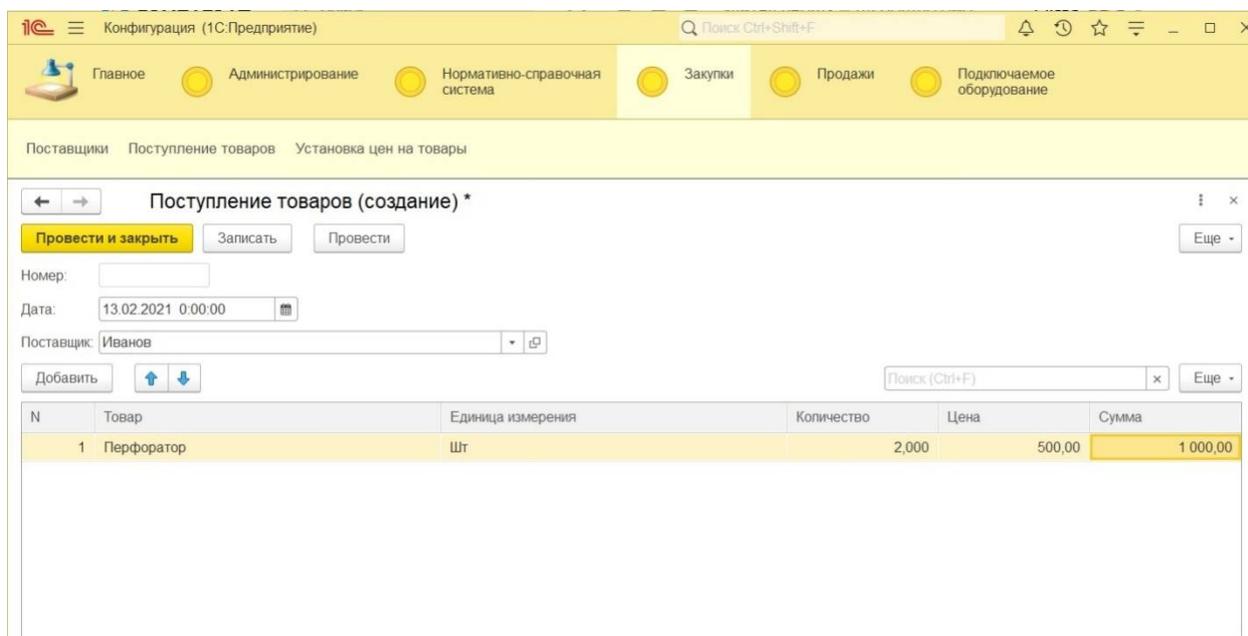


Рис. 2. Внешний вид окна пользователя

Основные виды онлайн-касс. Фискальный регистратор Атол 30Ф

Какая же ККТ подойдет для решения поставленных задач?

Онлайн-кассы бывают следующих видов:

- автономные кассы (полноценно функционируют без подключения к девайсам; могут работать как от аккумулятора, так и от сети; имеют небольшие размеры; беспроводная передача данных через Wi-Fi, Bluetooth);

- смарт-терминалы (кассы, представляющие планшет с сенсорным экраном и встроенным устройством печати чеков; имеют собственное ПО для решения кассовых задач);
- фискальные регистраторы (управляются с помощью кассовой программы, подключенной к ПК; имеют небольшие габариты);
- облачные решения (кассовые программы с подключенным фискальным накопителем; чеки формируются только в электронном виде).

Для мелкого и среднего бизнеса с широким ассортиментом товаров удобнее использовать фискальные регистраторы.

Одним из самых распространенных бюджетных решений является фискальный регистратор Атол 30Ф [1], который полностью адаптирован под 54-ФЗ, укомплектован фискальным накопителем с возможностью интеграции с торговыми программами, включая облачное ПО, способен обрабатывать до 100 чеков в день. Устройство поддерживает 6 видов налогов, возможность печати штрихкода и логотипа на чеке. Подключается к компьютеру через USB-кабель.

Данные в ОФД передаются по технологии EoU, которая устанавливается вместе с драйвером оборудования. Драйвер для Атол 30Ф (версия 10.8) доступен для скачивания на официальном сайте разработчика <http://fs.atol.ru/>. После установки драйвера в диспетчере устройств создаются два СОМ-порта: один виртуальный для подключения ККТ, второй – для передачи данных EoU.

Драйвер «АТОЛ:ККТ с передачей данных в ОФД (54-ФЗ) 10.x» разработан в двухуровневой архитектуре: интеграционная компонента и основная поставка. Основная поставка драйвера непосредственно взаимодействует с оборудованием. Интеграционная компонента – перенаправляет вызовы от 1С к основной поставке драйвера.

Для интеграции Атол 30Ф с 1С необходимо реализовать механизм добавления нового драйвера из файла с возможностью его установки, а также функционал, описанный выше в товароучетной системе.

Работа выполнена под научным руководством к.т.н., доцента кафедры ЭВМ Н.Н. Гринченко.

Библиографический список

1. Атол: оборудование и ПО для автоматизации торговли. Электронный ресурс. <https://www.atol.ru>
2. Описание и возможности 1С: Предприятие 8. Электронный ресурс. <https://saransk.1cbit.ru/blog/opisanie-i-vozmozhnosti-1s-predpriyatie-8/>
3. Федеральный закон от 22.05.2003 N 54-ФЗ (ред. от 23.11.2020) «О применении контрольно-кассовой техники при осуществлении расчетов в Российской Федерации» // Собрание законодательства РФ. – 26.05.2003.
4. Федеральный закон от 3 июля 2016 г. N 290-ФЗ (ред. От 03.07.2018г) «О внесении изменений в Федеральный закон "О применении контрольно-кассовой техники при осуществлении наличных денежных расчетов и (или) расчетов с использованием платежных карт" и отдельные законодательные акты Российской Федерации» (с изменениями и дополнениями) // Собрание законодательства РФ. – 04.07.2016.

УДК 004.932; ГРНТИ 28.23.15

АЛГОРИТМЫ ОБРАБОТКИ ПРОСТРАНСТВЕННО-КООРДИНАТНЫХ ДАННЫХ

А.И. Ефимов

*Рязанский государственный радиотехнический университет имени В.Ф. Уткина,
Российская Федерация, Рязань, lexie62rus@mail.ru*

Аннотация. В работе рассматриваются алгоритмы обработки пространственно-координатных данных, в частности, трехмерных облаков точек, получаемых в процессе работы стереосистемы.

Ключевые слова: стереозрение, алгоритм ICP, облака точек, дескрипторы ключевых точек.

SPATIAL-COORDINATE DATA PROCESSING ALGORITHMS

A.I. Efimov

*Ryazan State Radio Engineering University named after V.F. Utkin,
Russia, Ryazan, lexie62rus@mail.ru*

The abstract. The paper considers algorithms for spatial coordinate data processing, in particular, three-dimensional point clouds obtained from a stereo system.

Keywords: stereovision, ICP algorithm, point cloud, keypoint descriptors.

На современном этапе развития средств вычислительной техники и компьютерного зрения, существует целый класс задач, в основе решения которых лежат методы обработки и сшивки трехмерных поверхностей и облаков точек. К ним можно отнести задачи геометрической реконструкции объектов, картографирования и локализации объектов на полученных картах, построения трехмерных моделей и многие другие. Основой для получения трехмерных поверхностей или облаков точек как правило являются алгоритмы стереозрения в различных их модификациях, реже – заранее созданные и смоделированные объекты.

Так как решение задачи сшивки трехмерных поверхностей и облаков точек находят широкое практическое применение, существуют алгоритмы, достаточно качественно решающие эту проблему. Однако, большинство из них имеют существенные ограничения на применение, и не существует абсолютно универсального алгоритмического решения данной задачи в реальных условиях. По этой причине создание такого рода подходов представляется весьма и весьма актуальным.

Итеративный алгоритм ближайших точек (ICP)

Классическим вариантом решения задачи сшивки трехмерных облаков является использование алгоритма ICP (Итеративный алгоритм ближайших точек), предложенного в работе [1]. Алгоритм состоит из трех основных шагов:

1. Связка точек по критерию ближайшего соседа;
2. Оценка параметров преобразования с помощью функции среднеквадратичной стоимости (вычисление преобразования (смещение + поворот), минимизирующего среднеквадратичное расстояние (MSE) между парными точками);
3. Применение рассчитанного преобразования по множеству и обновление среднеквадратичной ошибки.

Эти три шага повторяются. Доказано, что они обеспечивают равномерную сходимость MSE.

Данный алгоритм вызвал интерес и с тех пор появилось множество его вариаций:

1. Поиск соседа – является модификацией алгоритма ICP. Соседние точки должны быть известны в двух соединяемых частях модели;
2. Схема многих решений – время алгоритма уменьшается за счет того, что уменьшается количество точек, задействованных в алгоритме.

3. Развернуто-геометрический итеративный алгоритм ближайших точек отличается от исходного алгоритма тем, что преобразования идут с учетом геометрических свойств твердого тела. Вместо матрицы поворота и вектора сдвига, используются ось вращения h и критическая точка c .

4. Масштабируемый итеративный алгоритм ближайших точек.

Алгоритм ICP зарекомендовал себя как надежное средство сшивания трехмерных поверхностей, однако ему присущи и существенные недостатки:

1) необходимо задавать такие начальные приближения R и T для первой итерации, чтобы предотвратить попадание алгоритма в локальные экстремумы, иначе итерация не сможет сойтись к правильному результату;

2) совмещаемые облака точек должны иметь общую область перекрытия;

3) на каждой итерации необходимо выполнять трудоемкую операцию поиска ближайшей точки.

Алгоритм совмещения трехмерных облаков на основе дескрипторов ключевых точек (ICP)

В работе [2, 3] описан алгоритм совмещения трехмерных облаков точек, основанный на использовании сигнатур спин-изображений.

Кратко суть предложенного авторами подхода состоит в следующем. В каждой точке M_{i_0} облаков точек методом наименьших квадратов находятся оценки \hat{a}, \hat{b} коэффициентов a, b уравнения плоскости $z = ax + by$, проходящей через данную точку. Оптимальные оценки \hat{a}, \hat{b} находятся в результате минимизации уклонения искомой плоскости от некоторого множества точек $\{M_i\}_{i=1}^k$ из окрестности заданного радиуса r точки M_{i_0} . Указанные оценки \hat{a}, \hat{b} необходимы для формирования в каждой точке вектора нормали $\mathbf{n} = (\hat{a}, \hat{b}, -1)$. Все векторы нормалей затем нормируются.

Каждой точке M_{i_0} из одного и второго облаков точек ставится в соответствие k -мерный вектор $\{\gamma_j, \alpha_j, \beta_j\}_{j=1}^k$, элементами которого являются тройки чисел $(\gamma_j, \alpha_j, \beta_j)$ - дескрипторы точки M_{i_0} . Здесь: γ_j - угол между вектором нормали $\mathbf{n}_{i_0}^0$ текущей точки M_{i_0} и вектором \mathbf{n}_j^0 $j \in \{1, 2, \dots, k\}$ точки M_j из окрестности точки M_{i_0} ; β_j - алгебраическая проекция на вектор $\mathbf{n}_{i_0}^0$ вектора $\overline{M_{i_0}M_j}$, то есть $\beta_j = \text{Pr}_{\mathbf{n}_{i_0}^0} \overline{M_{i_0}M_j} = (\overline{M_{i_0}M_j}, \mathbf{n}_{i_0}^0)$; $\alpha_j = \sqrt{|\overline{M_{i_0}M_j}|^2 - \beta_j^2}$.

Сравнение дескрипторов точек из одного и второго облаков точек производится по следующему алгоритму. Пусть $\left\{ \left(\gamma_j^{(1)}, \alpha_j^{(1)}, \beta_j^{(1)} \right) \right\}_{j=1}^r$ - множество дескрипторов фиксированной точки $M_{i_0}^{(1)}$ из первого облака, а $\left\{ \left(\gamma_j^{(2)}, \alpha_j^{(2)}, \beta_j^{(2)} \right) \right\}_{j=1}^r$ - множество дескрипторов некоторой произвольной точки $M_i^{(2)}$ из второго облака, проверяемой на соответствие точке $M_{i_0}^{(1)}$. Для каждой точки $M_i^{(2)}$ из второго облака сначала проверяется условие близости по каждому параметру, а именно в цикле по j от 1 до r проверяется выполнение условий: $\left| \gamma_j^{(1)} - \gamma_j^{(2)} \right| < \varepsilon_1$, $\left| \alpha_j^{(1)} - \alpha_j^{(2)} \right| < \varepsilon_2$, $\left| \beta_j^{(1)} - \beta_j^{(2)} \right| < \varepsilon_3$. Здесь $\varepsilon_1, \varepsilon_2, \varepsilon_3$ - заранее заданные числа. Если хотя бы для од-

ного значения индекса j хотя бы одно из трех условий не выполнено, то осуществляется переход к новой точке второго облака. В противном случае находятся суммы

$$\sum_{\gamma} (M_i^{(2)}) = \sum_{j=1}^r |\gamma_{kj}^{(1)} - \gamma_{ij}^{(2)}|, \quad \sum_{\alpha} (M_i^{(2)}) = \sum_{j=1}^r |\alpha_{kj}^{(1)} - \alpha_{ij}^{(2)}|, \quad \sum_{\beta} (M_i^{(2)}) = \sum_{j=1}^r |\beta_{kj}^{(1)} - \beta_{ij}^{(2)}|, \quad i \in I. \quad (1)$$

Для каждой суммы в составе (1) находятся точки $M_s^{(2)}$, $s=1,2,3$, в каждой из которых достигается минимум соответствующей суммы из (1):

$$M_{i_1}^{(2)} = \arg \min_{M_i^{(2)}} \sum_{\gamma} (M_i^{(2)}), \quad M_{i_2}^{(2)} = \arg \min_{M_i^{(2)}} \sum_{\alpha} (M_i^{(2)}), \quad M_{i_3}^{(2)} = \arg \min_{M_i^{(2)}} \sum_{\beta} (M_i^{(2)}). \quad (2)$$

Если $i_1 = i_2 = i_3 = i^*$, то точка $M_{i^*}^{(2)}$ из второго облака идентифицируется как соответствующая точке $M_{i_0}^{(1)}$, иначе – несоответствующей.

Данный подход демонстрирует хорошие результаты, в тех случаях, когда в координатах точек совмещаемых облаков точек отсутствует шумовая составляющая. Число корректно установленных соответствий между отвечающими друг другу точек достигает 72-75% при полном отсутствии некорректных сопоставлений.

Исследование предложенного алгоритма совмещения на устойчивость к шумовой составляющей производилось следующим образом: к координате z в каждом облаке точек добавлялась нормально распределенная случайная составляющая с нулевым математическим ожиданием и с заданным уровнем шума. На рисунке 1 приведены результаты совмещения при $\sigma = 0,01$ и $\sigma = 0,1$.

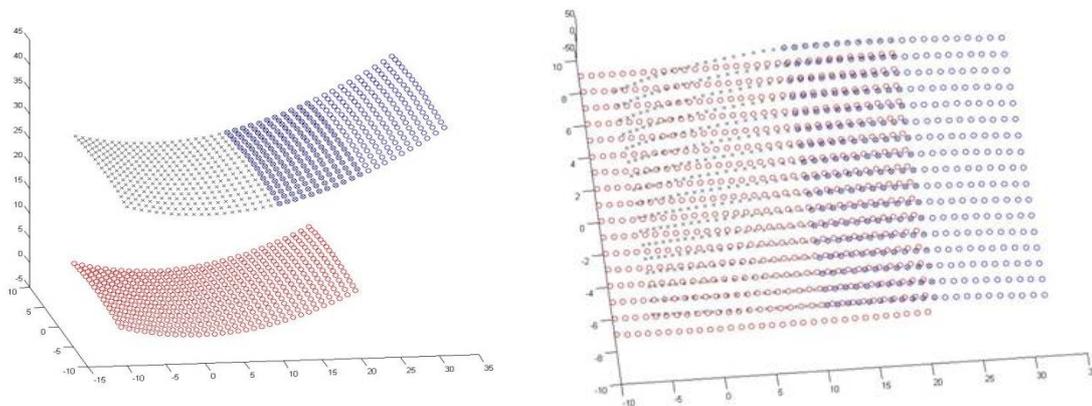


Рис. 1. Совмещение модельных облаков при $\sigma = 0,01$ и $\sigma = 0,1$ соответственно

При высоких значениях σ наблюдается явление «сгибания» крайних точек, соответствующих облаков точек (рисунок 2). Причина такого явления может заключаться в высокой чувствительности используемого метода совмещения на основе матрицы гомографии к шумовой составляющей, влекущей серьезные искажения в коэффициентах матрицы, и, как следствие, итоговых результатов.

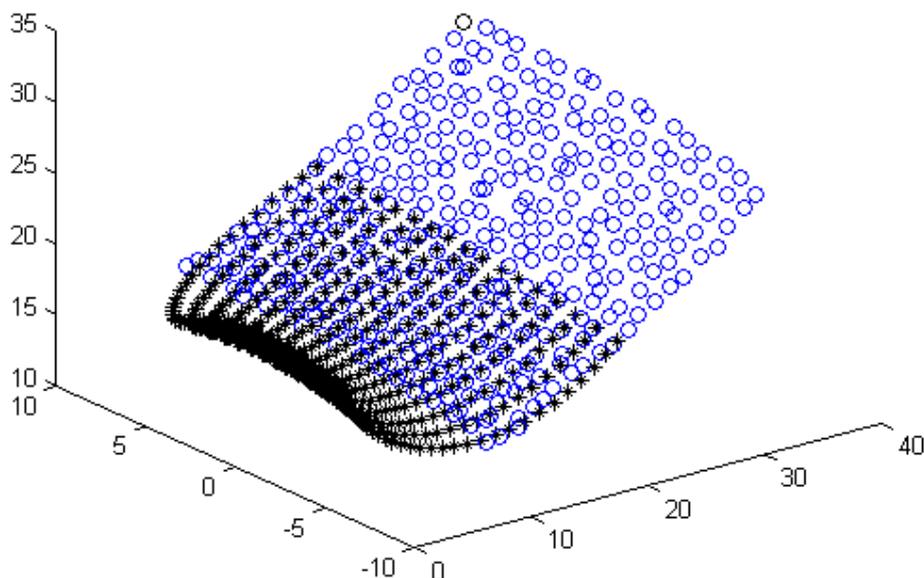


Рис. 2. Совмещение модельных облаков при $\sigma = 0,5$

Проведенное исследование позволяет сделать следующий вывод. Совмещение 3D поверхностей, основанное на преобразовании одного облака точек к другому с помощью матрицы гомографии, дает хорошие результаты при точно известных координатах точек в облаках точек и требует низких вычислительных затрат. Однако даже не очень высокий уровень ошибок в координатах точек в совмещаемых облаках может приводить к существенным искажениям результатов совмещения. В этом случае представляется целесообразным запуск итерационной процедуры последовательного нахождения последовательности описанных преобразований, критерием сходимости которых выступает среднее расстояние между точками в совмещаемых облаках $V = \{M_i(x_i, y_i, z_i)\}_{i=1}^n$ и $V' = \{M'_i(x'_i, y'_i, z'_i)\}_{i=1}^m$.

Библиографический список

1. Besl, Paul J.; N.D. McKay. A Method for Registration of 3-D Shapes" IEEE Transactions on Pattern Analysis and Machine Intelligence. 14 (2): 239–256. doi:10.1109/34.12179
2. Ефимов, А.И. Алгоритм совмещения трехмерных облаков точек / А.И. Ефимов // Биотехнические, медицинские и экологические системы, измерительные устройства и робототехнические комплексы - Биомед-системы-2019, 2019, С. 632-635.
3. Новиков, А.И. Алгоритмы автоматической идентификации объектов на разнородных изображениях и совмещения изображений / А.И. Новиков, А.И. Ефимов, М.Б. Никифоров, Д.А. Колчаев // Информационные технологии и нанотехнологии. Сборник трудов ИТНТ-2018. Самарский национальный исследовательский университет имени академика С.П. Королева. Самара, 2018, С. 650-663.
4. Sablina V.A., Efimov A.I., Novikov A.I. Combined approach to object contour superimposition in heterogeneous images. Proceedings of the 6th Mediterranean Conference on Embedded Computing. – Bar, Montenegro, 2017. P. 120-123.

УДК 004.65; ГРНТИ 50.41.21

РЕАЛИЗАЦИЯ СЛОЖНЫХ СВЯЗЕЙ С ПОМОЩЬЮ СОВРЕМЕННЫХ СУБД

А.А. Вьюгина, С.Н. Баранова

Рязанский государственный радиотехнический университет имени В.Ф. Уткина,
Российская Федерация, Рязань, baranova.sv.n@gmail.com

Аннотация. При проектировании баз данных возникает необходимость реализации сложных связей. Не все современные СУБД позволяют удобно работать с такими видами связей. Данная статья посвящена разбору возможностей, которые предоставляют современные СУБД для реализации сложных связей.

Ключевые слова: базы данных, рекурсивные связи, наследование, Oracle, MySQL.

IMPLEMENTATION OF COMPLEX RELATIONSHIPS USING MODERN DBMS

A.A. Vyugina, S.N. Baranova

Ryazan State Radio Engineering University named after V.F. Utkin,
Russia, Ryazan, baranova.sv.n@gmail.com

The summary. When designing databases, there is a need to implement complex relationships. Not all modern DBMSs allow you to conveniently work with such types of links. This article is devoted to the analysis of the possibilities that modern DBMS provide for the implementation of complex relationships.

Keywords: databases, recursive relationships, inheritance, Oracle, MySQL.

Сложные связи являются неотъемлемой частью при проектировании баз данных. Часто в различных областях имеются отношения между сущностями, которые не могут быть реализованы простыми связями, или их использование ведет к усложнению структуры или денормализации базы данных. Поэтому возникает необходимость в СУБД, которые поддерживали бы функционал для работы с разными типами сложных связей.

В данной работе будут рассмотрены 2 примера сложных связей: рекурсия и наследование.

Рекурсивная связь представляется как связь сущности самой с собой. Такой вид связи рекомендуется использовать при необходимости реализовывать древовидную структуру.

Хорошим примером такой связи является иерархия подчиненности сотрудников в какой-либо организации. На рисунке 1 представлена ER-диаграмма реализации описываемой сложной связи.

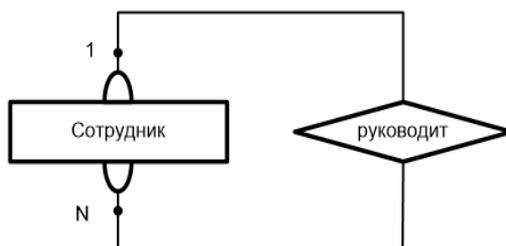


Рис. 1. Диаграмма, показывающая рекурсивную связь

В таком случае в виде таблицы такая связь должна иметь следующие особенности: первичный ключ сущности выступает одновременно в виде внешнего ключа дочерней сущности [1]. В таком случае выявляется одна проблема: необходима особая внимательность к поддержанию целостности базы данных, в том числе избегать возможность бесконечной рекурсии.

Таким образом, рекурсия в базе данных показывается расширением атрибута с помощью внешнего ключа, представляющего собой необязательную ссылку на первичный ключ, который является по сути родительским элементом.

На 2020 год самыми популярными среди СУБД являются Oracle и MySQL [2], поэтому для рассмотрения практической реализации использования рекурсивной связи будут использованы именно они.

СУБД Oracle имеет специальную конструкцию CONNECTBY для реализации рекурсивной связи [3]. Ее использование позволяет определить взаимосвязь между родительскими атрибутами и дочерними. Самый простой вариант – указать столбцы, через которые устанавливается связь. Для выборки иерархии необходимо иметь представление о первом элементе ее элементе. Чтобы указать вершину дерева, можно использовать оператор STARTWITH.

Для ограничения выборки можно использовать стандартное предложение WHERE.

Рекурсивный запрос выполняется следующим образом: выделяются все отношения, которые удовлетворяют фразе STARTWITH, затем на основании найденных строк производится поиск дочерних строк, после операция повторяется для новых строк и так далее, пока не будет построена иерархия связей.

Конструкция выглядит следующим образом:

```
SELECT {Столбцы}
FROM {Таблица}
STARTWITH {Условие начального элемента иерархии}
CONNECTBY {Условие для построения дерева}
```

Для понимания работы рекурсивной связи в СУБД MySQL необходимо дать понятие выражения общей таблицы (Common Table Expression, CTE) – временное представление, которое можно использовать несколько раз в запросе, но который не сохраняет данные. Выражения общей таблицы могут быть рекурсивными. В таком случае набор результирующих отношений строится итеративно. Для использования CTE существует специальный оператор WITHRECURSIVE [4]. Данный оператор содержит ряд требований:

- все выборки должны быть связаны с UNIONALL или UNIONDISTINCT 1;
- нельзя использовать никаких групповых агрегатных функций;
- ссылаться на рекурсивное выражение общей таблицы можно ссылаться только

один раз.

В целом конструкция выглядит следующим образом:

```
WITH РекурсивныйЗапросAS
(
  {Запрос на начальный элемент иерархии}
UNIONALL
  {Запрос к целевой таблице и CTE}
)
SELECT * FROM РекурсивныйЗапрос
```

Следующим сложным видом связи можно представить наследование. Данный вид связи реализуется в виде взаимосвязи одной родительской сущности и несколькими дочерними. В таком случае необходимо, чтобы атрибут первичного ключа родителя присутствовал в составе первичного ключа дочерних сущностей. Можно сказать, что при реализации наследования родительский класс является классом обобщенных сущностей, который включает все атрибуты, определенные для всех дочерних таблиц.

Пример построения таких связей представлен на рисунке 2. Он представляет собой общую родительскую сущность «Преподаватель» и 3 его дочерних сущности.

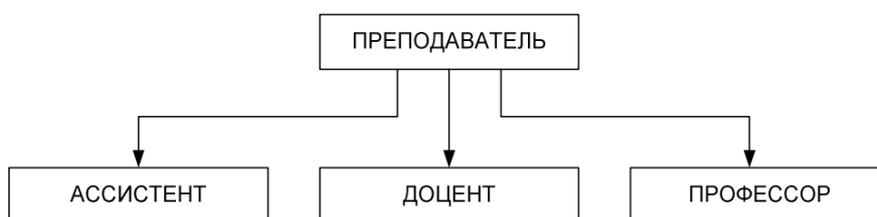


Рис. 2. Диаграмма связи наследования

Возможность реализации данного типа связи также будет рассмотрена в СУБД Oracle и MySQL.

Oracle не позволяет реализовать наследование сущностей [3], но однако имеет возможность наследования между пользовательскими типами. Благодаря данной технологии возможно построить некоторую взаимосвязь между сущностями.

Описание обычного типа возможна следующим образом:

```
CREATETYPEНазваниеРодительскогоТипаASOBJECT  
(Поля родительского типа) NOTFINAL;
```

Затем создается подтип, который будет наследовать все атрибуты и методы от данного супертипа:

```
CREATETYPEНазваниеДочернегоТипа  
UNDERНазваниеРодительскогоТипа(Поля дочернего типа);
```

В СУБД MySQL не поддерживается наследование таблиц ни в каком виде [4]. Однако существует возможность имитировать данный функционал с помощью внешних ключей.

Примером СУБД, в которой явно реализован механизм наследования таблиц, является PostgreSQL [5]. Конструкция выглядит следующим образом:

```
CREATETABLEРодительскаяТаблица (Поля таблицы);  
  
CREATETABLEДочерняяТаблица (Поля таблицы)  
INHERITS (РодительскаяТаблица);
```

При таком построении любое обращение к дочерней таблице будет возвращать результат с присоединенными полями родительской. Однако любые ограничения и ключи должны отдельно быть указаны для таблиц, как для самостоятельных сущностей, так как они не наследуются.

Сложные связи тяжело реализуются на реляционной модели данных и мало поддерживаются современными СУБД, хотя присутствуют введения некоторой функциональности. Часто это связано с неэффективностью работы: запросы к базе данных долго выполняются за счёт необходимости выполнения объединения нескольких таблиц. Однако, несмотря на существенный недостаток, использование сложных связей позволяет упростить саму структуру базы данных для пользователя системы, что показывает необходимость проведения дальнейшей разработки в этом направлении.

Работа выполнена под научным руководством к.т.н., доцента кафедры ЭВМ Н.Н. Гринченко.

Библиографический список

1. Грошев А.С. Основы работы с базами данных : учебное пособие / Грошев А.С.. — Москва : Интернет-Университет Информационных Технологий (ИНТУИТ), Ай Пи Ар Медиа, 2021. — 255 с. — ISBN 978-5-4497-0914-1. — Текст : электронный // Электронно-библиотечная система IPR BOOKS : [сайт]. — URL: <http://www.iprbookshop.ru/102038.html>
2. Базы данных. Тенденции общемировые и в России– 2020. – Текст: электронный. – URL: <https://habr.com/ru/post/533880/>
3. Lance Ashdown, Donna Keesling, Tom Kyte«Oracle Database Database Concepts», 21с, Copyright © 1993, 2020, Oracle and/or its affiliates.
4. MySQL 8.0 Reference Manual – 2021. – Текст : электронный. – URL: <https://dev.mysql.com/doc/refman/8.0/en/preface.html>
5. Документация к PostgreSQL 13.2 – The PostgreSQL Global Development Group, 2021. – Текст: электронный. – URL: <https://postgrespro.ru/media/docs/postgresql/13/ru/postgres-A4.pdf>

УДК 004.021; ГРНТИ 50.05.03

АНАЛИЗ ПАРАЛЛЕЛЬНЫХ АЛГОРИТМОВ

Н.И. Ципорков

*Рязанский государственный радиотехнический университет имени В.Ф. Уткина,
Российская Федерация, Рязань, tsepster@gmail.com*

Аннотация. В работе рассматриваются процесс анализа параллельных алгоритмов, показатели, по которым можно определить качество распараллеливания алгоритма для вычислительной задачи.

Ключевые слова: параллельный алгоритм, нижняя граница, верхняя граница, ускорение, эффективность, стоимость.

ANALYSIS OF PARALLEL ALGORITHMS.

N.I. Tsiporkov

*Ryazan State Radio Engineering University named after V.F. Utkin,
Russia, Ryazan, tsepster@gmail.com*

The summary. The paper discusses the process of analyzing parallel algorithms, indicators by which it is possible to determine the quality of parallelization of an algorithm for a computational problem.

Keywords: parallel algorithm, lower bound, upper bound, acceleration, efficiency, cost

Анализ параллельных алгоритмов — это процесс определения, насколько хорош алгоритм, насколько он быстро работает, насколько ресурсозатратна его работа и также насколько эффективно он использует доступные ресурсы.

После того, как алгоритм для решения какой-то проблемы был разработан, он обычно оценивается с использованием следующих критериев: время работы, количество используемых процессоров и стоимость. Иногда используются критерии, которые специфичны для конкретной технологии, когда известно, что данный алгоритм предназначен для работы на компьютере, основанном на данной технологии.

Так как ускорение вычислений — это основная причина проектирования параллельных алгоритмов, то самым важным критерием будет являться время работы. Оно определяется с момента начала работы алгоритма до его завершения. Так как не все процессоры начинают и заканчивают работу одновременно, то время работы равно времени с момента начала вычислений первым процессором до момента, когда последний процессор завершит работу.

Перед реализацией алгоритма (последовательного или параллельного) принято проводить теоретический анализ времени, необходимого для решения поставленной вычислительной задачи. Обычно это делается путем подсчета количества основных операций или шагов, выполняемых алгоритмом в худшем случае. Время работы параллельного алгоритма обычно определяется путем подсчета двух видов шагов: вычислительных шагов и шагов маршрутизации. Вычислительный этап — это арифметическая или логическая операция, выполняемая над элементом данных в процессоре. С другой стороны, на этапе маршрутизации данные перемещаются от одного процессора к другому через общую память или через сеть коммуникации. Для задачи размера n время параллельной работы алгоритма в худшем случае, функция от n , будет обозначаться через $t(n)$. Время работы также зависит от количества процессоров [1].

В общем, вычислительные шаги и шаги маршрутизации не обязательно требуют одинакового количества единиц времени. Шаг маршрутизации обычно зависит от расстояния между процессорами и обычно занимает немного больше времени, чем вычислительный шаг.

Учитывая вычислительную проблему, для которой только что был разработан новый последовательный алгоритм, разработчики алгоритмов часто задают следующие два вопроса:

1. Является ли это максимально быстрым алгоритмом решения проблемы?

2. Если нет, то как он сравнивается с другими существующими алгоритмами для этой проблемы?

Ответ на первый вопрос обычно получается путем сравнения количества шагов, выполняемых алгоритмом, с известной нижней границей количества шагов, необходимых для решения проблемы в худшем случае.

К примеру, мы хотим вычислить произведение двух матриц размера $n \times n$. Поскольку результирующая матрица имеет n^2 элементов, то, как минимум, такое количество шагов требуется любому алгоритму умножения матриц просто для получения результата.

Такие нижние границы, как в примере выше, обычно известны как очевидные или тривиальные нижние границы, поскольку они получаются путем подсчета количества шагов, необходимых во время ввода и / или вывода.

Если количество шагов, выполняемых алгоритмом в худшем случае, равно (или того же порядка) нижней границе, то алгоритм является самым быстрым из возможных и считается оптимальным. В противном случае, возможно, придется изобрести более быстрый алгоритм, или можно будет улучшить нижнюю границу. В любом случае, если новый алгоритм быстрее, чем все известные алгоритмы для проблемы, то мы говорим, что он установил новую верхнюю границу количества шагов, необходимых для решения этой проблемы в худшем случае. Поэтому вопрос 2 всегда решается путем сравнения времени работы нового алгоритма с существующей верхней границей проблемы (установленной самым быстрым из ранее известных алгоритмов) [1].

При оценке параллельного алгоритма для имеющейся проблемы вполне естественно делать это с точки зрения наилучшего доступного последовательного алгоритма для этой проблемы. Хорошим показателем качества параллельного алгоритма является его ускорение.

Ускорение, которое получается при использовании параллельного алгоритма для p процессоров, в сравнении с последовательным алгоритмом определяется величиной [2]:

$$S_p(n) = T_1(n) / T_p(n),$$

где $T_1(n)$ – время выполнения последовательного алгоритма;
 $T_p(n)$ – время выполнения параллельного алгоритма для p процессоров;
 n – входные данные.

Вторым по важности критерием при оценке параллельных алгоритмов является количество процессоров, которое требуется для решения проблемы. При наличии нескольких процессоров проблема обслуживания, в частности, усугубляется, и цена, уплачиваемая за гарантию высокой степени надежности, резко возрастает. Следовательно, чем большее количество процессоров использует алгоритм для решения проблемы, тем дороже становится получение решения. Для задачи размера n количество процессоров, требуемых алгоритмом, как функция n , будет обозначаться $p(n)$. Иногда количество процессоров не зависит от n .

Стоимость параллельного алгоритма определяется как:

Стоимость = время параллельной работы \times количество используемых процессоров

Другими словами, стоимость равняется количеству шагов, выполняемых совместно всеми процессорами при решении проблемы в худшем случае. Это определение предполагает, что все процессоры выполняют одинаковое количество шагов. Если это не так, тогда стоимость является верхней границей общего количества выполненных шагов. Для задачи размера n стоимость параллельного алгоритма, функция от n , будет обозначаться как $C(n)$. Таким образом, $C(n) = p(n) * t(n)$. Параллельный алгоритм не является оптимальным по стоимости, если существует последовательный алгоритм, время работы которого меньше стоимости параллельного алгоритма.

Предположим, что известна нижняя граница количества последовательных операций, необходимых в худшем случае для решения проблемы. Если стоимость параллельного алго-

ритма для этой задачи совпадает с этой нижней границей с точностью до постоянного мультипликативного множителя, то алгоритм считается оптимальным по стоимости. Это связано с тем, что любой параллельный алгоритм может быть смоделирован на последовательном компьютере. Если общее количество шагов, выполненных во время моделирования, равно нижней границе, это означает, что, когда дело доходит до стоимости, этот параллельный алгоритм не может быть улучшен, поскольку он выполняет минимально возможное количество шагов. Время работы оптимального по стоимости параллельного алгоритма можно сократить за счет использования большего числа процессоров. Точно так же мы можем использовать меньше процессоров, сохраняя при этом оптимальность затрат, если мы будем согласны на большее время работы [1].

Параллельный алгоритм не будет оптимальным по стоимости, если существует последовательный алгоритм, время работы которого меньше стоимости параллельного алгоритма.

Когда не известен оптимальный последовательный алгоритм для решения проблемы, эффективность параллельного алгоритма для этой проблемы используется для оценки его стоимости.

Эффективность использования параллельным алгоритмом процессоров при решении поставленной задачи определяется следующим соотношением [2]:

$$E_p(n) = T_1(n) / (pT_p(n)) = S_p(n) / p.$$

Эффективность параллельного алгоритма показывает среднюю долю времени, в течение которой процессоры будут реально использоваться при решении задачи.

При оценке качества параллельного алгоритма предполагается, что известны наилучшие значения для показателей ускорения и эффективности. Однако, не для каждой вычислительной задачи возможно получить идеальные значения ускорения ($S_p=p$) и эффективности ($E_p=1$) [2].

Стоит также отметить, что время, которое требуется параллельному алгоритму для получения входных данных и, после завершения вычислений, для возврата выходных данных, следует рассматривать как постоянное.

Работа выполнена под научным руководством к.т.н., доцента кафедры ЭВМ Н.Н. Гринченко.

Библиографический список

1. Selim G. Akl. The Design and Analysis of Parallel Algorithms, 1989 – pp. 412.
2. Показатели эффективности параллельных алгоритмов [Электронный ресурс] URL: <https://intuit.ru/studies/courses/4447/983/lecture/14931?page=12>.

УДК 621.396; ГРНТИ 49.03.05

ФИЗИЧЕСКИ НЕКЛОНИРУЕМЫЕ ФУНКЦИИ МИКРОПРОЦЕССОРОВ

Е.Р. Комлева, М.Б. Никифоров

*Рязанский государственный радиотехнический университет им. В.Ф. Уткина,
Российская Федерация, komleva62rus2014@yandex.ru*

Аннотация. В данной работе рассматривается концепция физически неклонировуемых функций (PUF—PhysicalUnclonableFunction) применительно к СБИС.

Ключевые слова: аутентификация ИС, секретные ключи, конфиденциальная информация.

PHYSICALLY UNCLONABLE FUNCTIONS OF MICROPROCESSORS

E.R. Komleva, M.B. Nikiforov

*Ryazan State Radio Engineering University named after V. F. Utkin,
Russian Federation, komleva62rus2014@yandex.ru*

Abstract. In this paper, we consider the concept of physically unclonable functions (PUF - Physical Unclonable Function) in relation to VLSI.

Keywords: IC authentication, secret keys, confidential information.

Введение

Физически неклонировуемые функции (physically unclonable functions— PUF) [1,2,3] являются инновационными примитивами для формирования секретных признаков из сложных физических характеристик интегральной схемы (ИС), а не для хранения секретов в цифровой памяти. Например, летучий секрет может быть сгенерирован из случайных характеристик задержки проводов и транзисторов. Поскольку PUF формируются в результате случайных изменений процесса изготовления ИС, секрет чрезвычайно трудно предсказать или повторить.

PUF значительно повышают физическую безопасность за счет генерации летучих секретов, которые существуют только когда чип включен и работает. Это требует от взломщика организовать атаку во время работы ИС и использования секрета, что значительно сложнее, чем обнаружение энергонезависимых ключей; атака по сторонним каналам должна точно измерять задержки PUF без изменения задержек или обнаруживать энергонезависимые ключи в регистрах без отключения питания или систем обнаружения несанкционированного доступа, которые очищают регистры. В статье рассмотрены секретные ключи для криптографических систем. Схемы PUF также могут быть использованы в качестве аппаратных генераторов случайных чисел [4]. Однако в данной статье основное внимание уделяется только аутентификации устройств и генерации ключей.

Физически функциональные функции

Физически случайная функция или физически неклонировуемая функция – это функция, которая сопоставляет набор проблем с набором ответов, основанных на неразрешимо сложной физической системе. Следовательно, это статическое отображение является “случайным”.

Функция может быть оценена только с помощью физической системы и уникальна для каждого физического экземпляра. PUF может быть реализован в различных физических системах, в данной работе основное внимание уделяется PUF кремниевых СБИС, которые основаны на скрытой информации синхронизации и задержки интегральных схем [2, 3]. Даже при одинаковых масках компоновки, различия в производственном процессе вызывают значительные различия в задержках между различными элементами ИС.

Как упоминалось во введении, PUF обеспечивают значительно более высокую физическую безопасность, извлекая секреты из сложных физических систем, а не храня их в энергонезависимой памяти. Еще одним преимуществом PUF является то, что они делают это без

каких-либо специальных производственных процессов или этапов программирования и тестирования.

Арбитр PUF

На рисунке 1 показана кремниевая схема задержки PUF, основанная на мультиплексорах и арбитре (D-триггер). Схема имеет многоразрядный вход X и вычисляет 1-битный выходной сигнал Y на основе относительной разницы задержек между двумя путями с одинаковой длиной компоновки. Входные биты определяют пути задержки, управляя мультиплексорами. Здесь пара мультиплексоров, управляемых одним и тем же входным битом $X[i]$, работает как коммутационная схема (пунктирные поля на рисунке). В мультиплексорах сигналы проходят через две задержки слева, если входной управляющий бит $X[i]$ равен нулю. В противном случае верхний и нижний сигналы переключаются.

Таким образом, схема может создать пару путей задержки для каждого входа X . Чтобы оценить выход для конкретного входа, восходящий сигнал подается на оба пути одновременно, сигналы проходят через два пути задержки, и арбитр (защелка) в конце цикла прохождения решает, какой сигнал быстрее. Выход равен единице, если сигнал на вход данных защелки (D) быстрее, и нулю в противном случае.

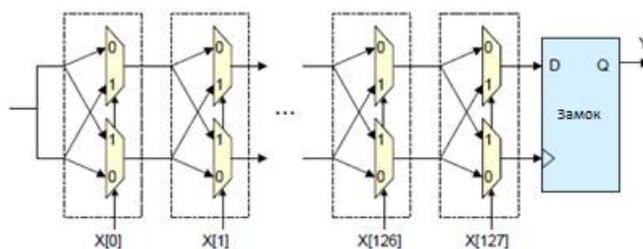


Рис. 1. Схема задержки арбитра PUF

Поскольку схема PUF довольно проста, злоумышленники могут попытаться построить точную временную модель и узнать параметры из многих пар входов-выходов [5]. Чтобы предотвратить эти атаки на этапе построения модели, выход схемы PUF может быть запутан путем XOR нескольких выходов или выход PUF может быть использован в качестве одного из сигналов управления MUX.

Обратите внимание, что атака на построение модели не имеет отношения к генерации криптографического ключа, где выходные данные PUF никогда не подвергаются прямому воздействию.

Существует два способа построения k -битного отклика на 1-битном выходе этой схемы задержки PUF. Одна схема может использоваться k раз с разными входами. Задача используется в качестве основы для генератора псевдослучайных чисел (например, сдвигового регистра линейной обратной связи). Затем схема задержки PUF вычисляется k раз, используя k различных битовых векторов от генератора псевдослучайных чисел, служащего входом X для настройки пути задержки. Кроме того, можно многократно дублировать саму схему PUF с одним выходом, чтобы получить k битов с одной оценкой.

Экспериментальные исследования разработчики [3] проводили со схемой PUF на основе арбитра с 64 ступенями. Схема была изготовлена и испытана в процессе TSMC 0,18 мкм, однополигональном 6-уровневом металлическом корпусе. Результаты испытаний показывали, что две идентичные схемы PUF на двух разных чипах имеют разные выходы для одного и того же входа с вероятностью 23% (межчиповая вариация). С другой стороны, несколько измерений на одном чипе отличаются только с вероятностью 0,7%.

Поскольку контролируется относительная разница задержек, PUF устойчив к изменениям окружающей среды. Для реалистичных изменений температуры от 20 до 70 градусов Цельсия и регулируемых изменений напряжения $\pm 2\%$ выходной шум составляет 4,8% и 3,7% соответственно. Даже при повышении температуры на 100С и изменении напряжения на 33% выходной шум PUF все равно остается ниже 9%. Эта вариация значительно меньше, чем межчиповая вариация в 23%, что позволяет идентифицировать отдельные чипы. Идеально симметричная схема на рисунке 1 увеличила бы межчиповую вариацию до 50%. Схема, изготовленная на тестовых чипах, имеет систематические перекосы в компоновке, потому что провода были автоматически проложены с помощью инструментов САПР и присутствует перекосяк в защелке, используемой в качестве арбитра. Удаление перекосов при тщательной компоновке увеличило бы вариацию между чипами.

Генерация криптографических ключей

Во многих приложениях безопасности криптографические примитивы, такие как шифрование, цифровые подписи и код аутентификации сообщений, играют центральную роль. К сожалению, выходы из описанных схем PUF не подходят в качестве криптографических ключей. Из-за шума выходы, вероятно, будут немного отличаться при каждой оценке даже на одной и той же ИС для одной и той же задачи. С другой стороны, криптографические примитивы требуют, чтобы каждый бит ключа оставался постоянным. Более того, некоторые примитивы, такие как RSA, требуют ключи для удовлетворения конкретных математических свойств, в то время как выходы PUF случайным образом определяются производственными вариациями.

В этом разделе мы рассмотрим, как PUF могут генерировать летучие секретные ключи, которые могут быть использованы для криптографических операций. Есть два компонента. Во-первых, процесс исправления ошибок, который состоит из инициализации и повторной генерации, гарантирует, что PUF может последовательно производить один и тот же выход, даже если есть значительные изменения окружающей среды, такие как колебания напряжения и температуры. Во-вторых, ключ-процесс генерации преобразует выходные данные PUF в криптографические ключи. Общий процесс показан на рис.2.

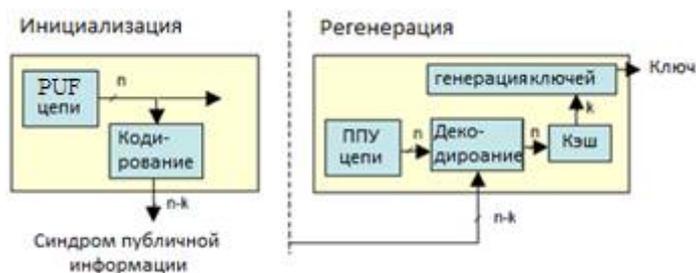


Рис. 2. Генерация криптографических ключей с помощью PUF

На этапе инициализации выходной сигнал генерируется из схемы PUF, а синдром исправления ошибок для этого выходного сигнала вычисляется и сохраняется для последующего использования. Например, код БЧХ (код Боуза-Чоудхури-Хоквенгема) может быть использован для вычисления синдрома. Синдром - это информация, которая позволяет корректировать битовые перевороты в повторно генерируемых выходах PUF. Если схема маскирования используется с RO PUF (ringoscillator-RO), битовый вектор, который выбирает пары осцилляторов, также будет сохранен вместе с синдромом. Обратите внимание, что синдром и этот битовый вектор являются общедоступной информацией и могут храниться в любом месте (на чипе, вне чипа или удаленно на сервере).

Чтобы повторно сгенерировать тот же выход PUF, PUF сначала производит выход из схемы. Если есть сохраненный битовый вектор, то он используется для выбора пар. Затем, PUF использует синдром из инициализационного шага для устранения каких-либо изменений в выходной цепи. Таким образом, PUF может последовательно воспроизводить выходные данные с шага инициализации.

Ясно, что синдром раскрывает информацию о выходе схемы задержки PUF. В целом, однако, учитывая b -битный синдром, злоумышленники могут узнать не более b -бит о PUF выходе схемы задержки. Поэтому, чтобы получить k секретных битов после исправления ошибки, мы генерируем $n = k + b$ битов из схемы задержки PUF. Даже с этим синдромом противнику все равно нужно угадать по крайней мере k битов, чтобы найти правильный ответ PUF. Например, мы можем использовать BCH (127,64,21) код для надежной генерации 64-битных секретов. Код BCH (n, k, d) может исправлять до $(d - 1)/2$ ошибок из n битов с синдромом $(n - k)$ -бит ($b = n - k$).

В то время как маска может раскрыть информацию о том, какое кольцо частоты осциллятора находится далеко друг от друга, она не раскрывает информацию о знаке сравнения, то есть о генерируемых битах. Если кольцевой генератор используется много раз для генерации битов, то вполне возможно, что информация о порядке кольцевых генераторов может быть извлечена из маски. Это можно легко исключить, используя каждый генератор только один раз для генерации одного бита.

Для криптографических операций, использующих в качестве ключа случайно выбранное число, выводится код исправления ошибок (ECC), он может быть просто хэширован до желаемой длины и используется в качестве криптографического ключа. Например, примитивы симметричных ключей, такие как AES, могут использовать хэшированный вывод PUF.

Для криптографических операций, ключи которых должны удовлетворять специальным свойствам (например, пара ключей RSA), хэширование вывода PUF используется в качестве начального значения для алгоритма генерации ключей.

Таким образом, PUF может генерировать ключи для любой криптографической операции. Отметим, что PUF просто генерируют ключи, которые можно использовать со стандартным алгоритмом. В криптографических алгоритмах не требуется никаких изменений. PUF с ее возможностью генерации ключей может быть плотно интегрирована с процессором для обеспечения физической безопасности процессора [6].

Технологии памяти для хранения секретов

Обычные решения для встраивания секретов в микросхемы и вычислительные устройства полагаются на оперативную память с батарейным питанием или на энергонезависимые запоминающие устройства, такие как ПЗУ, предохранители или flash / EEPROM. При использовании этих технологий встроенная память программируется ключом, который должен оставаться секретным. PUF обеспечивают уникальные секреты для каждой ИС с более сильной физической безопасностью. Кроме того, в отличие от ключей в памяти, PUF не требуют программирования секретов и их трудно дублировать даже производителем микросхем. Поскольку схема PUF использует только стандартную цифровую логику, нет необходимости в дополнительной маске или специальных этапах изготовления, и PUF можно легко масштабировать до передовых технологий.

Алгоритм создания секретных ключей микропроцессоров

Одним из возможных вариантов создания PUF-ключей микропроцессоров (МП) является использование внутренней памяти МП: РОН, КЭШ, портов ввода-вывода и др. В микросхемах общего применения подобная аутентификация не представляется целесообразной,

там достаточно приемки ОТК. В ответственной аппаратуре специального назначения, где индивидуальная отбраковка имеет значение, в первую очередь надо определить необходимую степень защиты. Укрупненный алгоритм создания секретных ключей микропроцессоров может быть следующим.

Шаг 1. Формирование требований к степени защиты. Это анализ условий применения, определение объёма поставок и общего тиража производства, оценка вероятностных характеристик взлома.

Шаг 2. Расчёт требуемой разрядности ключей.

Шаг 3. Выбор элементов МП, пригодных для генерации ключей. Ранжирование элементов по сложности чтения ключей. Наиболее просто реализуемые ключи на основе портов ввода-вывода. Если их разрядность недостаточна для удовлетворения требований шага 2, придется использовать более трудно-доступные регистры.

Шаг 4. Разработка аппаратно-программного комплекса для аутентификации МП.

Заключение

Физические неклонированные функции (PUF) могут обеспечивать недорогую аутентификацию ИС и генерировать летучие ключи как для симметричных, так и для асимметричных криптографических операций. Использование PUF в качестве идентификаторов ИС, СБИС, включая микропроцессоры, позволит маркировать микросхемы беззатратным, высоконадёжным способом. Ключи сами формируются в процессе производства. С ростом сложности чипа растёт и надёжность маркировки. По сути PUF чипа является электронным аналогом отпечатка пальца человека.

Библиографический список

1. B. Gassend, D. Clarke, M. van Dijk and S. Devadas. Controlled physical random functions. In Proceedings of 18th Annual Computer Security Applications Conference, December 2002.
2. J.-W. Lee, D. Lim, B. Gassend, G. E. Suh, M. Van. Key in integrated circuits with identification and authentication applications. In Proceedings of the IEEE VLSI Circuits Symposium, June 2004.
3. D. Lim. Extracting secret keys from integrated circuits. Master's thesis, Massachusetts Institute of Technology, May 2004.
4. R. Pappu. Physical One-Way Functions. PhD thesis, Massachusetts Institute of Technology, 2001.
5. S. R. Nassif. Modeling and forecasting of manufacturing variations. In Proceedings of ASP-DAC2001, 2001
6. P. Tuyls, B. Skoric, S. Stallinga, A. Akkermans, and W. Oprey. Information theoretical security analysis of physical unclonable functions. In Proceedings of Conference on Financial Cryptography and Data Security 2005, volume 3570 of Lecture Notes in Computer Science, 2005.
7. D. S. Boning and J. E. Chung. Statistical metrology: Understanding spatial variation in semiconductor manufacturing. In Proceedings of SPIE 1996 Symposium on Microelectronic Manufacturing, 1996.
8. K. A. Bowman, S. G. Duvall, and J. D. Meindl. Impact of die-to-die and within die parameter fluctuations on the maximum clock frequency distribution for gigascale integration. Journal of Solid-State Circuits, 37(2):183–190, February 2002.
9. C. W. O'Donnell, G. E. Suh, and S. Devadas. PUF-based random number generation. In MITCSAIL CSG Technical Memo 481, November 2004.
10. B. Skoric, P. Tuyls, and W. Oprey. Robust key extraction from physical unclonable functions. In Proceedings of the Applied Cryptography and Network Security Conference 2005, volume 3531 of Lecture Notes in Computer Science, 2005.
11. G. E. Suh, C. W. O'Donnell, I. Sachdev, and S. Devadas. Design and implementation of the aegis single-chip secure processor using physical random functions. In Proceedings of the 32nd Annual International Symposium on Computer Architecture, June 2005.

УДК 681.391; ГРНТИ 50.33

СОВРЕМЕННЫЕ МЕТОДЫ КОРРЕКЦИИ ОШИБОК В ПАМЯТИ

Чан Туан Зунг

*Рязанский государственный радиотехнический университет имени В.Ф.Уткина,
Российская Федерация, Рязань, dungdk4@gmail.com*

Аннотация. В работе рассматриваются современные методы коррекции ошибок для накопителей SSD на основе NAND флеш - памяти. Приводятся их основные характеристики, обсуждаются достоинства, недостатки, а также и области применения.

Ключевые слова: методы коррекции ошибок, накопитель SSD, NAND флеш - память, коды Хемминга, коды Рида – Соломона, коды БЧХ, коды LDPC.

MODERN METHODS OF CORRECTION OF ERRORS IN MEMORY

Tran Tuan Dung

*Ryazan State Radio Engineering University named after V.F.Utkin
Russia, Ryazan, dungdk4@gmail.com*

Annotation. The paper discusses modern methods of error correction for SSD drives based on NAND flash memory. The main characteristics, advantages, disadvantages and areas of application in memory are discussed.

Keywords: error correction methods, SSD drive, NAND flash memory, Hamming codes, Reed-Solomon codes, BCH codes, LDPC codes.

Помехоустойчивые коды (ЕСС) используются во флэш-памяти NAND для обнаружения и исправления битовых ошибок. С уменьшением размеров технологических узлов и увеличением сложности памяти частота ошибок по битам продолжает расти. При массовом использовании устройств MLC / TLC, где 2 или 3 бита данных хранятся в каждом транзисторе с плавающим затвором, эта проблема стала еще более важной, и для ее решения внедряются мощные схемы ЕСС. Методы коррекции ошибок позволяют исправлять ошибки в памяти и бороться с повреждением данных.

В настоящее время существует много методов коррекции ошибок во Флэш-памяти NAND, например коды Хемминга, коды Рида-Соломона, коды БЧХ, коды с низкой плотностью проверок на четность – Low-density parity-check code (LDPC). В данной статье мы рассмотрим особенности использования, а также достоинства и недостатки этих методов.

Исправление ошибок во флэш-памяти NAND выполняется путем добавления битов ЕСС на каждую страницу пользовательских данных, так что добавленных битов ЕСС будет достаточно для исправления одного или нескольких ошибочных битов на странице.

Коды Хемминга

Когда использовались только устройства SLC, требованием ЕСС было исправление одного бита на одной странице данных. Поскольку размер страницы обычно составлял 512 байт, для обеспечения однобитовой коррекции для 512 байт пользовательских данных использовались коды Хэмминга. Для кодов Хэмминга при длине блока 2^n бит необходимое количество проверочных битов составляет n бит. Таким образом, для 512 байт данных количество требуемых проверочных битов равно 12 [1].

Код Хемминга подходит только для исправления ошибок в устройствах SLC из-за ограниченных возможностей исправления ошибок.

Коды БЧХ и коды Рида-Соломона

Коды Рида-Соломона и БЧХ используются для исправления многобитовых ошибок. Коды Рида-Соломона [11] – это линейные блочные коды, которые представляют собой особый класс кодов БЧХ.

Коды БЧХ: $(n - k) \leq mt$;

Коды Рида-Соломона: $(n - k) = 2t$;

при:

n – длина кодового блока ($n = 2^m - 1$);

k – длина информационной части сообщения;

t – количество исправляемых ошибок.

Для кодов БЧХ $(n - k) \leq mt$. Поскольку $(n - k)$ – это количество проверочных битов, максимальное количество проверочных битов равно mt .

Для кодов Рида-Соломона для k -символьного сообщения добавляются $2t$ проверочных символов чтобы сгенерировать кодовое слово Рида-Соломона размером n символов. Такой код может исправить t символьных ошибок. Минимальное расстояние для кода Рида-Соломона составляет $2t + 1 = (n - k + 1)$, и поэтому коды Рида-Соломона являются оптимальными, поскольку минимальное расстояние $(n - k + 1)$ является максимально возможным значением для линейного кода длина блока n [2].



Рис. 1. Кодовое слово Рида-Соломона

Чтобы проанализировать эффективность кодов БЧХ было выполнено моделирование с использованием кодов БЧХ в канале с АБГШ (с аддитивным белым гауссовским шумом) при использовании модуляции BPSK (двоичная фазовая манипуляция). Параметры кодов БЧХ, используемых при моделировании, следующие:

Таблица 1. Параметры кодов БЧХ (BCH) при моделировании

Кодовая скорость	Длина блока	Длина сообщения	Количество ошибок, которые можно исправить
0,87	255	223	4
0,86	511	439	8
0,84	1023	863	16

На рисунке 2 приведены результаты моделирования, проведенного с кодами БЧХ с различной длиной блока. По мере увеличения длины блока t увеличивается, и, следовательно, количество требуемых проверочных битов также увеличивается.

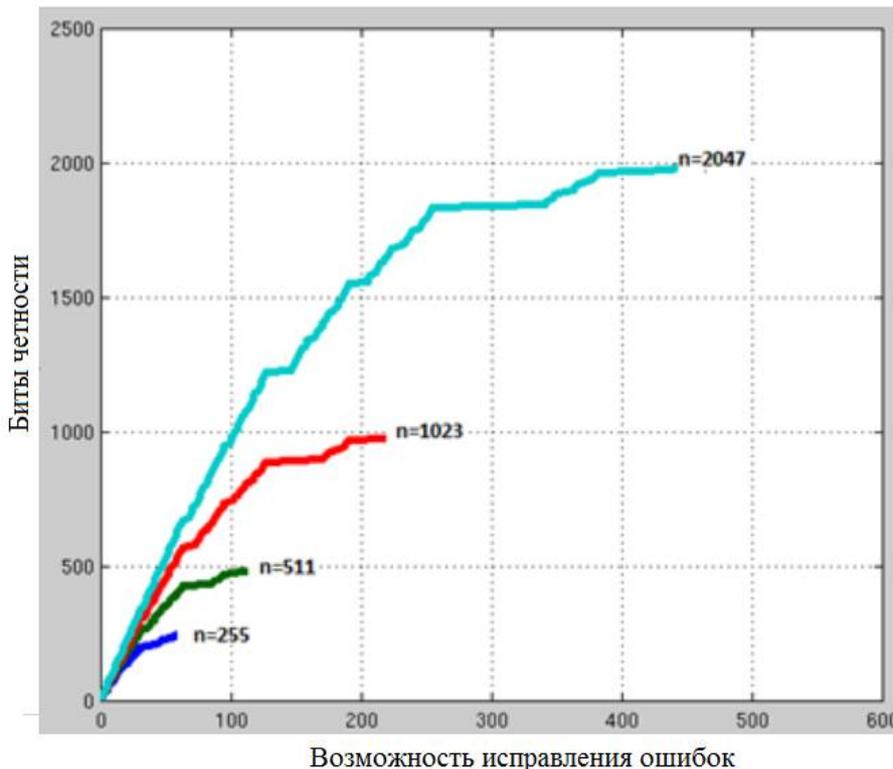


Рис. 2. Коды БЧХ с различной длиной блока

Для различных длин блоков кода БЧХ была проанализирована зависимость эффективности от кодовой скорости, которая показана на рисунке. 3. Для одинакового количества исправляемых ошибок увеличение длины блока приводит к увеличению кодовой скорости. Таким образом, более длинные блоки имеют лучшую эффективность, чем более короткие блоки.

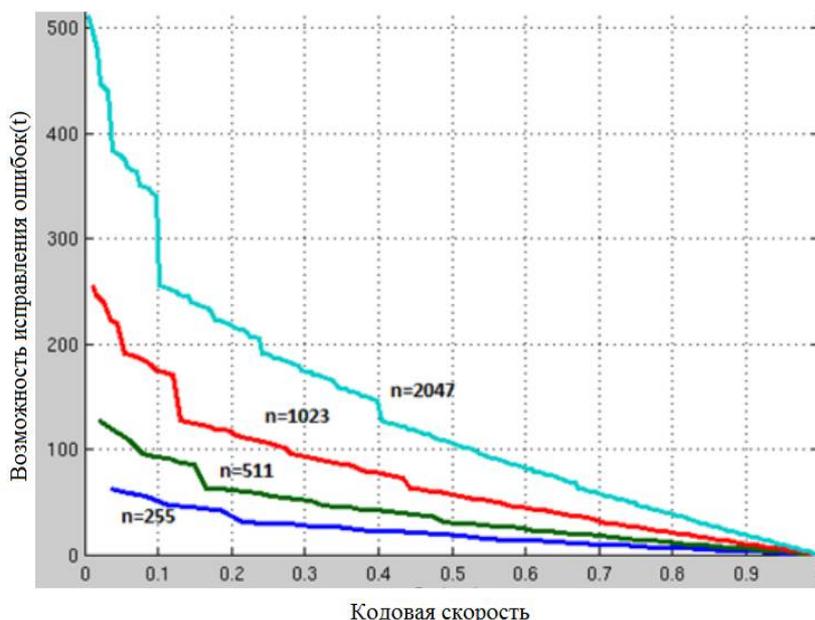


Рис. 3. Эффективность кодов БЧХ для различных длин блоков

Результаты моделирования, при которых получались зависимости вероятности битовой ошибки после декодирования от отношения сигнал/шум в канале связи, представлены на рисунке 4.

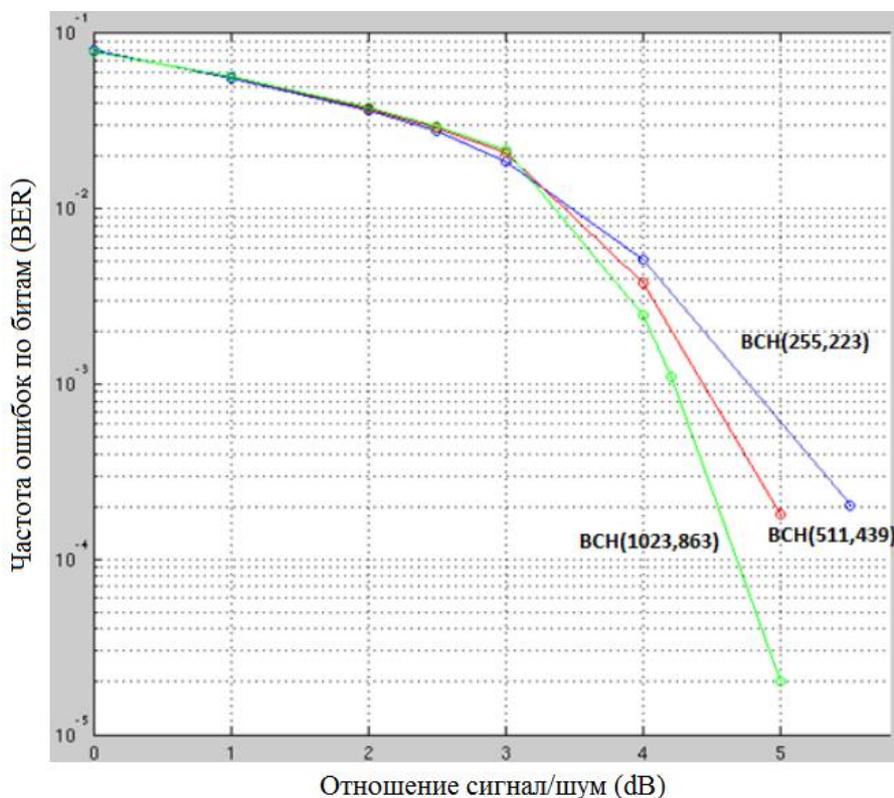


Рис. 4. Эффективность кодов БЧХ

В результате можно сделать следующие выводы:

Результаты моделирования показывают, что по мере увеличения кодовой скорости эффективность кодов БЧХ ухудшается.

Можно видеть, что с точки зрения числа исправляемых ошибок код БЧХ (511, 439) эквивалентен коду БЧХ (255, 223) (оба могут исправить 8 ошибок). Однако БЧХ (511, 439) по кодовой скорости оказывается эффективнее, чем БЧХ (255, 223).

Коды Рида-Соломона эффективны для исправления пакетных ошибок, поскольку исправление одного символа может исправить несколько кластеризованных битовых ошибок. Однако, как показывает практика, ошибки, встречающиеся в NAND Flash, чаще имеют случайное распределение, появляются индивидуальные битовые ошибки, а не сгруппированные (пакетные) ошибки. Для такого типа распределения ошибок коды БЧХ более эффективны, и по этой причине коды БЧХ более подходят для исправления ошибок во флэш-памяти NAND [3].

Коды LDPC

Коды LDPC особенно подходят для использования во флэш-памяти NAND из-за их способности обрабатывать как жесткую (двоичную), так и мягкую (вероятности битов) информацию. Таким образом, с доступностью как жестких, так и мягких данных (как в случае с флэш-памятью NAND) эффективность кодов LDPC довольно близка к теоретическим пределам, то есть пределу пропускной способности Шеннона [4].

Для анализа эффективности кодов LDPC было выполнено их моделирование для канала с АБГШ с использованием модуляции QPSK (четырехпозиционная фазовая модуляция). Параметры LDPC кодов, используемых при моделировании, следующие:

Таблица 2. Параметры LDPC кодов при моделировании

Кодовая скорость	Размеры N - матрицы
1/2	32400x64800
2/3	21600x64800
4/5	12960x64800
9/10	6480x64800

Зависимости вероятности битовой ошибки на выходе декодера от отношения сигнал/шум, полученные с помощью компьютерного моделирования, показаны на рисунке 5.

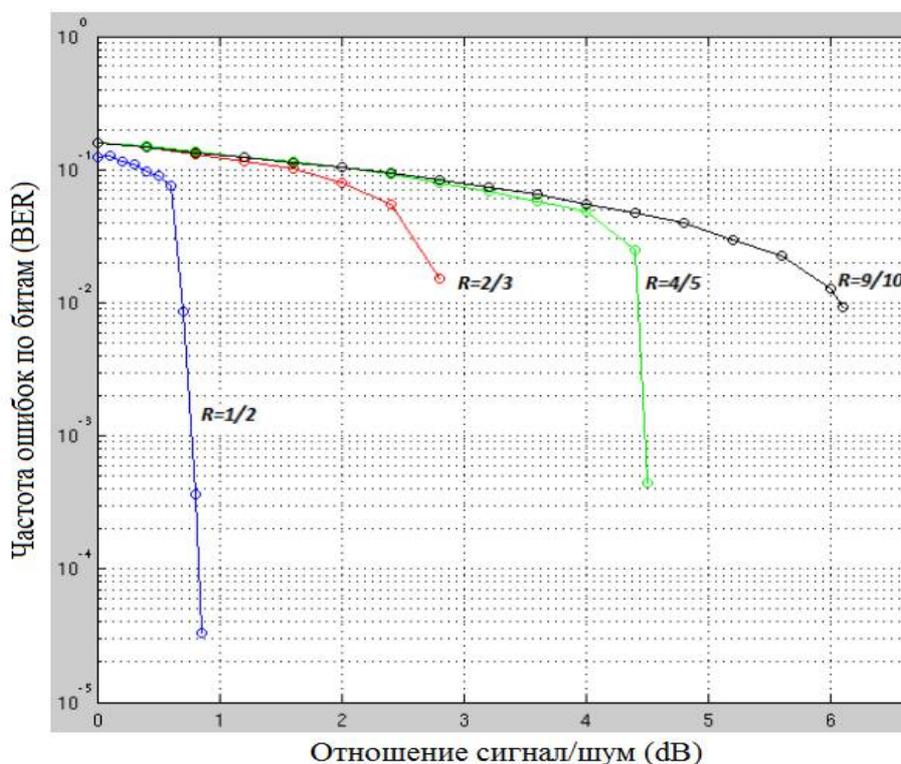


Рис. 5. Эффективность LDPC кодов

Результаты моделирования показывают, что по мере увеличения кодовой скорости эффективность кодов LDPC ухудшается.

Таким образом, можно сделать общие выводы о том, что:

1. Код Хемминга подходит только для исправления ошибок в устройствах SLC из-за ограниченных возможностей исправления ошибок. Коды Рида-Соломона и БЧХ используются для исправления многобитовых ошибок.

2. Коды Рида-Соломона эффективны для исправления пакетных ошибок, поскольку исправление одного символа может исправить несколько кластеризованных битовых ошибок. Однако ошибки, встречающиеся в NAND Flash, чаще имеют случайное распределение, индивидуальные битовые ошибки, а не сгруппированные (пакетные) ошибки. Для такого типа распределения ошибок коды БЧХ более эффективны, и по этой причине коды БЧХ лучше подходят для исправления ошибок во флэш-памяти NAND.

3. Коды LDPC лучше подходят для использования во флэш-памяти NAND из-за их способности обрабатывать как жесткую (двоичную), так и мягкую (вероятности битов) информацию. Они чаще используются для исправления ошибок в устройствах MLC и TLC. Коды LDPC обеспечивают лучшее исправление ошибок, чем коды БЧХ при той же кодовой

скорости. Они хорошо работают, когда доступны как мягкая, так и жесткая битовая информация. При этом LDPC коды обеспечивают показатели эффективности, близкие к Шеннону для блоков большой длины.

Библиографический список

1. Овечкин Г.В., Помехоустойчивое кодирование символьной информации в цифровых системах передачи и хранения данных 2006.
2. Золотарев В.В., Овечкин Г.В. Помехоустойчивое кодирование. Методы и алгоритмы. Справочник. М.: Горячая линия – Телеком, 2004.
3. <https://siblec.ru/telekommunikatsii/teoreticheskie-osnovy-tsifrovoj-svyazi>.
4. Sarah Johnson, Introducing Low-Density Parity-Check Codes, ACoRN Spring School 2006.

УДК 004.03; ГРНТИ 20.53

ОСОБЕННОСТИ ФУНКЦИОНИРОВАНИЯ ИНФОРМАЦИОННО-АНАЛИТИЧЕСКИХ СИСТЕМ

А.Ю. Стеблюк

*Рязанский государственный радиотехнический университет имени В.Ф. Уткина,
Российская Федерация, Рязань, steblyuk98@mail.ru*

Аннотация. В работе рассматриваются особенности организации и функционирования информационно-аналитических систем, указываются их основные составляющие, а так же отмечаются возможные сферы их применения.

Ключевые слова: информационно-аналитические системы, хранилище данных, витрины данных, транзакционные системы, аналитическая пирамида

FEATURES OF FUNCTIONING OF INFORMATION-ANALYTICAL SYSTEMS

A.Yu. Steblyuk

*Ryazan State Radio Engineering University named after V.F. Utkin,
Russia, Ryazan, steblyuk98@mail.ru*

The summary. The paper examines the features of the organization and functioning of information and analytical systems, indicates their main components, as well as notes the possible areas of their application.

Keywords: information and analytical systems, data warehouse, data marts, transactional systems, analytical pyramid.

В современном мире, где информационные технологии совершили значительный скачек, возникла потребность в контроле и мониторинге больших объемов данных, циркулирующих в рамках работы какого-либо предприятия, обрабатывать которые вручную человек не в силах. Поэтому инженерами были разработаны специализированные системы для контроля и мониторинга процессов и потоков данных, протекающих в производственном предприятии. Разновидностей таких систем существует широкое множество, в данной работе следует обратить особое внимание на информационно-аналитические системы.

Информационно-аналитические системы представляет собой отдельный тип информационных систем, предназначенных для анализа и последующей обработки потоков данных. Основная задача таких систем заключается в осуществлении поиска, сбора, последующей обработки и представлении полученных данных в виде, пригодном для восприятия, анализа и применения оператором такой системы [1]. Заявленный функционал позволяет организовать оптимальный управленческий или иного рода процесс, протекающий в рамках конкретного предприятия.

Говоря об уровнях, на которых осуществляется функционирование информационно-аналитических систем, можно отметить следующие:

1. Уровень обнаружения, сбора, хранения и передачи информации;
2. Уровень аналитики, на котором осуществляется обработка поступающей информации, а именно её обобщение, сортировка и выполнения анализа, в результате которого формируются выводы с последующим составлением прогнозов.

В соответствии с отмеченными уровнями современные информационно-аналитические системы, служащие для обеспечения специалистов данными о различного рода вопросах и проблемах в работе предприятия, состоят из определенного набор подсистем. В классическом варианте реализации информационно-аналитической системы, выделяют следующие две её составляющих:

1. Подсистема, предназначенная для сбора и хранения информации о предприятии. Она осуществляет сбор и фильтрацию полученных данных, а также последующее накопление информации, обеспечивая её оперативное применение для предстоящих аналитических операций [4, 5, 6].

2. Подсистема, отвечающая за доступ к информации, её дальнейший анализ и составление отчетности обслуживаемой организации [3]. Обычно содержит в своём составе блоки, обеспечивающие доступ к требуемой информации, её безопасность и своевременную блокировку несанкционированного доступа к ней.

Подсистемы, принадлежащие к первому типу, основаны на технологии оперативной обработки транзакций, которая содержит в своей основе концепцию хранилища данных. Хранилище данных служит для хранения достоверных обобщенных данных о предприятии, отражающих его работу за длительный временной промежуток, а также информацию о внешней среде его деятельности. Зачастую, вместе с хранилищем информации, для достижения положительного эффекта применяют информационные витрины, представляющие собой специальные базы аналитики данных, организованные на основе одного из отдельных подразделений или направлений работы предприятия.

Аналитическая пирамида

Ранее упомянутые составляющие подсистем могут быть отражены в «аналитической пирамиде», представляющую собой информационную инфраструктуру (рисунок 1):



Рис. 1. Аналитическая пирамида

Аналитическая пирамида, по своей сути, представляет собой организованную определенным образом иерархическую структуру, в которой различные классы информационных систем отнесены на отдельные уровни. В её основании лежат транзакционные системы, принципы которых заложен в суть работы и функционирования информационно-аналитических систем. По мере движения от основания пирамиды к её вершине происходит агрегирование обрабатываемых данных в информацию, предназначенную для поддержки принятия управленческих решений.

Возвращаясь к теме информационно-аналитических систем, важно подчеркнуть похожую тенденцию с «аналитической пирамидой» - главной задачей формирования таких систем является передвижение информационной базы предприятия из неопределенной среды в логическим образом организованную систему, являющуюся информационным источником, который способен удовлетворить основные нужды и потребности предприятия [2, 3].

Как говорилось ранее, в основе базиса информационно-аналитических систем лежит хранилище информационных данных, которое:

1. Считается информационной базой, служащей лишь для чтения;
2. Подвергается обновлению из операционных систем на постоянной основе;
3. Предполагает один подход к обозначению и сохранению информации, независимо от её организации.

В качестве прикладного примера приведем схему структуры некоего абстрактного предприятия (рисунок 2):

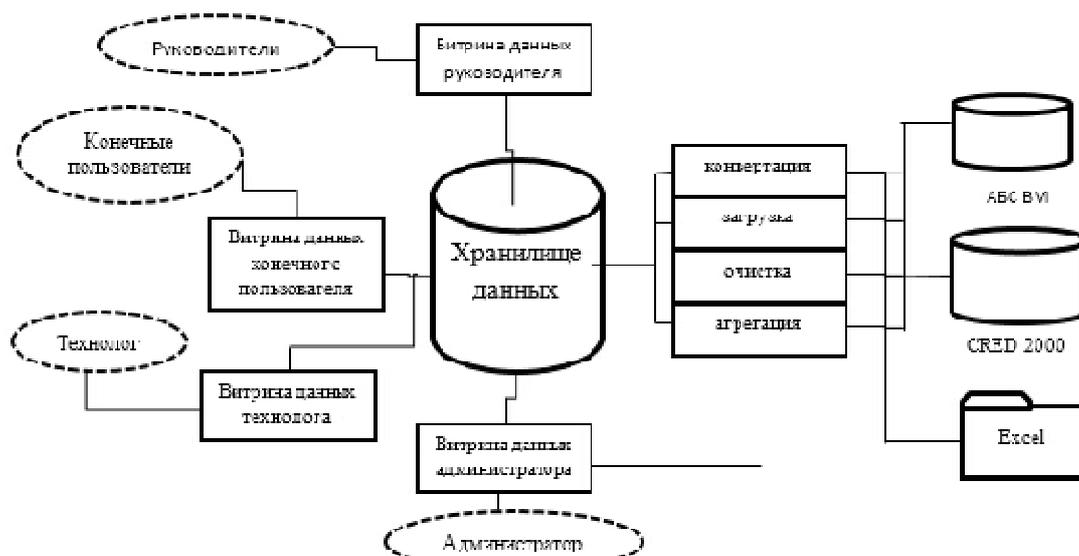


Рис. 2. Структурная схема информационно-аналитической системы

Как видно из приведенного рисунка, в данной системе используется несколько витрин данных, которые обеспечивают представление аналитических данных для каждого конкретного вида пользователей, будь-то руководитель предприятия, администратор или конечный пользователь, например, обычный оператор данной системы. В центре схемы расположено хранилище данных, служащее для накопления и безопасного хранения общих данных предприятия, полученных в процессе его работы.

Интерфейс информационно-аналитических систем

Также, вспоминая основные функции информационно-аналитических систем, стоит подробнее остановиться на их способности корректного отображения данных для пользователя данной системы.

Для обеспечения такой возможности, помимо использования витрин данных для каждого вида пользователей, следует грамотно подойти к проектированию интерфейса данных систем. С точки зрения представления информации главным принципом должно являться абстрагирование от лишних подробностей обработки данных и работы системы, а также вывод нужной информации в форме, доступной для восприятия пользователя.

Что касается последнего, учеными были выявлены некоторые принципы проектирования визуального интерфейса пользователя, которые позволяют достичь наибольшей наглядности представления информации, что приводит к приросту производительности работы пользователя такой системы.

Основными принципами проектирования интерфейсов систем являются:

1. Принцип «Золотого сечения»;

Данный принцип заключается в правильных пропорциях информационных блоков, что повышает наглядность представляемой информации.

2. Кошелек Миллера;

Этот принцип гласит, что в дизайне интерфейсов стоит группировать элементы блоков не более, чем семь элементов в группе.

3. Принцип группировки;

Согласно этому правилу, интерфейс программы должен быть разбит на ясно очерченные блоки элементов с заголовком для каждого.

4. Бритва Оккама и др.

Перечисленные принципы проектирования интерфейсов являются наиболее широко используемыми в современных прикладных инженерных решениях. С помощью них можно достичь значительного прироста в продуктивности работы оператора той или иной системы, что, в свою очередь, повысит качество принимаемых решений по дальнейшей политике функционирования предприятия и увеличит его прибыль.

Пример разработанного интерфейса системы

Приведем возможную реализацию интерфейса информационно-аналитических систем, который был разработан в процессе проведения данного исследования (рисунок 3).

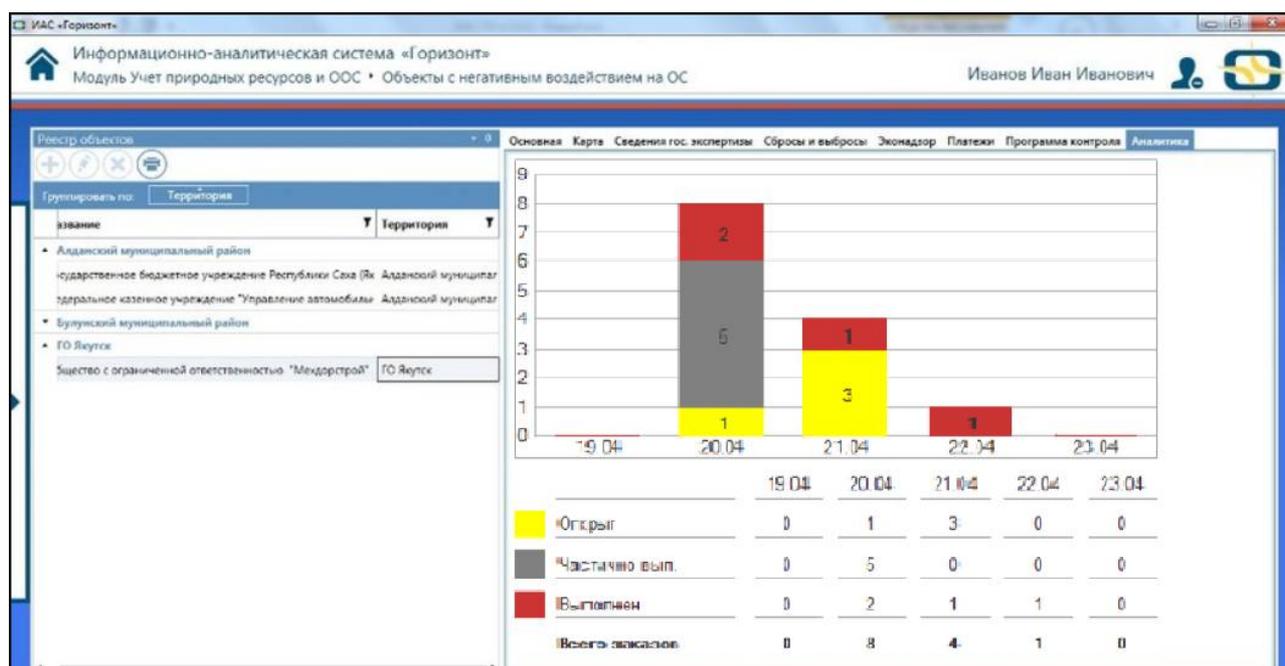


Рис. 3. Пример интерфейса информационно-аналитической системы

Приведенный интерфейс позволяет представить информацию об учете природных ресурсов и выполнении заказов на осуществление замеров выбросов в атмосферу в удобном для пользователя виде. В процессе проектирования данного интерфейса были учтены приведенные ранее принципы, например, принцип «Золотого сечения», в соответствии с которым были выбраны размеры информационных блоков для акцентирования внимания пользователя на особенно важных частях интерфейса.

Заключение

Таким образом, информационно-аналитические системы позволяют значительно упростить процесс мониторинга и контроля информации в течении работы производственного предприятия, а грамотно спроектированный интерфейс данных систем дает значительное увеличение качества работы и продуктивности оператора информационно-аналитической системы. Итогом проведенной работы является разработанный интерфейс для информационно-аналитической системы по контролю за природными ресурсами и выбросами в атмосферу.

Библиографический список

1. Исаев Д.В. Системы бизнес-интеллекта и аналитические приложения. [Текст] - Финансовая газета, 2005, №31 (711). – с.14-15, №32 (712). – с.15.
2. Кравченко Т.К., Пресняков В.Ф. Инфокоммуникационные технологии управления предприятием. [Текст] – М.: ГУ-ВШЭ, 2003. – 272 с.
3. Лодон Дж., Лодон К. Управление информационными системами / Пер.с англ. под ред. Д.Р.Трутнева. [Текст] – СПб.: Питер, 2005. – 912 с.
4. Исаев Д.В. Аналитические информационные системы. [Текст] – М.: ГУ-ВШЭ, 2008. – 60 с.
5. Каплан Р., Нортон Д. Стратегическое единство. Создание синергии организации с помощью сбалансированной системы показателей. [Текст] – М.: Вильямс, 2006. – 384 с.
6. Информационно-аналитические системы: актуальность и примеры [Электронный ресурс] – URL: https://spravochnick.ru/informatika/informacionno-analiticheskie_sistemy_aktualnost_i_primery/

УДК 004.054; ГРНТИ 50.13.15

ОБЗОР И АНАЛИЗ МЕТОДОВ ОЦЕНКИ СОСТОЯНИЯ ОБОРУДОВАНИЯ

Е.С. Кабанова

*Рязанский государственный радиотехнический университет имени В.Ф. Уткина,
Российская Федерация, Рязань, i.am.kabanova.e.s@gmail.com*

Аннотация. В работе рассматриваются различные методы оценки состояния оборудования. Приводятся их основные особенности, достоинства и недостатки, а также области применения.

Ключевые слова: микропроцессорная система (МПС), внутрисхемный эмулятор (ВСЭ), постоянное запоминающее устройство (ПЗУ), логический анализатор (ЛА), память логических последовательностей (ПЛП).

REVIEW AND ANALYSIS OF EQUIPMENT STATE ASSESSMENT METHODS

E.S. Kabanova

*Ryazan State Radio Engineering University named after V.F. Utkin,
Russia, Ryazan, i.am.kabanova.e.s@gmail.com*

Annotation. The paper discusses various methods for assessing the state of equipment. Their main features, advantages and disadvantages, as well as areas of application are given.

Key words: microprocessor system (MPS), in-circuit emulator (VSE), read-only memory (ROM), logic analyzer (LA), logical sequence memory (PLP).

На предприятиях, осуществляющих все этапы жизненного цикла производства оборудования, часто используется термин «техническое состояние». Техническое состояние описывает состояние оборудования в определенный момент времени при определенных условиях и регулируется специальной документацией. Контроль технического состояния - проверка соответствия полученных реальных значений требованиям, установленным документацией. Далее на основании полученных данных определяется один из заданных типов технического состояния оборудования в данный момент времени.

С целью установления фактического состояния оборудования и дальнейшего исключения возможности ошибок, сбоев и неисправностей на производстве проводятся диагностические мероприятия, технические обследования и различные тестирования.

В производстве оборудования на основе микропроцессорных систем (МПС) важным этапом является тестирование именно этой составляющей, ведь процессор, являющийся «мозгом» системы, координирует выполнение основного функционала оборудования. В случае неисправности или ненадлежащей работы МПС неизбежны существенные, а порой и фатальные ошибки в работе оборудования.

В настоящее время работа с микропроцессорными системами почти всегда предполагает работу с однокристальными микроконтроллерами, сочетающими высокую универсальность и доступность. Рассмотрим основные средства контроля, диагностики и отладки МПС с учетом использования этого класса микропроцессорных систем.

Отладка оборудования включает тестирование отдельных устройств микропроцессорной системы. Осциллографы, вольтметры, амперметры, частотомеры, генераторы импульсов и анализаторы сигналов широко используются для автономной отладки оборудования на схемном уровне. На более высоком уровне используются внутрисхемные эмуляторы, эмуляторы ПЗУ, логические анализаторы, отладочные платы, а также специальные средства отладки, которые встраиваются в БИС на этапе их разработки [1].

Внутрисхемные эмуляторы

Самым мощным и универсальным инструментом отладки является внутрисхемный эмулятор (ВСЭ), он делает процесс работы управляемым и изменяемым. ВСЭ - это программный и аппаратный инструмент, который может заменить эмулируемый МС в реальной схеме [1]. Аппаратный интерфейс позволяет программисту или программному обеспечению отлаживать встроенную систему. ВСЭ временно устанавливается между встроенной системой и внешним терминалом или персональным компьютером, чтобы программист мог наблюдать и изменять происходящее во встроенной системе, не имеющей собственного дисплея или клавиатуры. ВСЭ служит «суррогатным» ЦП (центральным процессором) для микрокомпьютера во встроенной системе (рисунок 1). ВСЭ может помочь инженерам-проектировщикам в разработке продукта, а также помочь программистам или конечным пользователям в обновлении, изменении или сопровождении продукта. Используя ВСЭ, технические специалисты могут тестировать новые, измененные элементы прошивки встроенного устройства без сохранения и записи изменений. Этот процесс гарантирует, что конечный продукт функционирует именно так, как ожидают продавец и потребитель.

По мере роста сложности процессора и, что более важно, тактовой частоты, ВСЭ стали слишком дорогими, и их доступность снизилась. Сейчас они, как правило, доступны только для бюджетных устройств.

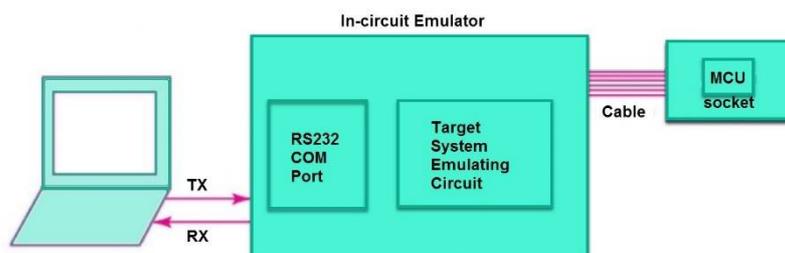


Рис. 1. Принцип работы внутрисхемного эмулятора

Оценочные платы

С внутрисхемными эмуляторами тесно связан класс инструментов отладки, таких как оценочные платы. Они являются своеобразными конструкторами для создания прототипов электронных устройств. Хотя их часто путают с другими типами плат, оценочные платы призваны помочь в оценке некоторых основных компонентов системы. Эти платы обычно содержат сложное полупроводниковое устройство, такое как датчик, процессор, микропроцессор. Помимо основного компонента, могут быть различные типы общих входных и выходных разъемов, память, графические дисплеи и встроенные платы, блоки питания. Оценочная плата используется для определения того, подходит ли полупроводниковый компонент для конкретного применения. Отладочная плата - это производственная версия оценочной платы со всеми связанными с ней компонентами и разъемами.

Эмуляторы ПЗУ

При создании тестовых воздействий в МПС используется эмулятор ПЗУ - фактически, подключаемый модуль оперативной памяти, который заменяет ее во время отладки, что помогает пользователю избежать многократной перезаписи памяти. Для выполнения операций, связанных с доступом к памяти, эмулятор должен быть подключен к внешнему устройству, роль которого, чаще всего, выполняет внешний компьютер. По каналу связи программа загружается с компьютера в оперативную память эмулятора. Для согласования эмулятора и микропроцессорной системы используется буферный адаптер, выполненный в виде сменного блока и индивидуально подобранный в зависимости от характеристик МПС. Эмуляторы ПЗУ также используются для имитации внутренней памяти команд микроконтроллера. После отладки программа с помощью программатора заносится в энергонезависимую память контроллера и тестируется без использования эмулятора на реальных данных. Реализация эмуляторов ПЗУ достаточно недорогая и простая, поскольку эмуляция самих функций микроконтроллера не происходит. Это делает эти устройства универсальными и пригодными для работы с различными типами микроконтроллеров.

Для качественной отладки МПС требуются следующие возможности инструментов [1]:

- синхронный мониторинг состояния многоразрядных шин;
- регистрация последовательностей состояний в связи с редкими и разовыми событиями;
- высокая частота работы.

Логические анализаторы

Простое оборудование для контроля цифровых схем (амперметры, вольтметры, осциллографы) не удовлетворяет заданным требованиям, в отличие от логических анализаторов (ЛА), основной областью применения которых является этап разработки микропроцес-

сорной системы. На этом этапе, когда проверенное оборудование и отлаженные программы недоступны, проявляется универсальность и гибкость логических анализаторов, помогающих найти причины неисправностей. Память логических последовательностей (ПЛП) является основным элементом логических анализаторов. Тактовые сигналы для записи в ПЛП могут поступать либо от внутреннего генератора ЛА, либо от тестируемой схемы, и в этом случае эти сигналы могут стробироваться другими внешними сигналами. Запись в ПЛП происходит циклически, поэтому при заполнении памяти запись будет продолжаться с нулевого адреса. То есть в памяти всегда хранится количество последних принятых слов, равное емкости ПЛП. Режим регистрации продолжается до наступления определенного события или группы событий, указанных оператором, и может быть прекращен либо при наступлении события, либо через определенное количество тактов после него. В случае отложенного прекращения регистрации, состояния сигналов в контрольных точках будут записаны в моменты времени до наступления события и сразу после него, что значительно облегчает анализ (рис. 2).

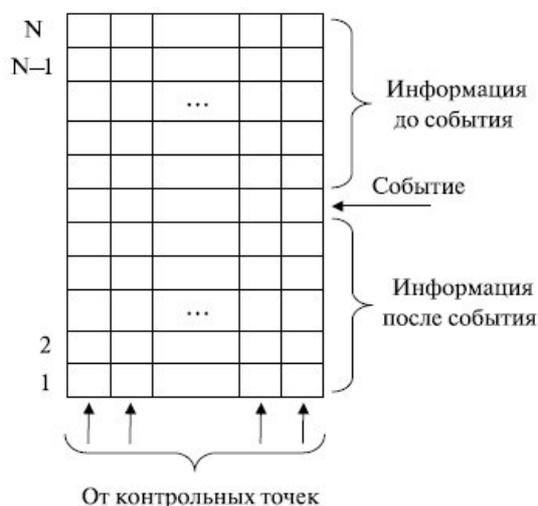


Рис. 2. Структура памяти логических последовательностей

В современных логических анализаторах существует огромное множество способов подключения, условий запуска и режимов отображения. Высокая скорость и большая емкость ПЛП делают эти устройства универсальными инструментами для отладки, а наличие генератора тактовых последовательностей позволяет использовать их в качестве автономных устройств для поиска и устранения неисправностей. Использование логического анализатора при оценке состояния оборудования требует наличия высококвалифицированного персонала. Для поиска новой неисправности необходимо перенастроить прибор и разработать методику испытаний, поэтому на стадии изготовления и эксплуатации он малоприменим. Здесь получили более широкое распространение принципы сигнатурного анализа, которые, не требуя высокой квалификации персонала, позволяют быстро и точно найти вышедший из строя компонент МПС с использованием недорогого и компактного оборудования.

Сигнатурные анализаторы

В мае 1977 был разработан сигнатурный анализ. Этот подход опирается на уникальный алгоритм сжатия потоков данных. Каждая точка системы характеризуется некоторым числом, называемым сигнатурой (ключевым кодом).

В основе сигнатурного анализа лежит метод преобразования длинных последовательностей двоичных сигналов в сигнатуры. Базовый анализатор сигнатур принимает входные данные от тестируемого узла. Пусковой и конечный импульсы фиксируются для запуска и

завершения выборки. Затем, импульсы от тестируемого узла передаются в сдвиговый регистр для получения шестнадцатеричного эквивалента сигнала. Если сигнатура, полученная анализатором сигнатур, совпадает с сигнатурой того же узла на заведомо исправной плате, это означает, что эта область работает правильно (рисунок 3). Путем зондирования различных участков платы с помощью анализатора сигнатур и сравнения результатов с ожидаемыми значениями, часто подробно описанными в руководствах по ремонту и обслуживанию, можно найти проблемную область [2].

Технология позволяет обнаруживать неисправности на уровне компонентов в широком спектре электронных модулей - от бытовой до специализированной электроники. Простота, быстродействие и способность работать практически без участия человека делают использование сигнатурного анализа удобным в любых условиях.



Рис. 3 . Принцип работы сигнатурного анализа

Эмуляторы JTAG

С недавних пор в микропроцессоры и микроконтроллеры начали интегрировать модули с функциями отладки, встроенные непосредственно в микросхему [1]. Каналом связи здесь служит JTAG-интерфейс, от чего такое отладочное оборудование и получило название «эмулятор JTAG». Такие эмуляторы управляются одним контролирующим устройством (чаще всего ПК, оснащенный соответствующим ПО и адаптером интерфейса), к которому могут быть подключены несколько тестируемых устройств. В соответствии с программой, эмулятор JTAG генерирует тестовые сигналы и сравнивает результаты с эталонными. Недостатком использования таких устройств является обязательное наличие встроенной в микроконтроллер Flash-памяти программ, требуемой для загрузки отладочной программы. В других случаях использование эмуляторов JTAG просто невозможно.

При существующем разнообразии возможных тестируемых устройств существует достаточное количество методов тестирования и диагностики для любых условий и принципов работы системы. В зависимости от устройства, стадии его изготовления и проверяемого функционала сотрудник может выбрать любой из доступных методов оценки состояния оборудования как с использованием специализированных устройств, так и без них.

В целях проверки статических режимов работы логических микросхем оптимальным выбором будет применение сигнатурного анализа. Этот метод позволяет оперативно проверять работоспособность устройств с возможностью проверки как новых, так и демонтированных из платы микросхем. Выполнение проверок и тестов в этом случае не требует участия высококвалифицированного персонала. Кроме того, сигнатурный анализ является единственным решением для автоматизированного определения типа микросхемы. Поскольку задача тестирования и определения типа методом сигнатурного анализа микросхем требует наличия интеллектуального устройства для выполнения алгоритма тестирования и базы

данных, содержащей эталонную информацию по микросхемам, возможно проектирование приставки к ПК, подключаемой через внешний порт, или создание специализированного приложения для анализа, работающего непосредственно под управлением компьютера. Это обусловлено наличием в стандартном комплекте IBM-совместимого компьютера многих компонент, необходимых для решения данной задачи (микропроцессора, составляющего основу компьютера; жесткого диска, предназначенного для хранения информации; внешних портов ввода-вывода - последовательных COM1, COM2 и параллельного LPT; клавиатуры и дисплея - для ввода и вывода информации соответственно). Программная реализация выбранного метода для ПК может быть выполнена с использованием любого языка программирования и при необходимости дополнена новым функционалом.

Библиографический список

1. Методы и средства отладки микропроцессорных систем // www.intuit.ru/studies/courses/604/460/lecture (дата обращения: 21.02.2021).
2. Абдуллаев М.М, Болтаев Ж.Б, Улмасхужаев З.А. Анализ модели неисправности и диагностики много-входных микропроцессорных устройств многоканальным сигнатурным анализом // Международный научный журнал "Интернаука". 2017. №3.

УДК 004.042; ГРНТИ 50.41.17

ПРОЕКТИРОВАНИЕ АРХИТЕКТУР СИСТЕМ РАБОТЫ С БОЛЬШИМИ ДАННЫМИ

С.Н. Баранова

*Рязанский государственный радиотехнический университет имени В.Ф. Уткина,
Российская Федерация, Рязань, baranova.sv.n@gmail.com*

Аннотация. В связи с нарастающим объемом информации появляется необходимость проектировать системы, позволяющие обрабатывать огромные потоки данных. В статье определяются проблемы при разработке архитектуры систем работы с большими данными и рассматриваются примеры существующих вариантов систем.

Ключевые слова: большие данные, архитектура системы, лямбда-архитектура, каппа-архитектура.

DESIGNING ARCHITECTURES FOR BIG DATA SYSTEMS

S.N. Baranova

*Ryazan State Radio Engineering University named after V.F. Utkin,
Russia, Ryazan, baranova.sv.n@gmail.com*

The summary. Due to the growing volume of information, it becomes necessary to design systems that can handle huge data flows. The article identifies problems in the development of the architecture of systems for working with big data and considers examples of existing system options.

Keywords: big data, system architecture, lambda architecture, kappa architecture.

В настоящее время стремительное развитие получает направление разработки систем работы с большими данными. С каждым годом все больше различных областей деятельности начинают сталкиваться с необходимостью хранить, обрабатывать и извлекать полезную информацию из различных источников данных. В связи с этим становится актуальной задача проектирования таких систем, которые удовлетворяли бы потребностям по вычислительной мощности, и, при этом, показывали оптимальное время обработки.

Чтобы понять, какие компоненты при проектировании систем работы с большими данными являются необходимыми, нужно рассмотреть проблемы, возникающие при работе с большими данными. К ним можно отнести следующие:

1. Реализация распределения всех слоев системы для возможности дальнейшей масштабируемости и доступности.

2. Обеспечение работы системы, несмотря на сбои различных аппаратных или программных частей.

3. Обеспечение баланса между быстродействием, надежностью, масштабируемостью и согласованностью данных.

Исходя из перечисленных трудностей, можно сформулировать требования, которые должна обеспечивать система работы с большими данными: возможность записывать и хранить большой объем операций, справляться с переменной нагрузкой, справляться с большим количеством аналитических вычислений, предоставлять высокую доступность [1].

Самым простым изображением системы работы с большими данными является многослойная архитектура [2]. На рисунке 1 показаны основные слои, которые необходимы для обеспечения полной функциональности при работе с большими данными. Выделяют следующие логические слои:

1. Источники больших данных. Источники данных могут выдавать различного рода информацию, в первую очередь, отличающуюся по своей структурированности. К тому же источник может работать в реальном времени или поставлять пакеты данных в определенный момент времени.

2. Слой хранения данных. На этом уровне производится сохранение исходных данных и при необходимости предварительное преобразование (например, к определенному формату).

3. Аналитический слой. Все собранные данные могут быть обработаны и проанализированы, для чего требуется точно определить методы и алгоритмы. В крайних случаях данные могут поступать непосредственно от источника.

4. Слой потребления. Визуализация результата слоя аналитики может быть предназначена для различных типов пользователей: от человека до другой системы.



Рис. 1. Многослойная архитектура

Исходя из выделенных слоев, можно определить набор обязательных компонентов для любой архитектуры системы работы с большими данными. Взаимодействие этих элементов отображено на рисунке 2. К источникам данных можно отнести хранилища различных данных приложений, различные файлы-журналы, получаемые данные в режиме реального времени. Далее система может работать в двух режимах: по мере поступления данных в виде пакетов или в реальном времени.

Прием сообщений в реальном времени также необходимо сохранить для дальнейшей потоковой обработки. Хранилище таких данных должно обеспечивать возможность его горизонтального масштабирования. Далее над полученной информацией проводится обработка, содержащая в себе фильтрацию, статистическую обработку и другие подготовительные операции.

В противном случае данные могут статически поступить в виде пакетов, которые обычно хранятся в распределенном хранилище, позволяющем записывать огромные объемы в различных форматах. Данный вид хранилища также называется озером данных. Далее над пакетами выполняются те же процессы, которые проводятся и над данными, поступающими в режиме реального времени. Существенным отличием является только поступление на вход пакетной обработки длительных пакетных заданий.

Предварительно обработанные данные могут быть записаны в необходимом формате для запросов аналитики, которые должны храниться в специализированном хранилище аналитических данных. Для представления в системах работы с большими данными производится обработка в целях анализа и создания отчетов, благодаря чему пользователь может получить важную выходную информацию.

Если обратить внимание на процессы, проходящие через компоненты, то многие из них являются повторяющимися. В таком случае их можно автоматизировать, воспользовавшись технологиями оркестрации [3].

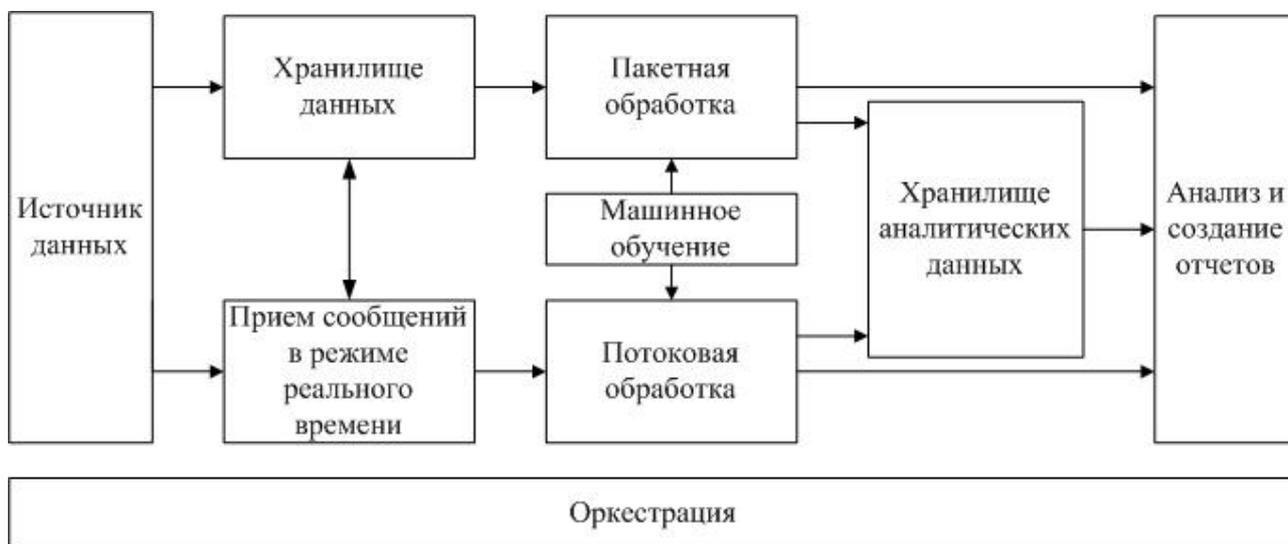


Рис. 2. Взаимодействие компонент архитектуры

На основании выявленных компонент Натаном Марцом была предложена к построению лямбда-архитектура [4], которая заключается в обработке по двум путям, называемым холодным и критическим.

Холодный путь соответствует пакетному уровню. На нем все данные поступают на хранение, а обработка выполняется пакетами. Далее пакеты поступают на уровень обслуживания, где подготавливаются для эффективного выполнения запросов.

Критический путь также называется уровнем ускорения и соответствует обработке в режиме реального времени. Его обработка позволяет обновлять данные на уровне обслуживания при необходимости, если поступают дополнительные данные.

Различие в двух путях исключительно в скорости выполнения и точности вычисление: критический путь поддерживает большую скорость и меньшую точность в отличие от холодного.

Результаты выполнения обоих путей сходятся в приложении аналитики. По какому пути будут получены данные, зависит от конкретной задачи пользователя.

Все данные, попадающие в хранилище по холодному пути, никогда не изменяются и хранятся в том виде, в котором были изначально записаны. Если требуется что-то обновить, то данные записываются отдельно в виде новой записи с отмеченным временем.

Схема лямбда-архитектуры представлена на рисунке 3.

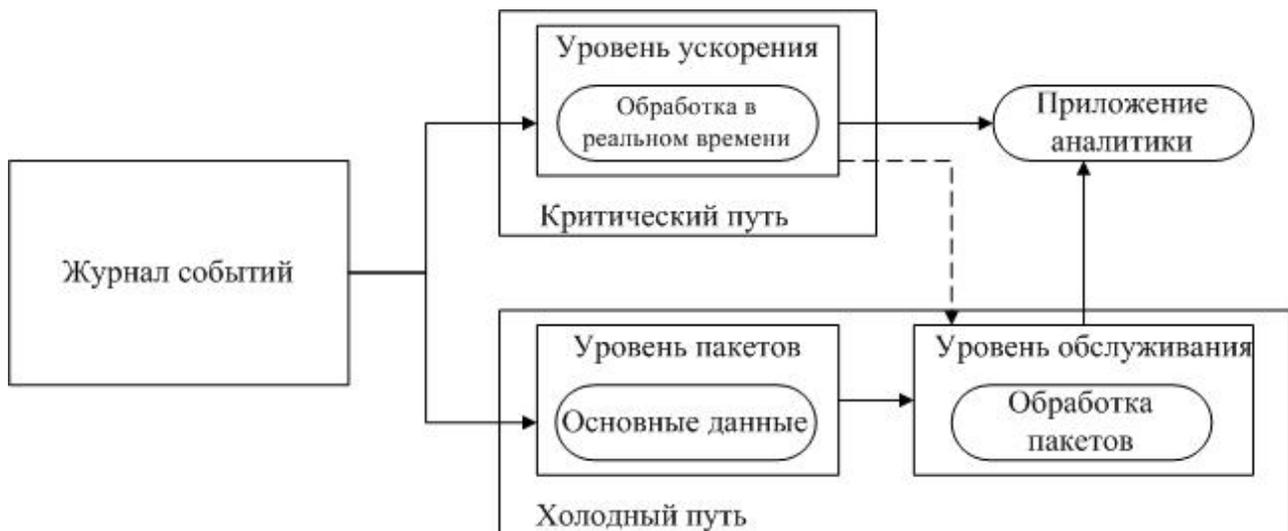


Рис. 3. Лямбда-архитектура

Проблема рассмотренной архитектуры заключается в двойной обработке данных на обоих путях, что усложняет логику работы архитектуры, а также ее управление.

В качестве развития лямбда-архитектуры Джейм Крепсом была предложена каппа-архитектура [4]. Здесь данные также подвергаются разделению, но их обработка проводится исключительно на потоковом пути.

Все данные проходят обработку в реальном времени и также сохраняются в долгосрочном хранилище, где остаются неизменяемыми. При этом если необходимо провести повторную обработку над каким-либо набором данных, то необходимо провести поток к уровню ускорения.

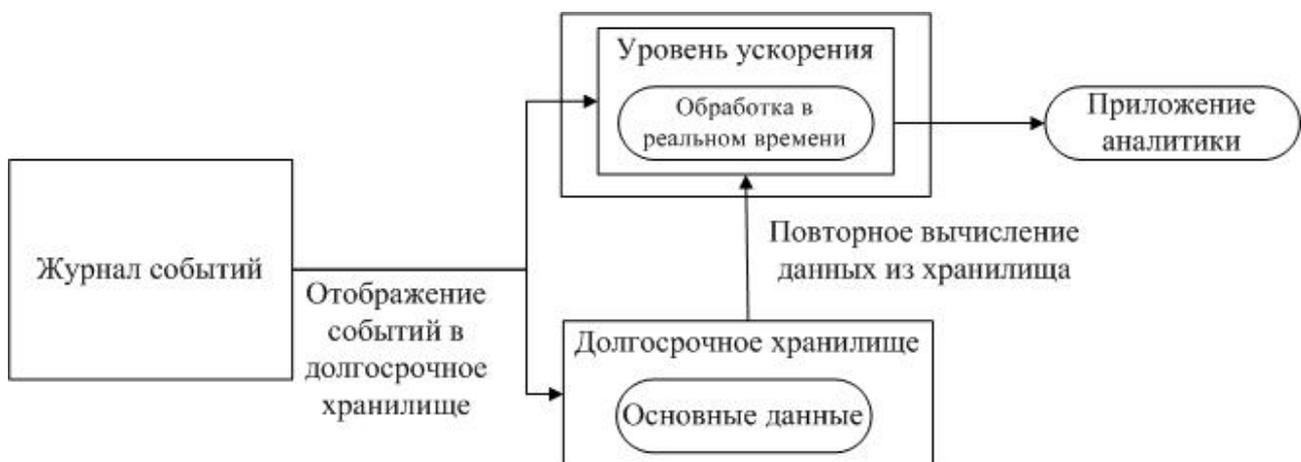


Рис. 4. Каппа-архитектура

Представленные в данной работе архитектуры являются наиболее общими для реализации в системах работы с большими данными. Лямбда применяется при необходимости работы с озером данных или прогнозирования. Каппа – в случаях анализа непосредственно в реальном времени.

Наиболее важной частью архитектуры систем работы с большими данными является путь ускорения. Однако при поступлении данных в реальном времени блок анализа и обработки данных подвергается большой нагрузке, так как обязан провести действия абсолютно над всеми данными. В таком случае появляется слабое место во всей архитектуре критиче-

ского пути. Наиболее общим решением подобной проблемы может стать дополнительный блок выявления аномалий. Выявление аномалий - это общий термин для техник анализа данных с целью выявления редких данных или событий, который можно определить, как выброс данных. Зачастую такие данные не несут в себе важной информации, так как являются изначально некорректными. Измененный вид каппа-архитектуры представлен на рисунке 5.

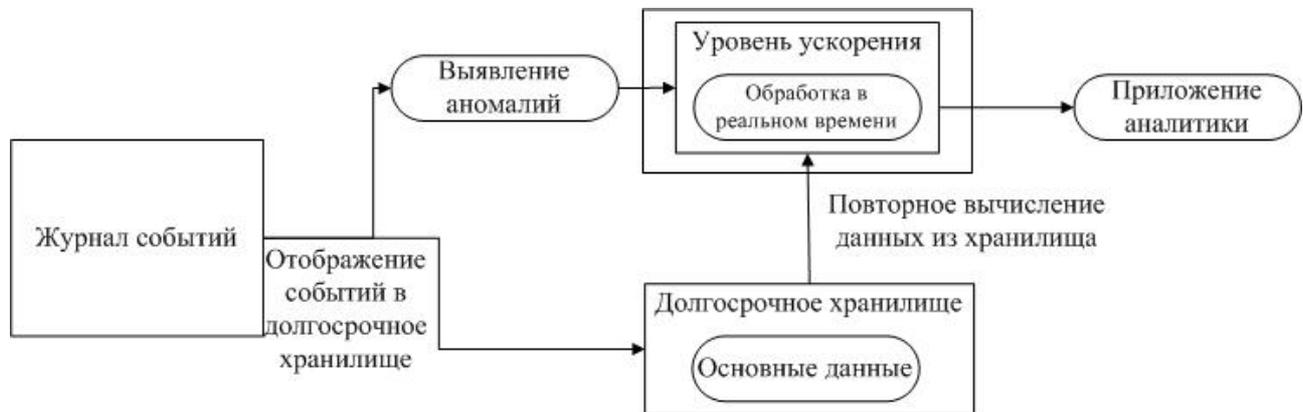


Рис. 5. Измененная каппа-архитектура

Лямбда и каппа архитектуры позволяют решить часть трудностей, которые несут за собой большие объемы данных. Однако они содержат лишь общие характерные черты для построения систем работы с большими данными. Использование блока выявления аномалий позволит оптимизировать работу уровня ускорения: увеличится быстродействие анализа данных и уменьшится погрешность, которая появлялась из-за использования сырых данных, поступающих сразу от источника.

Работа выполнена под научным руководством к.т.н., доцента кафедры ЭВМ Н.Н. Гринченко.

Библиографический список

1. Воронова Л.И. Big Data. Методы и средства анализа : учебное пособие / Воронова Л.И., Воронов В.И. — Москва : Московский технический университет связи и информатики, 2016. — 33 с. — ISBN 2227-8397. — Текст : электронный // Электронно-библиотечная система IPR BOOKS : [сайт]. — URL: <http://www.iprbookshop.ru/61463.html>
2. Рыбальченко М.В. Архитектура информационных систем. Часть 1 : учебное пособие / Рыбальченко М.В. — Таганрог : Издательство Южного федерального университета, 2015. — 92 с. — ISBN 978-5-9275-1765-7. — Текст : электронный // Электронно-библиотечная система IPR BOOKS : [сайт]. — URL: <http://www.iprbookshop.ru/78664.html>
3. Стиль архитектуры для обработки больших данных [Электронный ресурс] // AzureApplicationArchitectureGuide | MicrosoftDocs. URL : <https://docs.microsoft.com/ru-ru/azure/architecture/guide/architecture-styles/big-data>
4. Варианты архитектуры для обработки больших данных [Электронный ресурс] // AzureApplicationArchitectureGuide | MicrosoftDocs. URL : <https://docs.microsoft.com/ru-ru/azure/architecture/data-guide/big-data/>

УДК 519.713; ГРНТИ 27.45

РЕШЕНИЕ СТРОКОВЫХ ЗАДАЧ С ПОМОЩЬЮ СУФФИКСНОГО АВТОМАТА

А.С. Карпова

Рязанский государственный радиотехнический университет имени В.Ф. Уткина,
Российская Федерация, Рязань

Аннотация. В работе рассматривается алгоритм построения суффиксного автомата и примеры решения задач с его помощью.

Ключевые слова: суффиксный автомат, суффиксные ссылки, суффикс.

SOLVING STRING PROBLEMS USING THE SUFFIX AUTOMATON

A. S. Karpova

Ryazan State Radio Engineering University named after V.F. Utkin,
Russia, Ryazan

The summary. The paper considers an algorithm for constructing a suffix automaton and examples of solving problems using it.

Keywords: suffix automaton, suffix links, suffix.

В современном мире большое количество задач при разработке различных систем связано с обработкой строк. Рассмотрим задачу прогнозирования пользовательского ввода, основываясь на предыдущих введенных данных. Примером такой задачи может служить автодополнение при наборе текстовых сообщений в мессенджерах. Для решения подобной задачи можно воспользоваться структурой данных, называемой суффиксным автоматом.

Суффиксным является такой минимальный (количество вершин минимально) детерминированный (для одной строки нет двух различных путей) автомат строки s , который принимает все возможные подстроки строки s и только их.

Рассмотрим некоторые определения, необходимые для понимания алгоритма [1].

Состояние t_0 обозначает начальное, из которого должны быть достижимы остальные состояния $t_1..t_n$.

Каждая дуга автомате представляет собой переход, имеющий метки. Для каждого состояния метка должна быть уникальной.

Одно или более состояний являются терминальными. Метки на дугах, по которым осуществляется переход из начального состояния в терминальное, составляют строку, определяющую суффикс строки s .

Множество позиций, на которых оканчивается вхождение непустой подстроки t строки s составляют множество окончаний, называемое $endpos(t)$.

Подстроки t_1 и t_2 будут представляться $endpos$ -эквивалентными, если совпадут множества их окончаний. Получаем, что множество всех подстрок строки s разбивается на классы эквивалентности, которые соответствуют их множествам $endpos$.

Каждому состоянию v ставится в соответствие одна или несколько строк. С помощью $longest(v)$ будет определяться самая длинная из таких строк, а длина - $len(v)$, через $shortest(v)$ - самая короткая с длиной $minlen(v)$.

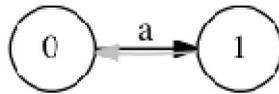
Каждое состояние $v \neq t_0$ имеет суффиксную ссылку, которая ведет в состояние, совпадающее с суффиксом строки $longest(v)$ с длиной $minlen(v) - 1$ [2].

Алгоритм построения суффиксного автомата для строки «abcb»

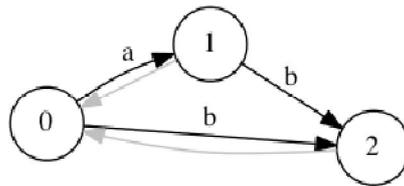
1. На первом этапе суффиксный автомат состоит только из начального состояния, для него $last = 0$, $len(0) = 0$:



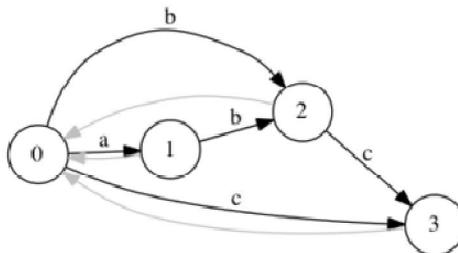
2. Дополняем автомат символом **a**, который представляет собой переход в новое состояние 1. Учитывая, что из начального состояния 0 нет суффиксных ссылок, то она добавляется из состояния 1 в начальное $link1 = 0, last = 1, len(1) = 1$.



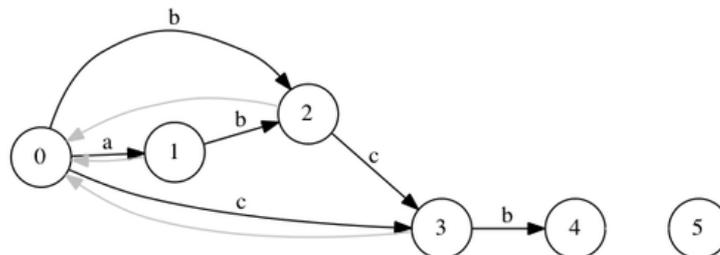
3. Далее укажем символ **b**, обозначающий переход из состояния 1 в 2. По суффиксной ссылке возвращаемся в состояние 0 и дополняем автомат переходом в состояние 2 из 0, указывая $link2 = 0, last = 2, len(2) = 2$.



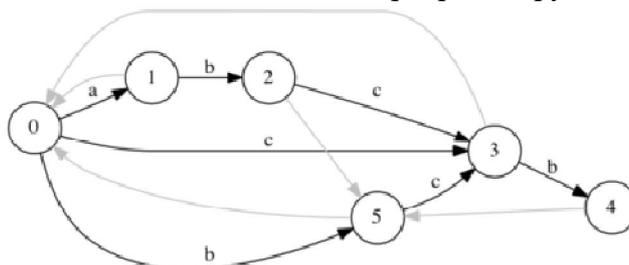
4. Аналогично добавляем символ **c**, для которого переход будет из начального состояния, а суффиксная ссылка обозначаться как $last = 3, len(3) = 3$:



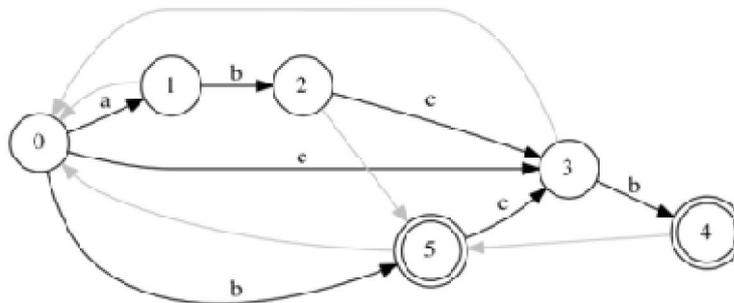
5. Добавляем символ **b**, являющийся переходом из состояния 3 в 4. По суффиксной ссылке возвращаемся в начальное состояние, из которого уже существует переход по символу **b**.



6. Если рассмотреть состояние 2, в которое уже существует переход по символу **b**, то получим, что $len(0) + 1 \neq len(2)$. Поэтому создаем новое состояние 5, в которое из 2 копируем как переходы, так и все суффиксные ссылки, указав $len(5) = len(0) + 1 = 1$. Суффиксную ссылку из состояния 2 направляем в 5 и добавляем еще одну ссылку из 4 состояния в 5, а также переход из начального в состояние 2 переориентируем в 5:



7. Построение автомата выполнено. Для того чтобы отметить терминальные вершины, необходимо найти состояние, которое будет принимать строку «abc**b**». Для этого, передвигаясь по суффиксным ссылкам, необходимо отмечать терминальными все пройденные состояния.



Использование суффиксного автомата при решении задачи автодополнения

Рассмотрим ситуацию, когда пользователь вводит определенный текст сообщения и отправляет его другому пользователю. В таком случае мы можем рассматривать все отправленные сообщения как один большой текст. Построим суффиксный автомат для такого текста.

Так как суффиксный автомат является минимальным детерминированным, осуществив переходы по символам слова w , мы попадем в состояние q , соответствующее *endpos* множества слов w_i из текста t , совпадающих со словом w .

Нетрудно заметить, что слова, сформированные из переходов от состояния q до состояний, не имеющих переходов или состояний, имеющих переход, соответствующий пробельному символу, мы получим все уникальные слова, которые идут после слова w в тексте t .

Таким образом, используя несложную рекурсивную функцию, можно получить набор слов, которые пользователь использовал в комбинации с произвольным словом w . Для ранжирования таких слов по частоте использования можно модифицировать алгоритм построения суффиксного автомата для подсчета количества -эквивалентных слов.

После отправки пользователем нового сообщения нет необходимости строить автомат с нуля, его можно достроить от последнего состояния. Новый автомат будет соответствовать тексту, полученному добавлением нового сообщения в конец текста, по которому был построен предыдущий автомат.

Пример результата построения суффиксного автомата и решения задачи автодополнения представлен на рисунке 1.

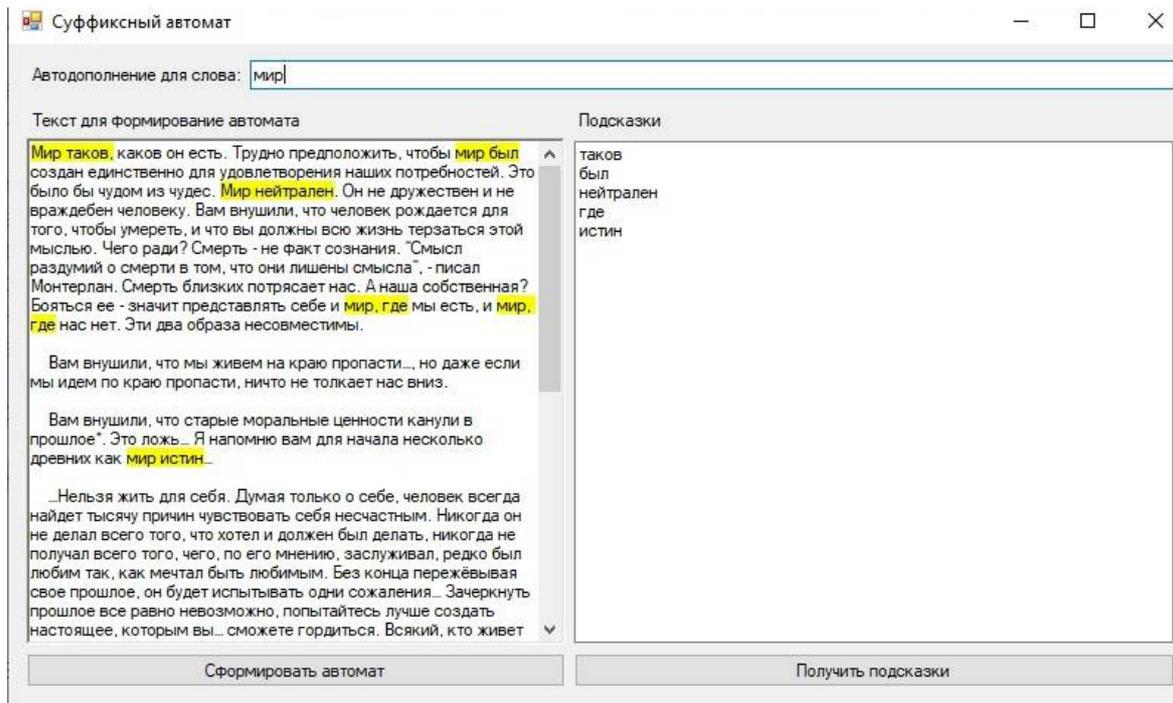


Рис. 1. Результат решения задачи автодополнения

Библиографический список

1. Рубцов А. А. Заметки и задачи о регулярных языках и конечных автоматах - М.: МФТИ, 2019. - 112 с.
2. Паращенко Д. А. Обработка строк на основе суффиксных автоматов - СПб.: ИТМО, 2007. - 35 с.
3. Ахо, Альфред Структура данных и алгоритмы - М.: Издательский дом «Вильямс», 2000. - 384 с.

УДК 004.932; ГРНТИ 28.23.15

РАЗРАБОТКА АЛГОРИТМА ГЕНЕРАЦИИ ЛАНДШАФТНЫХ ИЗОБРАЖЕНИЙ

И.В. Боровков

*Рязанский государственный радиотехнический университет имени В.Ф. Уткина,
Российская Федерация, Рязань, ivanborovkov68@gmail.com*

Аннотация. В данной статье описывается разработка алгоритма генерации ландшафтного изображения на основе исходного. Показан результат работы на примере изображения в оттенках серого. Среда MATLAB наиболее предпочтительна для реализации разработанного алгоритма.

Ключевые слова: ландшафтные изображения, обработка изображений, генератор изображений.

LANDSCAPE IMAGES GENERATION ALGORITHM DEVELOPMENT

I.V. Borovkov

*Ryazan State Radio Engineering University named after V.F. Utkin,
Russia, Ryazan, ivanborovkov68@gmail.com*

The summary. This article describes the landscape images generation algorithm development based on the source image. There is shown the result of the work using the grayscale image example. The MATLAB environment is the most preferable for the implementation of the developed algorithm.

Keywords: landscape images, image processing, image generator.

В наши дни космос активно эксплуатируется различными государственными и частными компаниями для получения актуальных данных о состоянии планеты. Во многих сферах деятельности человека спутниковые снимки дают исчерпывающую информацию об интересующих его процессах. Например, благодаря анализу спутниковых снимков можно наблюдать за активностью таяния ледников, что позволит предсказать дальнейшее поведение природы. В глобальной перспективе, это может помочь человечеству пережить грядущие природные катаклизмы с наименьшими потерями.

Принцип работы космических аппаратов, снимающих поверхность земли, состоит в фиксировании отраженного электромагнитного излучения (излучаемого солнцем) от поверхности земли [1]. Отраженное излучение в виде видимого света, тепла и радиоволн принимаются различными приемными системами искусственных спутников, преобразуются в радио сигнал и передаются на землю. Наземные системы приема спутниковой информации преобразуют эти данные в цифровые изображения, которые в дальнейшем проходят необходимую обработку. Первичные данные хранятся на различных видах носителей и могут передаваться через интернет. Так упрощенно выглядит схема получения данных космической съемки (дистанционного зондирования подстилающей поверхности земли).

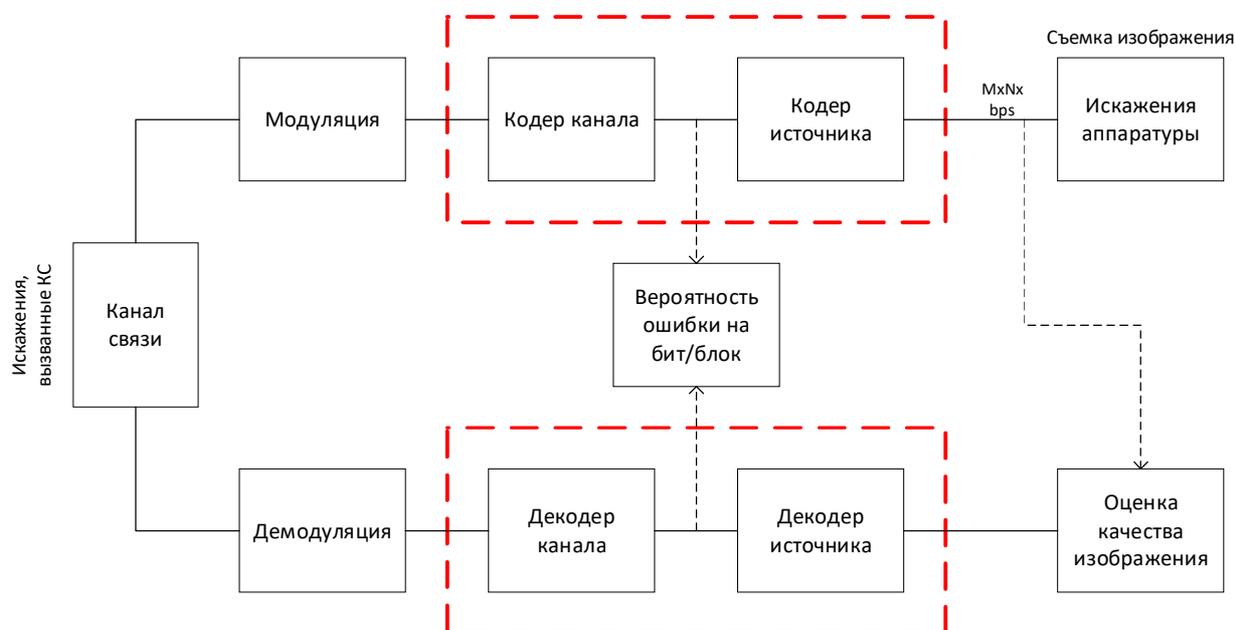


Рис. 1. Система передачи данных

Передача изображений осуществляется от источника информации на её приёмник. В свою очередь, источник и приёмник информации входят в состав одной большой системы передачи данных (СПД) [2]. На каждом этапе прохождения внутри системы данные подвергаются трансформациям и различным внешним воздействиям. На рисунке 1 представлены этапы прохождения изображения по СПД.

Источник данных – первый элемент данной схемы, является объектом исследований. Но необходимо также кратко описать остальные элементы СПД.

С источника данных сигнал поступает на кодер источника и кодер канала связи, где кодируется определенным образом. Кодирование – это представление символов одного алфавита символами другого.

После процесса кодирования сигнал подвергается модуляции. Это процесс изменения параметров переносчика несущей в соответствии с передаваемым сообщением. В случае передачи цифровых данных имеет место применение цифровой модуляции. В этом случае сиг-

нал состоит из дискретных единиц модуляции, которые впоследствии интерпретируются как цифровые данные.

После модуляции сигнал поступает на канал связи. Данный канал образуется на линиях связи, причём внутри одной линии связи можно образовать несколько каналов. Делается это путём временного или частотного разделения каналов между собой. Это позволяет передавать информацию по одной линии связи в нескольких направлениях одновременно, что значительно увеличивает скорость взаимного обмена данными между несколькими устройствами. По своей сути линия связи представляет собой некую физическую среду, по которой передаётся сигнал [3].

При передаче данных по каналам связи могут возникнуть следующие проблемы, влияющие на качество передаваемой по ним информации (в данном случае изображения):

1. Обрывы или физические повреждения канала связи. Такая проблема характерна для проводных и кабельных каналов связи. В данном случае изображение не передаётся вообще или же передаётся со значительными искажениями.

2. Влияние внешнего излучения. Это могут быть электромагнитные или радиоволны, ионизирующие и другие виды излучения. Оно оказывает негативное влияние как на проводные и кабельные, так и на беспроводные каналы связи. В результате в канале связи возникают помехи, которые накладываются на изображения в виде различных шумов.

3. Коллизии. Данная проблема возникает, когда по каналу связи передача происходит одновременно по двум разным направлениям. Виной этому может быть неправильно подобранная топология сети (например, коллизии часто встречаются в топологии «шина»), а также неправильно организованная процедура передачи данных в сети.

4. Несанкционированный доступ извне. При нежелательном вмешательстве информация, передаваемая по каналу связи, также может пострадать и поступить получателю в искажённом виде.

После прохождения по каналу связи, сигнал поступает на приёмник. Перед этим он подвергается демодуляции.

Демодуляция – это процесс, обратный процессу модуляции. Сигнал вновь принимает свой изначальный вид, из чего следует, что демодуляция происходит по правилам, обратным тем, что использовались при модуляции.

Поскольку ранее сигнал был закодирован кодером источником и кодером канала связи, для дальнейшей работы он должен быть декодирован с помощью декодеров. В процессе декодирования восстанавливается исходная кодовая комбинация, передаваемая по каналу связи. Поскольку существует огромное количество вариантов кодирования, необходимо учесть тот факт, что при декодировании должен быть использован специальный метод для аналогичного метода кодирования. Это может осуществляться на аппаратном и/или программном уровне, с помощью использования соответствующих протоколов передачи информации.

С точки зрения математики, изображение – это некоторый объект на плоскости. В зависимости от точки, вид этого объекта меняется. Согласно законам физики, данный объект испускает волны определённой длины, причём от длины волны зависит цветовое восприятие этого объекта. То есть предполагается, что некоторое количество энергии располагается на плоскости. Из-за этого изображение можно представить как некоторую функцию двух переменных x и y .

При этом область допустимых значений этой функции является ограниченной, так как значение цвета в каждой точке представляет собой величину из ограниченного диапазона видимого спектра. В связи с этим, для функции, определяющей изображение, выполняется следующее условие: $0 \leq f(x, y) \leq f_{\max}$.

Здесь (x, y) – координаты на плоскости изображения, $f(x, y)$ – значение яркости в конкретной точке, f_{\max} – некоторая постоянная величина.

Работа с изображениями предполагает проведение некоторых математических операций. Для этого главным условием является то, что изображение представлено в цифровом виде. Невозможно осуществлять операции фильтрации или сжатия над распечатанной фотографией или рисунком на бумаге. При оцифровке изображение представляется на физическом носителе в виде массива чисел, образующих прямоугольную матрицу (рисунок 2). Матрица имеет конечное количество элементов, которые называются пикселями. Значения элементов являются дискретными и входят в некоторое конечное счётное множество.

	1	2	3	...	N
1					
2					
...					
M					

Рис. 2. Представление изображения в виде матрицы $M \times N$

В данный момент для представления значений пикселя используется бинарная последовательность длиной 8 бит. Это значит, что множество возможных значений пикселя ограничивается 256 элементами, а само значение цвета пикселя $P_{ij} \in [0; 255]$.

Разработка алгоритма

Алгоритм генерации ландшафтного изображения представляет собой набор операций, которые необходимо произвести над исходным изображением, чтобы получить новое изображение.

Для реализации алгоритма необходимо:

1. Представить изображение в виде матрицы.
2. Если изображение не в оттенках серого, конвертировать его в оттенки серого.
3. Составить вектор $M \times R$ математических ожиданий по строкам матрицы.
4. Составить вектор $D \times R$ дисперсий по строкам матрицы.
5. Составить вектор $S \times R$ СКО по строкам матрицы.
6. Составить вектор $M \times C$ математических ожиданий по столбцам матрицы.
7. Составить вектор $D \times C$ дисперсий по столбцам матрицы.
8. Составить вектор $S \times C$ СКО по столбцам матрицы.
9. Составить две матрицы того же размера, что и исходное изображение.
10. $MR[I,j] = MxR[i] + SxR[i] * (rand - 0.5) * 2$.
11. $MC[I,j] = MxC[i] + SxC[i] * (rand - 0.5) * 2$.
12. $M[I,j] = 0.5 * (MC[I,j] + MR[I,j])$.

Так может выглядеть одна из реализаций алгоритма. Вариативность может иметь место в последних трёх шагах.

Составим для данного алгоритма схему (рис. 3 и 4).

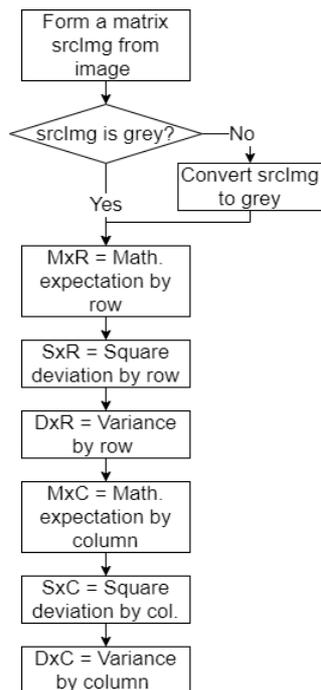


Рис. 3. Подготовительная часть алгоритма

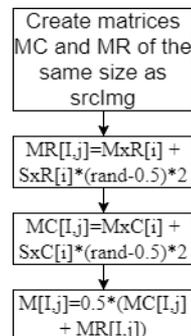


Рисунок 4. Вычислительная часть алгоритма

На рисунке 4 представлен один из вариантов алгоритма. На самом деле, можно к математическим ожиданиям по столбцам подмешивать СКО по строкам (и наоборот), таким образом получая наиболее далёкое от исходного изображение.

Демонстрация результата

Исходное изображение (представлено на рисунке 5) имеет разрешение 512*512 пикселей. Каждый пиксель представляет один из 256 возможных оттенков серого.



Рис. 5. Исходное изображение

Изображение, сгенерированное на основе исходного, представлено на рисунке 4. Как можно заметить, в отличие от исходного, данное изображение имеет большую контрастность, а также имеются следы помех, имитирующие аэрокосмическую съемку.



Рис. 6. Результат работы алгоритма

В рамках данной работы был разработан алгоритм генерации ландшафтных изображений на основе исходных, а также реализована программа, обеспечивающая удобный доступ к этому алгоритму.

Главная функция разрабатываемого программного обеспечения — возможность генерации изображения на основе заданного, была достигнута в результате проведенных исследований. Результатом работы программы стали полученные изображения в оттенках серого, отличающиеся от исходных.

К сожалению, используя только этот алгоритм, невозможно получить адекватное изображение местности, отличающееся от исходного контурами дорог, рек, полей и других объектов. Наиболее подходящим решением этой проблемы может стать нейросеть. В таком случае необходима достаточно обширная обучающая выборка.

Библиографический список

- 1 Костров Б. В. Основы цифровой передачи и кодирования информации: учебное пособие / Б. В. Костров. – Рязань: РГРТУ, 2010. – 196 с.
- 2 Злобин В. К. Спектральный анализ изображений в конечных базисах: монография / В. К. Злобин, Б. В. Костров, А. Г. Свирина. – М.: КУРС, 2016. – 181 с.
- 3 Гринченко Н.Н., Овечкин Г.В. Помехоустойчивое кодирование для цифровых систем связи // Известия ТРТУ. 2006. № 15 (70). С. 5-10.

УДК 004.052.44; ГРНТИ 49.03.03

ЦИКЛИЧЕСКИЕ ПОМЕХОУСТОЙЧИВЫЕ КОДЫ

С.М. Кузяков

Рязанский государственный радиотехнический университет имени В.Ф. Уткина,
Российская Федерация, Рязань, *kuzyakov.sm@gmail.com*

Аннотация. В данной статье приведены основные теоретические сведения о циклических помехоустойчивых кодах.

Ключевые слова: помехоустойчивое кодирование (ECC), линейные блочные коды, циклические коды.

CYCLIC ERROR-CORRECTING CODES

S.M. Kuzyakov

Ryazan State Radio Engineering University named after V.F. Utkin,
Russia, Ryazan, *kuzyakov.sm@gmail.com*

The summary. This article provides basic theoretical information about cyclic error-correcting codes.

Keywords: error-correction code (ECC), linear block codes, cyclic codes.

Большинство современных систем передачи информации включают в себя те или иные технологии помехоустойчивого кодирования для защиты от помех. В силу своей природы эти системы имеют различные технические характеристики, такие как, скорость обмена данными, частота возникающих в канале ошибок и другие. Эти различия порождают соответствующие требования к алгоритмам кодирования. Важной проблемой при проектировании систем связи является выбор оптимального средства защиты от помех.

Процесс помехоустойчивого кодирования заключается в модификации передаваемой информации по некоторому алгоритму, а именно: внесение избыточности, позволяющей исправить или только обнаружить ошибки, возникающие из-за посторонних воздействий в физическом канале передачи данных. Поиск и исправление ошибок основано на том, что не все возможные кодовые комбинации являются допустимыми.

Существует множество различных помехоустойчивых кодов, различающихся по своим характеристикам, такими как: алгоритмы кодирования и декодирования, структурой кода, разбиение символьной последовательности и другие. На основе многих из них могут быть выделены классы помехоустойчивых кодов.

Важным классом являются линейные коды, для них выполняется следующее условие: совокупность кодовых слов образует линейное пространство, то есть сумма любых двух его кодовых слов также является кодовым словом.

Циклическими называются линейные, блочные (n, k) коды, алфавит допустимых комбинаций которых содержит все циклические сдвиги своих кодовых слов.

Циклическим сдвигом кодового слова $v = (v_0, v_1, \dots, v_{n-1})$ называется сдвиг всех его символов на одну позицию вправо и перемещение n -го символа на первую позицию в результате которого получается кодовое слово $v_1 = (v_{n-1}, v_0, v_1, \dots, v_{n-2})$. Таким образом, i -кратный сдвиг задает кодовое слово $v_i = (v_{n-i}, \dots, v_{n-1}, v_0, v_1, \dots, v_{n-i-1})$ [1].

Слова циклического (n, k) кода можно рассматривать как элементы поля Галуа $GF(2^n)$. В таком случае кодовому слову $v = (v_0, v_1, \dots, v_{n-1})$ соответствует следующая полиномиальная запись $v(x) = (v_0x^0, v_1x^1, \dots, v_{n-1}x^{n-1})$. При подобном рассмотрении циклический сдвиг v на одну позицию, то есть преобразование слова $v(x) = (v_0x^0, v_1x^1, \dots, v_{n-1}x^{n-1})$ в $v_1(x) = (v_{n-1}x^{n-1}, v_0x^0, \dots, v_{n-2}x^{n-2})$, можно описать как умножение на некоторый полином $v_1(x) = a(x)v(x)$ [1].

Циклический сдвиг вектора v на i позиций тоже может быть записан в виде полинома, его можно вычислить как остаток от деления многочлена $x^i v(x)$ на $(x^n + 1)$. Это свойство используется в процедуре эффективного обнаружения ошибок.

Циклический (n, k) код строится на основе порождающего полинома $g(x)$ степени $r = n - k$: $g(x) = (g_0x^0, g_1x^1, g_2x^2, \dots, g_rx^r)$.

Необходимым условием является делимость многочлена $(x^n + 1)$ на порождающий полином без остатка. Для выбора подходящего $g(x)$, многочлен $(x^n + 1)$ раскладывается на неприводимые множители: $(x^n + 1) = m_1(x)m_2(x)m_3(x) \dots m_s(x)$.

В качестве $g(x)$ выбирается какой-либо множитель $m_i(x)$ или произведение некоторых из них. Произведение оставшихся формирует многочлен $h(x)$ степени k , такой что $g(x)h(x) = (x^n + 1)$. Он называется проверочным полиномом циклического кода [2].

Есть два способа построения циклического кода: несистематический, при котором кодовое слово не содержит явно информационный вектор, и систематический, при использовании которого те или иные k бит вектора кода содержат исходное сообщение.

Для формирования несистематического кода информационный вектор $u = (u_0, u_1, \dots, u_{k-1})$ представляется в виде многочлена и умножается на порождающий полином $g(x)$.

Систематическое кодирование происходит сложнее. Информационный вектор сдвигается вправо на $n - k$ позиций $x^{n-k}u(x)$. Далее находится остаток от его деления на порождающий полином $g(x)$: $x^{n-k}u(x) = a(x)g(x) + b(x)$.

Кодовое слово находится как сумма полинома $x^{n-k}u(x)$ и остатка от деления $b(x)$. В результате правые k бит кодового слова содержат исходное сообщение, оставшиеся левые биты называются проверочными: $v(x) = x^{n-k}u(x) + b(x)$

Порождающая матрица циклического кода строится на основе его порождающего полинома. Левые символы первой строки матрицы содержат коэффициенты порождающего многочлена $g_0, g_1, g_2, \dots, g_{n-r}$, оставшиеся правые содержат нули [2]. Вторая строка содержит те же коэффициенты, только смещенные на одну позицию вправо и нулевой первый символ. Оставшиеся строки строятся так же путем сдвига порождающего полинома. В общем случае порождающая матрица G циклического кода имеет вид:

$$G_{k \times n} = \begin{pmatrix} g_0 & g_1 & \dots & g_r & 0 & 0 & \dots & 0 & 0 \\ 0 & g_0 & \dots & g_{r-1} & g_r & 0 & \dots & 0 & 0 \\ 0 & 0 & \dots & g_{r-2} & g_{r-1} & g_r & \dots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \dots & 0 & g_0 & g_1 & \dots & g_r & 0 \\ 0 & 0 & \dots & 0 & 0 & g_0 & \dots & g_{r-1} & g_r \end{pmatrix}.$$

Кодирование с использованием данной матрицы порождает несистематический циклический код. Для формирования систематического кода необходимо сохранить информационные биты в правой части кодового слова, для этого порождающая матрица должна содержать единичную матрицу I_k в своей правой части. Этого можно добиться с помощью элементарных преобразований. Порождающая матрица систематического циклического (n, k) кода будет иметь следующий общий вид:

$$G_{k \times n} = \begin{pmatrix} b_{11} & b_{12} & b_{13} & \dots & b_{1r} & 1 & 0 & 0 & \dots & 0 \\ b_{21} & b_{22} & b_{23} & \dots & b_{2r} & 0 & 1 & 0 & \dots & 0 \\ b_{31} & b_{32} & b_{33} & \dots & b_{3r} & 0 & 0 & 1 & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ b_{k1} & b_{k2} & b_{k3} & \dots & b_{kr} & 0 & 0 & 0 & \dots & 1 \end{pmatrix} = (B_{k \times r} I_k).$$

Проверочная матрица циклического кода формируется на основе его проверочного полинома $h(x)$. Она строится по тому же принципу что и порождающая, но с тем отличием,

что коэффициенты в первой строке располагаются в порядке убывания индексов. Таким образом, проверочная матрица H несистематического циклического (n, k) кода будет иметь следующий общий вид:

$$H_{r \times n} = \begin{pmatrix} h_k & h_{k-1} & \cdots & h_0 & 0 & 0 & \cdots & 0 & 0 \\ 0 & h_k & \cdots & h_1 & h_0 & 0 & \cdots & 0 & 0 \\ 0 & 0 & \cdots & h_2 & h_1 & h_0 & \cdots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \cdots & 0 & h_k & h_{k-1} & \cdots & h_0 & 0 \\ 0 & 0 & \cdots & 0 & 0 & h_k & \cdots & h_1 & h_0 \end{pmatrix}.$$

В случае систематического кодирования H строится на основе порождающей матрицы этого кода. Она формируется из единичной матрицы I_r и транспонированной $B_{k \times r}^T$ и в общем случае, будет иметь следующий вид:

$$H_{r \times n} = (IB^T) = \begin{pmatrix} b_{11} & b_{21} & b_{31} & \cdots & b_{k1} & 1 & 0 & 0 & \cdots & 0 \\ b_{12} & b_{22} & b_{32} & \cdots & b_{k2} & 0 & 1 & 0 & \cdots & 0 \\ b_{13} & b_{23} & b_{33} & \cdots & b_{k3} & 0 & 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ b_{1r} & b_{2r} & b_{3r} & \cdots & b_{kr} & 0 & 0 & 0 & \cdots & 1 \end{pmatrix}.$$

Передачу данных через канал можно описать как сумму кодового слова $v(x)$ с некоторым вектором ошибки $e(x)$, в таком случае полученное слово имеет вид $v'(x) = v(x) + e(x)$. Из формул следует, что и при систематическом и при несистематическом кодировании кодовое слово делится без остатка на порождающий полином $g(x)$, тогда переданное сообщение можно описать как $v'(x) = a(x)g(x) + s(x)$, где $s(x)$ — синдром ошибки. Поиск и анализ $s(x)$ лежит в основе обнаружения и исправления ошибок в полученных сообщениях [1].

Обнаружение ошибок основано на кратности допустимых кодовых слов порождающему многочлену. Достаточно найти синдром ошибок полученного сообщения, и если он не равен нулю, то при передаче данных возникли ошибки.

Исправление ошибок основано на факте, что каждый вектор ошибок имеет свой синдром, который не зависит от передаваемого сообщения. Он может быть найден делением вектора ошибки $e(x)$ на порождающий полином кода $g(x)$.

Для исследования эффективности циклических кодов с различными характеристиками была разработана модель системы передачи данных (рис. 1). Пользовательский интерфейс (рис. 2), позволяющий задать параметры моделирования и представляющий его результат, был реализован с помощью интерактивной системы программирования *MATLAB*. Для создания самой модели была выбрана графическая среда имитационного моделирования *Simulink* и средства встроенной в нее библиотеки *CommunicationsToolbox*.

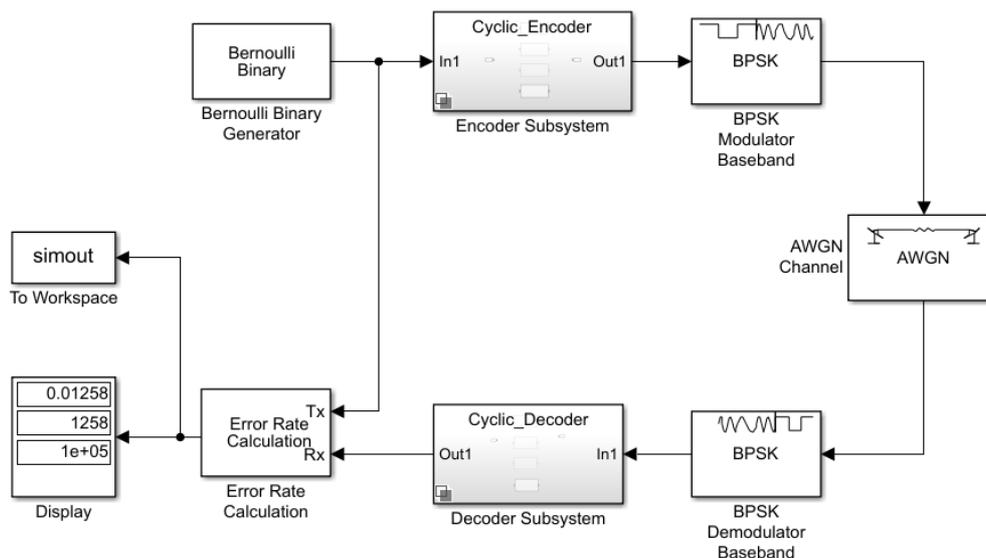


Рис. 1. Модель системы передачи данных

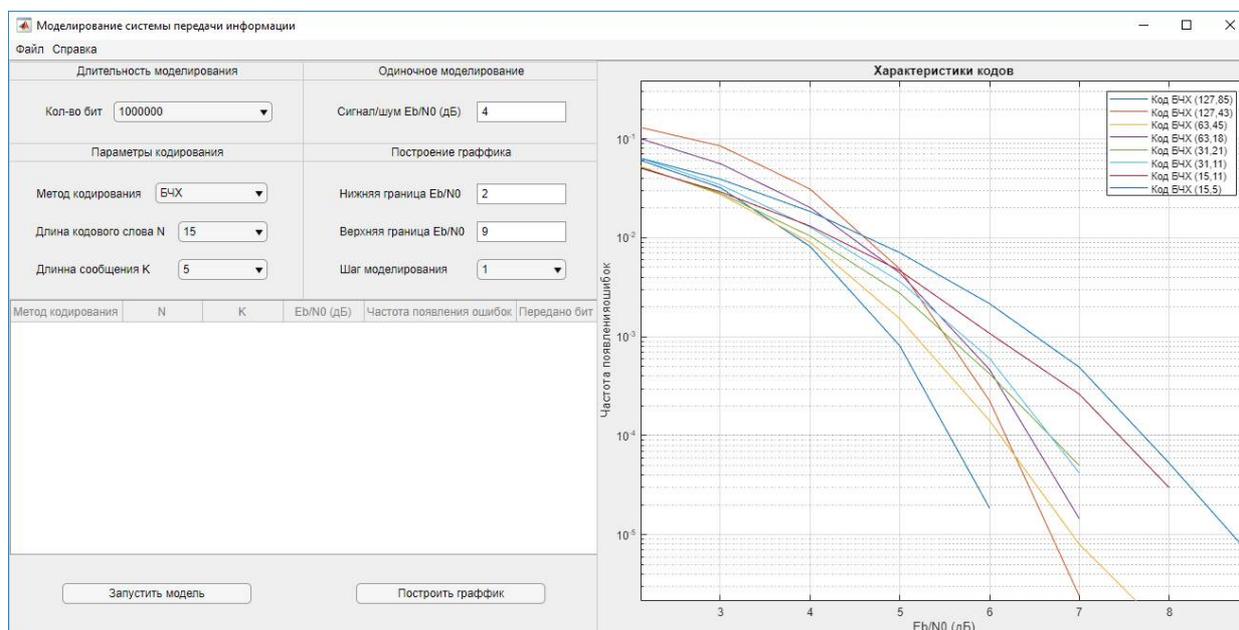


Рис. 2. Пользовательский интерфейс приложения

Различия между характеристиками реальных систем связи не позволяют всегда использовать универсальное средство защиты от помех. В каждом конкретном случае необходим выбор, если не метода кодирования, то, как минимум, его параметров. Это решение напрямую зависит от характеристик системы передачи данных, но в тоже время должно быть принято на этапе проектирования.

Решением данной проблемы может стать моделирование системы передачи информации с необходимыми параметрами и последующее тестирование эффективности различных алгоритмов кодирования для выбора наиболее оптимального из них.

Библиографический список

1. Когновицкий, О. С. Теория помехоустойчивого кодирования. Часть 1. Циклические коды : учеб. пособие / О. С. Когновицкий, В. М. Охорзин ; СПбГУТ. – СПб., 2013. – 94 с.
2. Золотарев В.В., Овечкин Г.В. Помехоустойчивое кодирование. Методы и алгоритмы. Справочник. М.: Горячая линия – Телеком, 2004. – 126 с.

УДК 004.4'242

ТЕХНИЧЕСКИЕ ОСОБЕННОСТИ АВТОМАТИЗАЦИИ ПРОЦЕССОВ ОБЕСПЕЧЕНИЯ КАЧЕСТВА ПРИ НЕПРЕРЫВНОМ РАЗВЕРТЫВАНИИ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ

А.И. Баранчиков, Н.И. Хизриева, Г.Д. Силкин

*Рязанский государственный радиотехнический университет имени В.Ф. Уткина
Российская Федерация, Рязань, grigory.silkin@gmail.com*

Аннотация. В статье представлена краткая информация о непрерывной интеграции, непрерывных поставках и непрерывном тестировании а также показан пример реализации на реальном действующем IT-проекте.

Ключевые слова: качество, тестирование, непрерывное тестирование, разработка ПО

TECHNICAL FEATURES OF PROCESS AUTOMATION QUALITY ASSURANCE FOR CONTINUOUS SOFTWARE DEPLOYMENT

A.I. Baranchikov, N.I. Khizrieva, G.D. Silkin

Annotation. The article provides a brief overview of continuous integration, continuous supply and continuous testing, and shows an example of implementation on a real-world running IT-project.

Keywords: quality, testing, continuous testing, software development

1. От непрерывной интеграции к непрерывному тестированию

Поскольку в настоящее время все больше и больше команд переходят на или уже используют какой-либо вариант методологии Agile, они рано или поздно сталкиваются с необходимостью наладить непрерывные процессы в разработке и обеспечении качества. Обычно все начинается с непрерывной интеграции и постепенно переходит к непрерывной доставке и непрерывному развертыванию.

Непрерывная интеграция

Разработчики, практикующие непрерывную интеграцию, как можно чаще вливают свои изменения в основную ветку. Все эти изменения проверяются путем создания сборки и запуска автоматических тестов для каждого билда. Таким образом вы избегаете проблемы с интеграцией, которые могут возникнуть ближе ко дню релиза.

Непрерывная интеграция требует больших усилий для автоматизации тестирования, чтобы убедиться, что приложение корректно работает, при интеграции новых коммитов в основную ветку.

Непрерывная доставка

Непрерывная доставка расширяет непрерывную интеграцию, поскольку она автоматически развертывает все изменения кода в тестовой или производственной среде после этапа сборки.

Это означает, что помимо автоматизированного тестирования у вас есть автоматизированный процесс выпуска, и вы можете развернуть свое приложение в любое время, нажав кнопку.

Непрерывное развертывание

Непрерывное развертывание идет на шаг дальше, чем непрерывная доставка. Благодаря этой практике каждое изменение, которое проходит все этапы производственного конвейера, передается вашему клиенту. Вмешательство человека отсутствует, и только упавший тест предотвратит внедрение нового изменения в производственную среду.

Непрерывное развертывание - отличный способ ускорить цикл обратной связи с вашими клиентами и снять давление с команды, поскольку больше нет дня релиза. Разработчики могут сосредоточиться на создании программного обеспечения, и они видят, как результат их работы запускается у клиента через несколько минут после отправки кода в основную ветку.

А как насчет измерения качества в этом процессе? Останавливается ли он на автоматических тестах только для сборки или есть какие-то еще тесты и метрики?

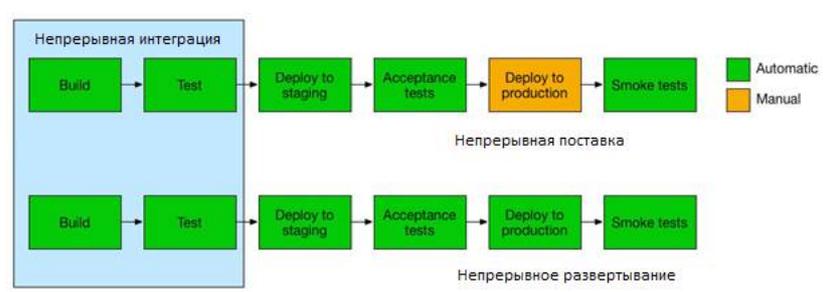


Рис. 1. Непрерывные поставки и Непрерывное развертывание

Непрерывное тестирование

Непрерывное тестирование в DevOps - это тип тестирования программного обеспечения, который включает тестирование программного обеспечения на всех этапах жизненного цикла разработки программного обеспечения. Целью непрерывного тестирования является оценка качества программного обеспечения на каждом этапе процесса непрерывной доставки путем раннего тестирования и частого тестирования.

В процессе непрерывного тестирования DevOps участвуют такие заинтересованные стороны, как разработчик, DevOps, QA. Непрерывное тестирование - это логический следующий шаг в процессе тестирования и автоматизации тестирования в Agile-командах.

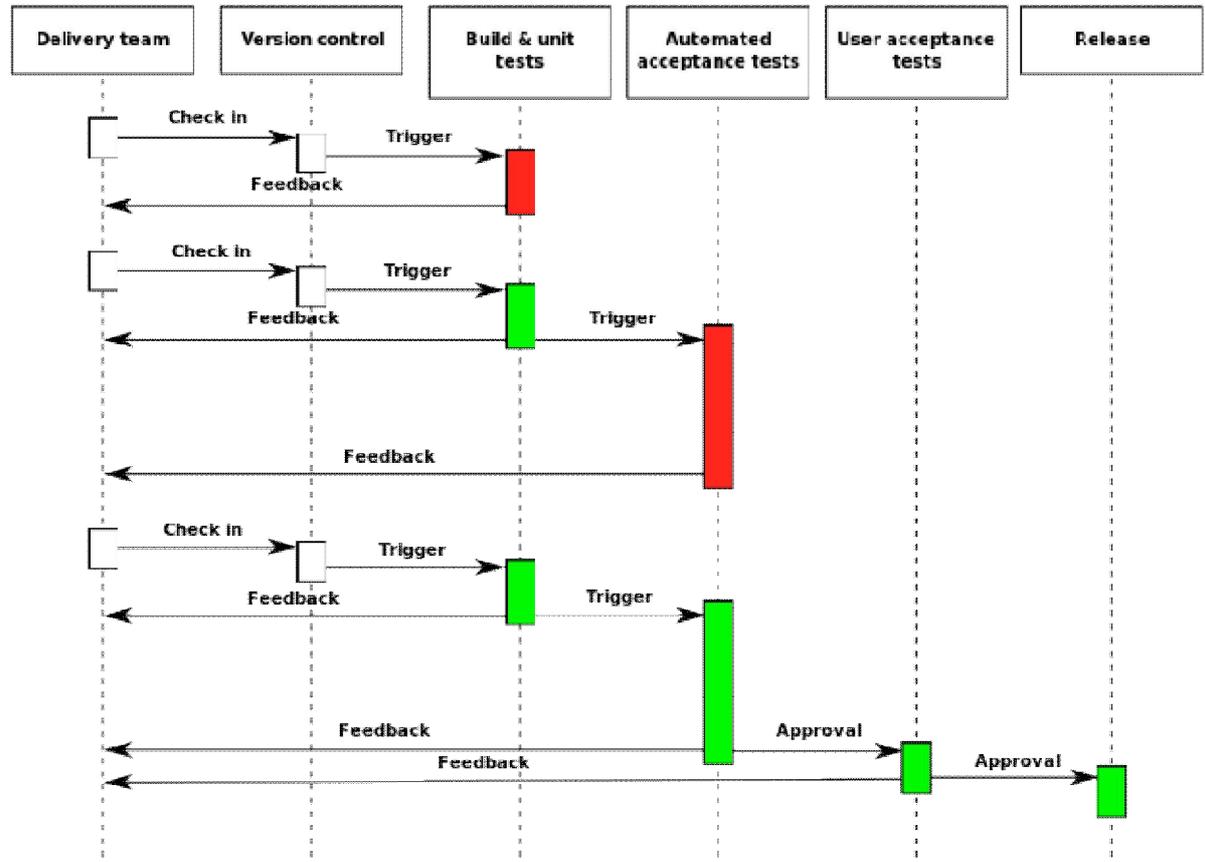


Рис. 2. Quality gates на IT проектах

2. Пример работы процессов автоматизации с реального проекта:
Краткое техническое описание проекта:

- Фронт (веб-приложение) – ReactJS + Java + SpringCloud, Миддл - кредитный конвейер, CRM – Java + SpringCloud + PostgreSQL, Фабрика процессов – Camunda, Интеграционная шина – Mule, Бизнес-хаб -сервис, Криптопровайдер – КриптоПро CSP, Распознавание – ABBYYFlexyCapture, Росреестр Gate – Технопром, СПР – Deductor, МДМ – Справочники, Хранилище документов –DocStore, Аналитический CRM – SAS, Хранилище данных– PostgreSQL, Pentaho

В данном проекте используются следующие QualityGates (врата качества на которых мы убеждаемся, что можем пустить приложение далее по жизненному циклу без вреда для поставки)

На уровне требований происходит как обсуждение “сырых” требований–груминг, так и оценка готовых к реализации требований – Планинг. Запланировано введение определения готовности требований.

На этапе кода имплементировано ревью кода более опытным разработчиком, Unit-тесты автоматизированно проверяющее работоспособность отдельных методов, компиляция решения. Запланирован более строгий подход к ревью покрытия юнит тестами.

На этапе интеграции запускаются APIи UIинтеграционные автотесты.

На этапе обеспечения качества проводится ряд мануальных тестов, исследовательское тестирование, регрессионные и системные тесты.

При выводе решения в эксплуатацию проводится пользовательское тестирование, сбор и обработка отзывов пользователей.



Рис. 3. Пример QualityGatesc реального промышленного ITпроекта

Выводы

При применении практик непрерывного тестирования и покрытия различными активностями, имеющими отношение к обеспечению качества кода и процессов работы команды результат будет стабильно высоким.

Библиографический список:

1. Собирается ли DevOps уничтожить QA? [Электронный ресурс] URL: <https://www.qualityhouse.com/index.php?page=knowledge-hub-striking-the-balance> (дата обращения – 02.02.2020)
2. Оптимизация количества тестов и покрытия [Электронный ресурс] URL: <https://www.youtube.com/watch?v=qHLA0RaVtrI> (дата обращения – 02.06.2020)
3. Что такое тестирование со сдвигом влево [Электронный ресурс] URL: <https://www.youtube.com/watch?v=nOLsAoD9E5M> (дата обращения – 02.06.2020)
4. Как эффективно управлять тестовыми данными используя CA Test Data Manager [Электронный ресурс] URL: <https://www.youtube.com/watch?v=vPzjC1bE1HI> (дата обращения – 02.07.2020)

5. Сдвиг вправо, влево, вниз и вверх. [Электронный ресурс] URL: <https://blazemeter.wistia.com/medias/eyg8yevriy?embedType=iframe&videoFoam=true&videoWidth=640> (дата обращения – 03.05.2020)
6. Тестируем на проде: Canary Deployment
[Электронный ресурс] URL: <https://habr.com/ru/company/oleg-bunin/blog/493026/> (дата обращения – 29.03.2021)
7. Boost your testing success with an online Continuous Testing Maturity Assessment
[Электронный ресурс]: IT компания Tricentis / URL: <https://www.tricentis.com/blog/boost-testing-success-continuous-testing-maturity-assessment> (дата обращения – 29.03.2021)
8. Баранчиков А.И. Синтез информационных структур хранения данных на основе анализа предметных областей. – Рязань: ГУП РО Рязанская областная типография. – 2014. – 231 с.

СЕКЦИЯ «ИНФОРМАЦИОННЫЕ СИСТЕМЫ И ТЕХНОЛОГИИ»

УДК 004.8; ГРНТИ 28.23

СВОЙСТВА ПРЕДСТАВЛЕНИЯ ИЕРАРХИЧЕСКИХ ДРЕВОВИДНЫХ ПОСЛЕДОВАТЕЛЬНОСТЕЙ

Е.В. Селиванов

ПАО «Сбербанк»,

Россия, Рязань, selivanov.ev.v@gmail.com

Аннотация. В статье рассматривается представление иерархических древовидных последовательностей. Обсуждаются идеи современных исследователей о работе с иерархическими структурами в области предиктивной аналитики. Предлагаются свойства представления иерархических древовидных последовательностей, которые позволяют получить ряд преимуществ перед современными моделями машинного обучения. Полученное представление является хорошей основой для построения интеллектуальных систем нового уровня, моделирующих работу неокортекса для получения предсказаний. Модели, обладающие обозначенными в статье свойствами, могут способствовать быстрому развитию исследований в сфере общего искусственного интеллекта.

Ключевые слова: иерархическое представление, последовательности, предсказательная аналитика, общий искусственный интеллект, моделирование, графы, данные.

PROPERTIES OF HIERARCHICAL TREE SEQUENCES REPRESENTATION

E.V. Selivanov

PJSC Sberbank,

Russia, Ryazan, selivanov.ev.v@gmail.com

The summary. The article discusses the hierarchical tree sequences representation. The ideas of modern researchers about working with hierarchical structures in the field of predictive analytics are discussed. Proposed properties of hierarchical tree sequences representation, which provide a number of advantages over modern machine learning models. The resulting representation is a good basis for constructing intelligent systems of a new level that simulate the work of the neocortex to obtain predictions. Models with the properties indicated in the article can contribute to the rapid development of research in the field of general artificial intelligence.

Keywords: hierarchical representation, sequences, predictive analytics, artificial general intelligence, modeling, graphs, data.

Представление иерархических древовидных последовательностей (ИДП), предложенное в статье «Новые методы в предсказательной аналитике» [1] в теории может помочь избежать многих проблем, свойственных современным моделям в области искусственного интеллекта, таких как: проклятие размерности, потеря контекста, малый горизонт предсказания и др. [2]. Это новое представление универсально для любого типа входного сигнала – в конечном итоге всё преобразуется во внутреннюю структуру этого представления – паттерны и связи. Однако, одним из серьёзных минусов рассматриваемого ИДП является сложность разработки алгоритмов для работы с ним. Для того, чтобы подойти к теме разработки алгоритма и перевести перечисленные теоретические преимущества ИДП в практическую область необходимо обогатить рассматриваемое представление нужными свойствами.

Сначала рассмотрим основные идеи структур данных и алгоритмов, предлагаемых исследователями в наши дни. Сейчас классические модели машинного обучения, такие как нейросети, вышли на своё «плато производительности». Они эффективно применяются для решения актуальных специализированных задач [3]. Передний край исследований в области предсказательной аналитики и искусственного интеллекта обращён к идеям унификации и расширению круга решаемых одной моделью проблем. Работа с иерархическими структурами и моделирование неокортекса человеческого мозга это новый шаг развития моделей машинного обучения и достаточно молодое по научным меркам направление. В нём можно

увидеть переход от модели нейрона к моделированию более высокоуровневых структур, таких как кортикальные колонки и др. Исследователи продолжают поиск наиболее удачных иерархических структур для построения интеллектуальных систем нового уровня.

Футуролог Рэймонд Курцвейл верит в большую эффективность алгоритмов, работающих с иерархическими последовательностями для построения общего искусственного интеллекта или AGI (от англ. ArtificialGeneralIntelligence). В качестве модели он предлагает использовать иерархические цепи Маркова [4].

В своей научно-популярной книге «Об интеллекте» [5] Джефф Хокинс публикует идею алгоритма «память-предсказание» для разработанной им модели НТМ (англ. HierarchicalTemporalMemory), в переводе – иерархическая временная память [6]. НТМ представляет из себя сложную модель связанных кортикальных колонок человеческого мозга. Модель требует выделения строго заданного количества своих элементов в каждом слое, определённого заранее количества слоёв и чёткого их иерархического разделения. В рамках проекта «Numenta» исследователями для модели НТМ был разработан алгоритм память-предсказание на языках python и java. В научной литературе достаточно трудно найти сравнения модели НТМ и алгоритма «память-предсказание» с классическими моделями искусственного интеллекта, такими как нейросети. Отчасти, это может быть следствием большой разницы в подходах к предсказанию. Опубликованные практические реализации программ на этой основе демонстрируют способность к предсказанию (с некоторой погрешностью) следующих элементов – простых последовательностей, таких как синусоида и т.п.

Идеи Джеффа Хокинса и Рэймонда Курцвейла схожи в основном аспекте. Это порождение в иерархических моделях восходящих и нисходящих потоков информации. Движение от частного к общему здесь представляет собой входной сигнал. Движение от общего к частному – ответ на входной сигнал, реакция или предсказание.

Рассмотрим, как вышеизложенные идеи можно применить к представлению ИДП и обогатить его новыми свойствами для реализации эффективных предсказывающих алгоритмов (рисунок 1).

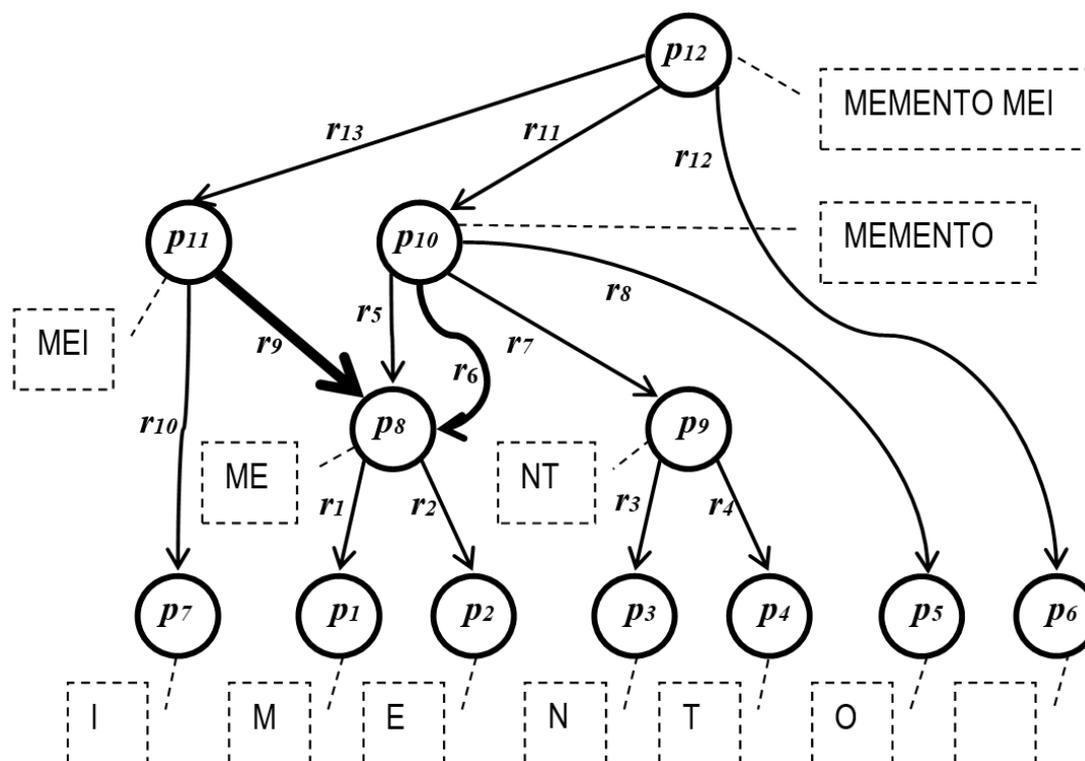


Рис. 1. ИДП с новыми свойствами

Несмотря на то, что паттерны $\{p_1, \dots, p_{12}\} \in P$ на рисунке 1 располагаются по уровням (буквы/слоги/слова/предложения), ИДП должно обеспечивать понимание, какой паттерн является родителем, а какой – потомком, независимо от выбранного способа отображения информации. Это свойство лежит в основе идеи о восходящих и нисходящих потоках в иерархических моделях. Таким образом, мы добавили к связи между паттернами свойство «направление», чтобы она являлась ориентированным ребром между вершинами графа.

Более того, в ИДП нет чёткого понятия «слой». Существует лишь понятие «уровень» для каждого конкретного паттерна, определяющееся входящими в него потомками. Такое решение было принято для того, чтобы избавиться от необходимости выделять под модель определённое количество ресурсов (слои, паттерны, связи) заранее, так как это делается в классических нейронных сетях и модели НТМ. Таким образом, алгоритм, работающий с ИДП, не будет ограничен выделенными ресурсами, а лишь ресурсами выбранной системы хранения информации – конкретной базы данных либо дисковым пространством. При этом создавать новые паттерны и связи можно лишь тогда, когда в этом будет реальная необходимость. Это свойство обеспечивает представлению ИДП отличную масштабируемость, а также экономичность в ресурсах. Задействовав современные облачные технологии, сервисы распределённого хранения и обработки данных [7], можно получить практически неограниченные ресурсы для ИДП.

Рассмотрим пример, используя ИДП, представленное на рисунке 1. Здесь последовательность индексов x связей $r_x \in R$ соответствует последовательности дочерних паттернов. Связи ориентированы в направлении от предка к потомкам.

Если входным сигналом является символ **M**, то, применяя идею о восходящем потоке информации, мы должны взять связанный с ним паттерн p_1 и, используя направленную связь r_1 , найти его предка $p_8 = \langle p_1, p_2 \rangle$. Теперь, находясь в узле p_8 , согласно идее о нисходящем потоке информации и используя свойство упорядоченности связей в ИДП [1], мы можем найти ребро со следующим порядковым номером - r_2 . Перейдя по нему «вниз», мы находим следующего потомка – паттерн p_2 . Таким образом мы получаем прогноз следующего элемента последовательности – символ **E**.

Однако, далее возникает вопрос – что делать, если количество предков паттерна более одного? Например, у паттерна p_8 два предка – $p_{10} = \langle p_8, p_8, p_9, p_5 \rangle$ и $p_{11} = \langle p_8, p_7 \rangle$, причём в p_{10} паттерн p_8 используется дважды. Таким образом, при восходящем потоке информации из узла p_8 нам нужно сделать выбор из трёх связей – r_5 , r_6 и r_9 . От этого зависит точность прогноза. Для этого введём новое свойство связей в ИДП – вес связи. Схематично вес связи показан на рисунке 1 с помощью толщины рёбер графа. В общем случае вес связи должен прямо коррелировать со статистикой частоты использования последовательностей для исследуемой области. Такое решение позволяет увеличить вероятность верного предположения о родительском паттерне в восходящем потоке, и соответственно, увеличить вероятность правильного прогноза в нисходящем потоке информации.

Вероятность правильного прогнозирования увеличивается прямо пропорционально количеству полученных входных сигналов – чем больше элементов последовательности мы получим, тем точнее предскажем следующие её элементы. Таким образом контекст помогает улучшать предиктивные качества системы, основанной на ИДП. При этом, в отличие от существующих и получивших большое распространение моделей машинного обучения [8], массивный контекст в ИДП не подвержен «проклятию размерности» из-за специфики структуры хранения информации в этом представлении [1]. Горизонт прогнозирования можно регулировать «высотой» восходящего потока информации. Чем выше уровень паттерна – тем дальше простирается наше предсказание о следующих элементах последовательности, но при этом может уменьшаться его точность. На практике необходимо найти баланс между дальностью предсказания и его точностью.

Здесь нельзя не упомянуть ещё одно не совсем очевидное свойство ИДП. При реализации восходящего потока информации, в тот момент, когда мы выдвигаем гипотезу о предке паттерна, мы не только получаем прогноз будущего, но ещё и одновременно предполагаем прошлое. Фактически ИДП позволяет сделать предположение о предшествующих элементах последовательности, даже если они физически не поступали в систему. Например, при входящем элементе p_6 «поднимаемся» по ребру r_{12} . Находим $p_{12} = \langle p_{10}, p_6, p_{11} \rangle$ и прогнозируем далее паттерн $p_{11} = \text{МЕИ}$, а вместе с этим предполагаем, что до поступившего на вход элемента была последовательность $p_{10} = \text{МЕМЕНТО}$. Таким образом контекст дополнительно «виртуально» расширяется. Это свойство ИДП можно использовать в алгоритмах глубокого анализа для уточнения прогноза либо для восстановления утерянных элементов последовательностей.

В большинстве современных моделей остро стоит проблема потери контекста. Это, например, выражается в несвязности генерируемых текстов или нити диалога, когда последующие реплики не зависят от предыдущих. Проблемы понимания и синтеза текстов на данный момент часто сводятся к задаче классификации и применения шаблонизированных фраз [9]. Очевидно, что задачи обработки естественного языка требуют применения новых и современных подходов.

Оценивая сложность озвученных выше проблем, попытаемся позаимствовать решение у природы. Добавим в представление ИДП два свойства для всех узлов $p_x \in P$: величину активации A_{xv} и время активации A_{xt} . Таким образом, паттерны в ИДП могут быть активированы (положительная A_{xv}) или деактивированы (отрицательная A_{xv}). При этом A_{xv} функционально зависит от текущего времени T :

$$A_{xv} \rightarrow 0 \text{ при больших значениях } \Delta T = T - A_{xt}.$$

Используя эти свойства совместно с весами связей, в ИДП алгоритм прогнозирования может делать более точные предсказания, основываясь на контексте предыдущих последовательностей. Такое поведение системы становится возможным, если в процессе обработки входных сигналов активировать (при подтверждении прогноза) и деактивировать (при ошибке) соответствующие паттерны в ИДП. Этот процесс схож с человеческим мышлением, когда мы, поговорив на одну тему, некоторое время более склонны интерпретировать следующие слова в контексте той же тематики. В ИДП «рисунок активации» паттернов также угасает со временем, как было показано в формуле выше. Деактивация в свою очередь помогает исключить ошибочные гипотезы. Такая поддержка важна, так как в реальных предметных областях ИДП гораздо сложнее представленного на рисунке 1. В нём может быть достаточно большое количество сильно связанных паттернов разного уровня.

В заключении отметим, что представленные в этой статье свойства ИДП дают ему объективные преимущества перед современными моделями машинного обучения из области предсказательной аналитики. Идеи исследователей и футурологов получили развитие и послужили отправной точкой для создания ИДП. Это представление в текущем его виде является хорошей основой для разработки алгоритмов машинного обучения нового качества. Оно обладает необходимыми свойствами для построения моделей в сфере общего искусственного интеллекта. ИДП является ярким представителем нового направления исследований в области моделирования иерархических структур неокортекса.

Библиографический список

1. Селиванов Е.В. Новые методы в предсказательной аналитике. // Современные технологии в науке и образовании – СТНО-2020 [текст]: сб. тр. III междунар. науч.-техн. форума: в 10 т. Т.5./ под общ. ред. О.В. Миловзорова. – Рязань: ИП Коняхин А.В. (BookJet), 2020. – С. 152-155.
2. Селиванов Е.В., Стружанцев А.И. Технологии искусственного интеллекта // Центральный научный вестник. – 2018. – № 6. – С. 44-47.

3. Шитова К.Г. Разработка интеллектуальной рекомендательной системы подбора кредитных продуктов для предприятий на основе нейронных сетей // Современные технологии в науке и образовании – СТНО-2020 [текст]: сб. тр. III междунар. науч.-техн. форума: в 10 т. Т.5./ под общ. ред. О.В. Миловзорова. – Рязань: ИП Коняхин А.В. (BookJet), 2020. – С. 112-115.
4. Kurzweil, R. How to Create a Mind: The Secret of Human Thought Revealed. –New York: Viking Books, 2012.–336 p.–ISBN 0670025291
5. Hawkins, J., Blakeslee, S. On Intelligence. –New York: Times Books, 2004. – 272 p. – ISBN 0805074562.
6. Хокинс Дж., Дайлип Дж. Временная Иерархическая Память. Концепции, Теория и Терминология. NumentaInc., 2006 [Электронный ресурс] URL: <https://pdfs.semanticscholar.org/72f5/4a24137796bb9d57f6bb64b1d37f4bb14c34.pdf>
7. Селиванов Е.В., Каширин И.Ю. Облачные технологии как новая ступень эволюции информационных сервисов глобальных сетей // Вестник Рязанского государственного радиотехнического университета. – 2014. – № 1. – С. 97-103.
8. Селиванов Е.В. Проблематика нейронных сетей [Текст] / Е.В. Селиванов // Новое слово в науке: стратегии развития: материалы III Междунар. науч.–практ. конф. (Чебоксары, 31 дек. 2017 г.) / редкол.: О.Н. Широков [и др.] – Чебоксары: ЦИНС «Интерактив плюс», 2017. – С. 147-149.
9. Лащилин, Селиванов Е.В. Основные модели машинного обучения в финансово-банковской сфере // Новые информационные технологии в научных исследованиях и в образовании: материалы XXV Юбилейной Всероссийской научно-технической конференции студентов, молодых ученых и специалистов. – Рязань: ИП Коняхин А.В. (BookJet), 2020. – С. 152-154.

УДК 004.9; ГРНТИ 20.01

РАЗРАБОТКА ВИРТУАЛЬНОГО ТУРА ПО ДОМУ-МУЗЕЮ СЕМЬИ АСЕЕВЫХ В ТАМБОВЕ

В.А. Немтинов*, А.А. Горелов*, А.Б. Борисенко*, Ю.В. Немтинова*, С.В. Трюфилкин*, В.В. Морозов***

**Тамбовский государственный технический университет,*

Россия, Тамбов, nemtinov@mail.gaps.tstu.ru,

***Тамбовский государственный университет им. Г.Р. Державина,*

Россия, Тамбов, jnemtinova@hotmail.com

Аннотация. Рассмотрены вопросы разработки виртуального тура по дому-музею семьи Асеевых в Тамбове с использованием программы 3D Vista Virtual Tour. При фото- и видеосъемке были использованы сферические камеры: Insta 360 pro и Instaonex и квадрокоптер типа DJI Mavic Pro 2.

Ключевые слова: виртуальный тур, программа 3D Vista Virtual Tour, камеры: Insta 360 pro и Instaonex .

DEVELOPMENT OF A VIRTUAL TOUR OF THE HOUSE-MUSEUM OF THE ASEYEV FAMILY IN TAMBOV

V. A. Nemtinov*, A. A. Gorelov*, A. B. Borisenko*, Yu. V. Nemtinova, S. V. Trufilkin*, V. V. Morozov***

Tambov state technical University,

Russia, Tambov, nemtinov@mail.gaps.tstu.ru,

*** Tambov state University. G. R. Derzhavina,*

Russia, Tambov, jnemtinova@hotmail.com

Abstract. An The issues of developing a virtual tour of the house-museum of the Aseyev family in Tambov using the 3D Vista Virtual Tour program are considered. When taking photos and videos, spherical cameras were used: Insta 360 pro and Instaonex and a quadrocopter of the DJI Mavic Pro 2 type.

Keywords: virtual tour, 3D Vista Virtual Tour program, cameras: Insta 360 pro and Instaonex.

Широко известный в России в предреволюционные годы Тамбовский фабрикант Михаил Васильевич Асеев был самым богатым в губернии. Комвольно-суконные фабрики, принадлежавшие ему, снабжали сукном многие торговые фирмы и швейные производства в различных регионах страны. Достаточно вспомнить о том, что в разгар Первой мировой войны половина всего личного состава царской армии, а это почти 5 миллионов солдат и «нижних

чинов». Весь офицерский корпус был одет в шинели, пошитые из сукна, изготовленного на фабриках М.В. Асеева. Огромное состояние Асеева позволило ему заниматься не только суконным производством. Его капиталы, пусть не в таком количестве, как от производства сукна, пополнялись еще и от участия в торговых операциях, и от строительства жилых домов.

Часть из тех жилых домов, несмотря на то, что в советский период не входили в число, которые хотели сохранить и продолжить их век, еще можно обнаружить на улицах Тамбова.

Два объекта, некогда принадлежавшие самому фабриканту, сохранились и являются образцами архитектурного творчества в Тамбовском крае.

Первый из них – дворец фабриканта Асеева, расположенный на улице Набережной в южной части города Тамбова. Дворец, построенный в 1905 году и парк, окружающий его, несомненно являются самым известным объектом культурно-исторического наследия во всей Тамбовской области. Более того, сегодня это не только самый главный экскурсоводческий объект в туристическом направлении, он является одним из наиболее значимых мест областного центра для проведения культурно-массовых мероприятий.

С Асеевским дворцом в Тамбове связана не только историческая сторона, с ним связаны и события, происходящие в городе в наше время. Последние имели широкое общественное значение семь лет назад. Тогда, после реконструкции дворца, он в обновленном варианте был выставлен на торги и едва не стал очередным заманчивым местом для размещения заведений увеселительного направления. Происходящие тогда массовые выступления городской общественности, получившие широкую огласку на федеральном телевидении и других СМИ, смогли отстоять отреставрированный объект. Сейчас часть здания является филиалом музея «Петергоф», а другая – музеем жизни и быта семьи фабриканта Асеева.

Можно много рассказывать об этом историческом объекте, связанном с именем одного из известных людей Тамбовщины, но важно то, что и Асеевский дворец за свою 115-летнюю историю, многое утратил в своем первоначальном варианте. В частности, на месте бывшего дома-флигеля художника Шевченко теперь находится фонтан. Это не единственное изменение, произошедшее на территории бывшего дворца фабриканта Асеева. Все это послужило причиной, по которой мы ввели рассматриваемый объект в перечень исследовательских работ авторов с последующим созданием виртуального музея Асеевского дворца и прилегающей территории, отражающий его состояние в последние предреволюционные годы.

Для проведения информационного анализа изменения отдельных параметров, характеризующих объекты культурно-исторического наследия, входящие в исторический комплекс (памятное место), по которым можно судить о динамике развития различных сфер жизнедеятельности исследуемой территории, авторами предложена методика. Ее использование позволяет получить усредненную информацию о количественном значении отдельного параметра, отнесенного к определенному временному периоду развития территории памятного места [1 - 5].

При разработке виртуального тура авторы использовали программу 3D Vista Virtual Tour. При фото- и видеосъемке были использованы сферические камеры: Insta 360 pro и Instaonex. Для создания снимков используются и фотографии высокой чёткости сделанные с помощью квадрокоптера типа DJI Mavic Pro 2. Создав проект в приложении 3dVista, осуществляется выгрузка файлов на собственный ftp-сервер. Для оффлайн доступа, не требующего подключения к интернету, есть возможность выгрузить файлы из проекта 3dVista в форме отдельного плеера. Это один файл, содержащий всё необходимое для запуска виртуального тура на любом персональном компьютере. Таким образом, созданный тур можно просматривать автономно и в режиме онлайн без установки специального программного обеспечения или плагинов на любом устройстве.

Виртуальный тур, содержащий различные медиа и инструменты, позволяют увидеть место на экране максимально реалистично. Технически виртуальный тур представляет собой преобразование плоского панорамного изображения в сферическое. Панорама – «плоская» фотография с полем обзора до 360°. Они создаются путем сшивания нескольких фотографий, например, с помощью 3DVistaStitcher, который входит в 3D Vista Virtual Tour Suite.

В настоящее время виртуальный тур по дом-музею семьи Асеевых в Тамбове включает в себя более 40 сферических панорам. На рисунке 1 представлен фрагмент сферической панорамы общего вида музея, включая вход в него.

При знакомстве с экспонатами музея посетителю предоставляется возможность ознакомиться с текстовым описанием экспонатов, а также в виде звукового сопровождения, поясняющего представленные видовые изображения.

На рисунках 2 - 4. представлены фрагменты сферических панорам зеленого зала и голубой комнаты музея.

Результаты данной работы рассчитаны на использование широким кругом специалистов, использующих методологию применения методов анализа исторических процессов и 3D визуализации объектов и предметов культурного наследия. В частности, результаты проекта использованы в Управлении образования по Тамбовской области в сфере подготовки кадров исторического и информационного направлений и формирования у них культуры исследований в области истории Тамбовского региона с использованием информационной технологии и математического и моделирования.



Рис. 1. Фрагмент сферической панорамы с высоты птичьего полета территории дома-музея семьи Асеевых в Тамбове

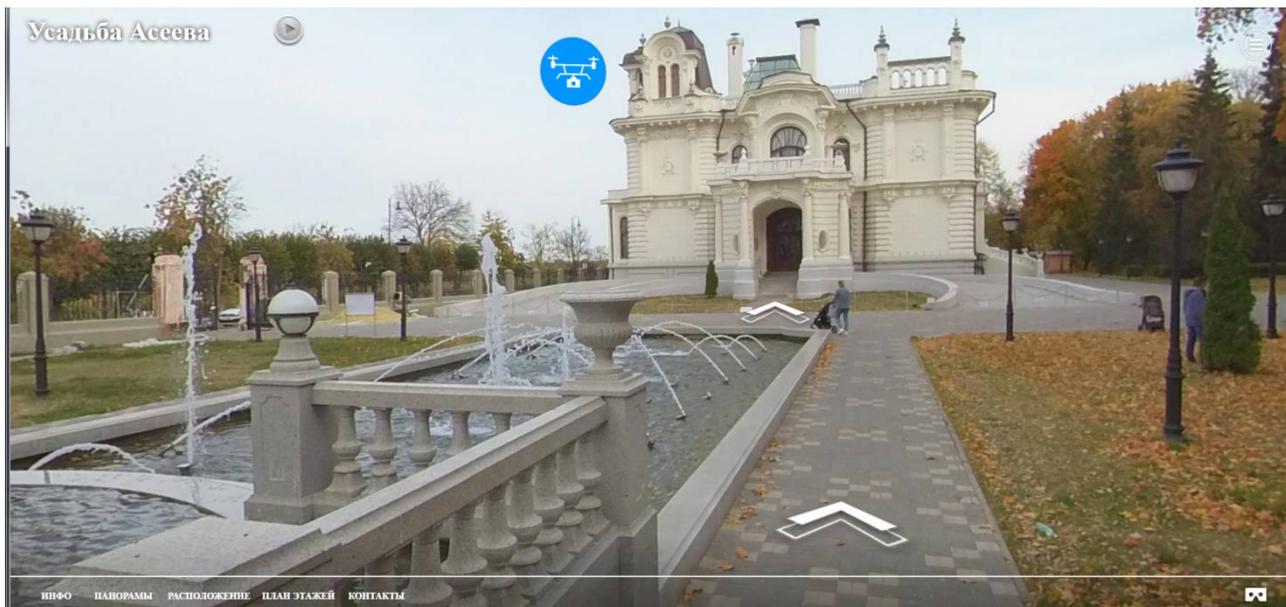


Рис. 2. Фрагмент сферической панорамы общего вида дома-музея семьи Асеевых в Тамбове

Сферой использования результатов проекта являются также школьные образовательные учреждения и музейные организации. Практические результаты рассчитаны и на широкий круг пользователей сети Интернет, интересующихся историей развития Тамбовского региона и Российской Федерации в целом.

Исследование выполнено за счет гранта Российского научного фонда (проект №19-18-00044 «Анализ исторических процессов памятных мест Тамбовского края, связанных с пребыванием в них известных деятелей России, с использованием технологии виртуального моделирования»).

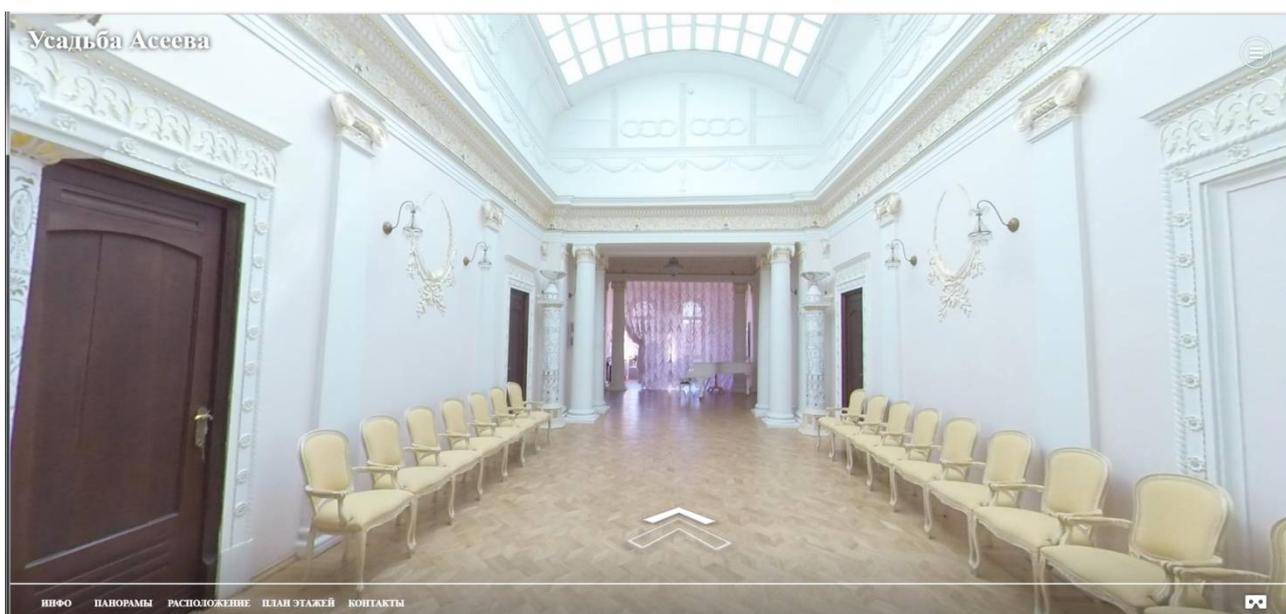


Рис. 3. Фрагмент сферической панорамы колонного зала музея



Рис. 4. Фрагмент сферической панорамы парадной столовой дома-музея

Библиографический список

1. Nemtinov V A, Gorelov A A, Nemtinova Y V and Borisenko A B Visualization of a virtual space and time model of an urban development territory - *Scientific Visualization* 2016, v. 8 (1) – pp. 120–132.
2. Nemtinov V A, Gorelov A A, Nemtinova Y V and Borisenko A B Implementation of technology for creating virtual spatialtemporal models of urban development history - *Scientific Visualization* 2018, v. 10 (3) – pp. 99–107.
3. Горелов А А, Немтинов В А Информационный анализ памятных мест, связанных с пребыванием и деятельностью В.Ф. Войно-Ясенецкого (святителя Луки) на Тамбовской земле - *Вопросы истории*. 2020, № 3. - С. 79-93.
4. Горелов И.А., Немтинов В.А. Информационный анализ динамики освоения исторически значимой территории "Базарная площадь" г. Тамбова. - *Вопросы истории*. 2019, № 11. - С. 33-44.
5. Горелов А А, Немтинов В А Формирование исторического портрета города с использованием информационных технологий. - *Вопросы истории*. 2018, № 7. - С. 106-111.

УДК – I21L 63

ВЛИЯНИЕ СОВРЕМЕННЫХ ТЕХНОЛОГИЙ НА КАЧЕСТВО ОБРАЗОВАНИЯ

А.А. Кабулов

*Ташкентский филиал Российского экономического университета им. Г.В. Плеханова
Узбекистан, Ташкент, kabulov51@mail.ru*

Аннотация. Качество образования сегодня во всем мире имеет первостепенное значение. В статье рассматриваются вопросы использования современных технологий, которые позволяют качественно повысить обучение в вузах на другой, более высокий уровень.

Ключевые слова: качество образования, национальная экономика, образовательные услуги, цифровые технологии, оценка качества.

INFLUENCE OF MODERN TECHNOLOGIES ON THE QUALITY OF EDUCATION

A.A. Kabulov

*Tashkent branch of the Russian University of Economics. G.V. Plekhanov
Uzbekistan, Tashkent, kabulov51@mail.ru*

Abstract. The quality of education is of paramount importance throughout the world today. The article discusses the use of modern technologies that will improve the quality of education in universities to a different, higher level.

Keywords: quality of education, national economy, educational services, digital technologies, quality assessment.

На сегодняшний день все образовательные учреждения, все преподаватели, клиенты, студенты, слушатели магистратуры и потребители выступают за качество образования. Но у каждой категории – свое понятие качества. В результате мы имеем различные факторы, которые сдерживают приближение системы образования к международным стандартам. В частности, нет единой системы обеспечения качества образования и системы, мотивирующей вузы внедрять принципы достижения и непрерывного улучшения качества своей работы.

Узбекистан в настоящее время продолжает реализовывать наиболее эффективные механизмы воспроизводства социально-политической структуры государства, качественного роста профессиональной рабочей силы, где должны быть учтены условия для поступательного развития национальной экономики. Совершенствование качества образования в республике и ее роль в рыночных преобразованиях общества является на сегодня основной задачей системы образования в Узбекистане.

Только высокое качество образования может стать решающим фактором устойчивого развития страны. В связи с этим сегодняшние проблемы повышения качества высшего образования в Республике Узбекистан должны быть тщательно проанализированы. Актуальность проблемы оценки, управления и повышение качеством образования объясняется еще и тем, что за последние десятилетия высшее образование приобрело широкомасштабный характер, что вызвало некоторые проблемы в эффективности образовательного процесса.

В национальном масштабе, так и в разрезе регионов управление развитием качественного образования является важнейшей составной частью Государственной Программы и требует создания систем мониторинга качества образования. Сейчас необходима такая методика оценки качества образования, которая бы не основывалась только на оценках в дипломе студента, а объективно оценивала бы действительные знания студентов, их способность реализовать на практике полученные результаты в образовании. В методику оценки качества образования следует ввести также склонность студента к новациям, стремлению к дальнейшему совершенствованию личности.

Качество – это то, что мы делаем, а не то, о чем мы говорим или показываем в анкетах для измерения рейтинга вуза по существующим методикам. Каждый вуз может использовать

свою систему измерения качества. Развитие человеческих ресурсов является самым важным компонентом Программы экономического развития Узбекистана. Усовершенствование через высшее образование способностей, знаний и жизненных навыков рабочего капитала, увеличивает продуктивность труда, которая, в свою очередь, ускоряет экономический рост страны.

В результате исследования в республике были выявлены следующие причины снижения качества высшего образования.

1. С распадом Советского Союза профессорско-преподавательский состав практически остался прежний, но часть из них выехала в страны ближнего и дальнего зарубежья. Доля новых, молодых преподавателей остается низкой и из-за отсутствия преподавательского опыта, они еще не обладают достаточной квалификации, что резко влияет на качество образования.

2. Качество образования в вузах регионов республики всегда было низким из-за недостаточной квалификации профессорско-преподавательского состава, материально-технической и учебно-методической базы, и в результате, несравнимо меньшего времени общения студентов с квалифицированными преподавателями, недостатком качественных методических пособий. Увеличение числа студентов филиалов и заочников объясняется, в основном, коммерческими интересами вузов республики.

3. Слабое нормативно-правовое обеспечение качества образования. В существующем законодательстве вообще отсутствует определение качества образования. В Положении об аттестации под аттестацией понимается соответствие качества образования государственным образовательным стандартам (ГОС). Однако, в ГОС не заложены четкие критерии и механизмы определения качества образования. Нет единого нормативного определения качества образования, что приводит к тому, что каждый понимает это по-своему. Все это является следствием того, что в образовательной политике и образовательной системе не уделяется должного внимания проблеме качества образования.

4. Недостаточно высокий процент выпускников обладает теми знаниями, умением и навыками, которое того требует сегодняшние рыночные условия жизни и, в первую очередь, требованиям рынка труда. Многие образовательные программы необходимо адаптировать и направить на формирование системного творческого мышления, высокой информационной, инновационной и цифровой культуры. Многие выпускники еще не обладают опытом работы в группе, не обладают навыками разработки и реализации реальных, практически важных и значимых проектов, слабо используют возросшие возможности новых информационных технологий.

5. Неэффективная система внутривузовского контроля качества образования. Здесь имеется в виду, что знания студентов оценивают те же преподаватели, которые их обучают. То есть, по сути, они ставят отметки самим себе. Во-вторых, существующая система не способствует мотивации преподавателей, так как материальное и моральное поощрение, их карьера, никак не зависит от качества знаний обучаемых ими студентов. В-третьих, у каждого преподавателя своя шкала качества образования, что приводит к отсутствию единой меры при измерении качества образования. Поэтому отметка «3» одного вуза может быть лучше отметки «5» другого вуза. При отсутствии единой меры измерения качества образования вообще теряется смысл понятия «качества образования». Кроме того, и преподавателям вуза становится невыгодным ужесточать требования к качеству знаний студентов, т.к. большой отсев студентов ведет к уменьшению финансовых поступлений. Особенно это относится к контрактным студентам, студентам филиалов и заочников.

6. Низкое качество в образовании «исходного сырья» – под этим подразумевается принятие на первый курс абитуриентов. Основная причина этого – неэффективная работа школ, лицеев, колледжей из-за ряда объективных и субъективных причин. В советское время был достаточно эффективный механизм входного контроля – приемные экзамены, который был разрушен. В настоящее время в Узбекистане можно поступить в вузы, оплачивая супер-

контракт, который может в 10 раз превышать обычный контракт, что приводит к тому, что в вузы поступают абитуриенты, не имеющие достаточных базовых знаний. Это, в свою очередь, ведет к тому, что в учебных группах учатся студенты с разным уровнем знаний. Преподавателям приходится ориентироваться на знания этих студентов, меньше времени уделяя одарённым студентам, из-за ее нехватки. Возникает вопрос, а как можно из «плохого сырья» сделать качественный продукт?

7. Большинство вузов республики крайне медленно перестраиваются в соответствии с требованиями рынка труда. У многих вузов до сих пор не сформулирована своя миссия, что имеет немаловажное значение в образовании. А ведь одно из главных требований современного образовательного менеджмента – это наличие четкой и ясной миссии, сформулированной на основе анализа социального заказа (работодателей, государства, местного сообщества и т.п.).

8. Несовершенная система государственного контроля качества высшего образования, которая осуществляется в виде аттестации. Отсутствуют единая мера и механизм объективного измерения качества образования. Каждая аттестационная комиссия трактует качество образования по-своему.

9. Недостаточная мотивация студентов к получению качественного образования. Причины этого были заложены еще в советские времена, когда все выпускники вузов гарантированно обеспечивались работой и возможностью сделать карьеру. Диплом гарантировал трудоустройство и, к сожалению, часто становился самоцелью. Это, в свою очередь, породило коррупцию как во время приема в вузы, так и во время обучения в нем. В наше время эти процессы также проявляются и, в конечном итоге, качество образования остается не на должном уровне.

10. Недостаточное финансирование, так как общеизвестно, качественное образование стоит дорого.

Конечно, этот анализ не является исчерпывающим, объективным со стороны автора и требует обсуждения, дополнения, корректировки по вышеуказанным предложенным причинам повышения качества образования в вузах Республики Узбекистан. Но, на наш взгляд, позволит повысить качество высшего образования в стране в результате использования современных цифровых технологий.

Несомненно, этому способствовало название в Узбекистане 2020 года, предложенного Президентом Республики Узбекистан: «Годом науки, просвещения и развития цифровой экономики» [1, с. 1].

Поэтому, в условиях реформирования и инновационного развития высшего образования в республике одной из важных составляющих повышения качества высшего образования является учебная и методическая работа, развитие педагогических навыков и информационных технологий, работа по формированию и комплектации учебно-методических комплексов нового поколения, развитие цифровизации во всех отраслях национальной экономики, и, конечно же, в образовании [2, с. 164].

Стремительная цифровизация и компьютеризация практически всех областей знания требует рассматривать информационные технологии как важнейшую составляющую фундаментальной подготовки специалистов, как актуальное научно-образовательное направление быстро развивающихся университетских учебных дисциплин. Сегодня качество вуза определяется не только количеством докторов науки, объемом томов в библиотеке, поскольку потребителей меньше всего это интересует. Их интересуют возможности получения неограниченных информационных ресурсов с помощью сети Интернет и электронных библиотек, на что и будет направлено это в предстоящих Государственных Программах по цифровизации национальной экономики.

В настоящее время необходимым условием перехода общества к информационной цивилизации является информационные технологии в образовании. Организации учебно-

воспитательного процесса, полное погружение обучаемого в информационно-образовательную среду, повышение качества образования, мотивация процессов восприятия информации и получение качественных знаний можно изменить с помощью современных технологий и телекоммуникаций. В различных сферах деятельности, в том числе в образовании новые информационные технологии, создадут среду компьютерной и телекоммуникационной поддержки организации и управления. На всех уровнях обучения и в образовательных программах необходимо осуществлять интеграцию информационных технологий.

Развитие образовательных учреждений по инновационному пути повышения качества образования невозможно без создания и совершенствования инфраструктуры информатизации и цифровизации, которая заключается, прежде всего, в повышении интеллектуальной деятельности за счёт использования информационных и телекоммуникационных технологий. Современные информационные технологии определяются как непрерывные процессы обработки, хранения, передачи и отображения информации, направленные на эффективное использование информационных ресурсов, средств вычислительной техники и передачи данных при управлении системами различного класса и назначения. Основу современных информационных технологий составляют: интерактивный режим работы, возможность хранения больших объёмов информации на машинных носителях, передача информации на любое расстояние в ограниченное время, интегрированность с другими программными продуктами гибкость процесса изменения данных и постановок задач.

Информационные технологии повышают качественный уровень эффективности работ в науке и образовании за счёт упрощения и ускорения процессов обработки, передачи и хранения информации, обеспечение точности и качества решаемых задач, возможности реализации ранее не решаемых задач, сокращение сроков разработки, трудоёмкости и стоимости научно-исследовательских работ. Наука и образование имеют много одинаковых задач. Это касается управления, как учебным процессом, так и научными исследованиями, применения математических интеллектуально-логических методов решения задач, оформления результатов, информационного обеспечения.

Качество и эффективность проводимых учебных занятий в значительной степени связана с уровнем использования компьютерной техники. Один из наиболее эффективных методов научного исследования – это вычислительный эксперимент, который позволяет изучить поведение сложных систем. Широко применяются возможности вычислительной техники для логического, структурного и функционального моделирования. При этом используются как системы универсального применения, такие как QuattroPro, MathCad, Excel, так и функционально-ориентированные программные средства. Наибольшее применение на этапе обработки результатов научных исследований и образовательных программ находят программные средства, которые обеспечивают выполнение математических расчётов с использованием математической статистики, теории ошибок, векторного и растрового анализа изображения, теории вероятности.

Информатизация университетского образования – это необходимое условие как повышения конкурентного уровня вузов на рынке образовательных услуг, так и качественной подготовки будущего специалиста в современных условиях интенсивного развития информационных и коммуникационных технологий. В развитии повышения качества и процесса информатизации образования проявляются тенденции активного внедрения новых средств и методов обучения, создания единого информационного образовательного пространства, методов обучения, ориентированных на использование технологий обработки данных, формирования систем непрерывного образования, числовой, графической, текстовой информации, искусственного интеллекта и дистанционного образования, мультимедиа и виртуальной реальности.

Одним из методов применения информационных технологий для преподавателей является автоматизация проверки знаний обучающихся, что также является способом повыше-

ния качества образования. Применение информационно-коммуникационных технологий в образовании позволило повысить качество обучения, создать новые средства эффективного взаимодействия преподавателя и обучаемого, средства воспитательного воздействия, ускорить передачу знаний.

К основным преимуществам использования информационных технологий в образовании перед традиционным обучением относят:

- существенное повышение мотивации студентов к обучению;
- активизацию умственной деятельности студентов;
- информационные технологии, которые значительно расширяют возможности учебной информации (применение анимации, звука, цвета, графики, что позволяет воссоздать реальную обстановку деятельности);
- наиболее широкое раскрытие способностей обучаемых;
- формирование рефлексии (обучающийся имеет возможность наглядно представить результат своих действий, определить этап в решении задачи, на которой сделана ошибка, и исправить её).

Информационные технологии в учебном процессе используются при проведении виртуальных лабораторных работ с использованием обучающих программ, изложении учебного материала, в системах проверки и контроля, при проведении теле- и видеоконференций. Опыт общения со студентами показал, что при использовании компьютерных систем обучения можно повысить скорость поиска нужной информации, её наглядность, качество обратной связи, эффективность учебных занятий, не менее чем на 30-40%.

Сегодня, характерной чертой обучения является постоянное совершенствование учебно-воспитательного процесса вместе с развитием и перестройкой общества, с созданием единой системы непрерывного образования в Республике Узбекистан. Чтобы привести содержание образования в соответствие с современным уровнем научного знания, повысить эффективность всей учебно-воспитательной работы и подготовить обучающихся к деятельности в условиях перехода к информационному обществу, в стране осуществляется полномасштабное реформирование вузовской системы качественного обучения. Поэтому информационные технологии становятся неотъемлемым компонентом содержания обучения, средством оптимизации и повышения эффективности учебного процесса, а также способствуют реализации многих принципов развивающегося обучения.

Таким образом, повышение качества высшего профессионального образования в Республике Узбекистан, сегодня, рассматривается как комплексная Государственная Программа, требующая эффективности преподавательской, научно-исследовательской и управленческой деятельности.

Библиографический список

1. Послание Президента Республики Узбекистан Шавката Мирзиёева Олий Мажлису. «Народное слово», 25 января 2020 г.
2. Кабулов А.А. Роль инноваций в отраслевом развитии экономических районов Республики Узбекистан. Сб. трудов II-й Научно-практической конференции: «Теория и практика финансово-хозяйственной деятельности предприятий различных отраслей». Керчь, 2020. С. 164-170.

УДК 004.946; ГРНТИ 28.17.33

ОПРЕДЕЛЕНИЕ НЕГАТИВНЫХ ФАКТОРОВ ПРИ ВЗАИМОДЕЙСТВИИ С ВИРТУАЛЬНОЙ РЕАЛЬНОСТЬЮ В ПРОЦЕССЕ ОБУЧЕНИЯ

А.Е. Архипов, А.О. Назарова

*Тамбовский государственный технический университет,
Российская Федерация, Тамбов, alexeiarh@gmail.com, nazarova.al.ol@yandex.ru*

Аннотация. В работе рассматривается влияние негативных факторов виртуальной реальности на физическое состояние человека. Проведена исследовательская работа по оценке и структуризации отрицательного воздействия виртуальной реальности на репрезентативную выборку обучающихся.

Ключевые слова: виртуальная реальность, негативные факторы обучения, сбор и обработка данных, киберболезнь.

DETERMINATION OF NEGATIVE FACTORS IN INTERACTION WITH VIRTUAL REALITY IN THE LEARNING PROCESS

A.E. Arkhipov, A.O. Nazarova

*Tambov State Technical University,
Russian Federation, Tambov, alexeiarh@gmail.com, nazarova.al.ol@yandex.ru*

The summary. The work examines the influence of negative factors of virtual reality on the physical condition of a person. Research work has been carried out to assess and structure the negative impact of virtual reality on a representative sample of students.

Keywords: virtual reality, negative learning factors, data collection and processing, cyber disease.

Введение

Актуальность внедрения технологий виртуальной реальности (VR) в сферу образования объясняется усложняющимися теориями и навыками, которые следует усвоить обучающемуся [1]. Преподавателям, которым с каждым годом приходится оперировать все большим количеством информации, стоит рассмотреть новые способы представления образовательного материала. Именно использование соответствующих технических и программных средств виртуальной реальности призвано облегчить представление знаний, а также обеспечить иммерсивность процесса обучения [2, 3]. Эффективность новых технологий тестируют на пилотных проектах, экспериментах, в рамках научно-исследовательских работ для выявления факторов влияния на все направления жизни – от стоимости и окупаемости до изменения психологического состояния человека.

Технологии виртуальной реальности могут помочь в решении большого круга экономических проблем, т.к. зачастую закупка VR оборудования и разработка программного обеспечения для него обходится дешевле, чем организация данных работ в реальности. Еще одним важным преимуществом является безопасность целого ряда специалистов, подвергающих угрозам свою жизнь и здоровье в процессе профессиональной деятельности [4].

Однако, как и любая современная технология, погружение в цифровое пространство имеет свои негативные стороны. В настоящее время все еще недостаточно изучено ее влияние на здоровье человека, при том, что зачастую использование VR-шлемов приводит к временному ухудшению физических показателей.

Негативные последствия от взаимодействия с цифровым пространством могут быть признаками киберболезни, основные симптомы которой сходны с симптомами морской болезни – это тошнота, рвота, головокружение, головная боль, умственное и физическое утомление, сонливость, нарушение концентрации внимания, дезориентация в пространстве и усталость глаз. Также могут проявиться бледность, затрудненное глотание, повышенное слюноотделение и холодный пот.

В работе [5] проведен подробный анализ недостатков технологий виртуальной и дополненной реальности. Авторами получены следующие результаты с точки зрения влияния на пользователя:

- тошнота, головокружение, головная боль, усталость глаз;
- нагрузка на шею и позвоночник;
- потеря ориентации, ощущения времени, реальности;
- столкновение с объектами реального мира, травмоопасность;
- рассеянное внимание, потеря фокуса, утомляемость.

Для достижения иммерсивности VR-гарнитура воздействует на три сенсорные системы человека: зрительную, вестибулярную и проприоцептивную. И если бы цифровое пространство было способно полноценно симулировать реальное окружение, то ответных физических реакций организма не возникло бы. Однако из-за своей новизны и несовершенства виртуальная реальность «обманывает» восприятие человека. Мозг фиксирует положение в пространстве с помощью органов зрения и вестибулярного аппарата, и именно их взаимодействие нарушается в процессе, ведь органы не могут синхронизировать зрительную информацию и положение тела в реальном пространстве.

Причины киберболезни обосновываются прежде всего спецификой аппаратного обеспечения и ограничениями существующих программных систем, которые не позволяют полностью исключить негативные физические последствия. Так, установленным фактом является прямая зависимость ширины угла обзора (поля зрения), которое предоставляет средство визуализации, и силы проявления симптомов киберболезни [6]. Самым очевидным примером является плохая эргономика некоторых бюджетных моделей VR-шлемов: при резких движениях положение экрана смещается относительно глаз, что также может вызвать симптомы морской болезни.

В работе [7] рассматриваются как причины, так и способы противодействия киберболезни. Основные причины возникновения киберболезни — это сенсорный конфликт и постуральная нестабильность.

Сенсорный конфликт – несовпадение информации, обрабатываемой зрительной системой, с информацией, получаемой вестибулярным аппаратом. В случае виртуальной реальности на основе шлемов VR, пользователи испытывают иллюзию самодвижения, когда человек чувствует телесное движение несмотря на то, что никакого движения не происходит. Человек может испытывать иллюзорные движения всего тела или отдельных частей тела, таких как руки или ноги.

Второй наиболее вероятной причиной является постуральная нестабильность (равновесие) – индивидуальная способность человека удерживать равновесие тела в той или иной позе, а также при изменении позы. То есть человек с более сильной физической устойчивостью менее подвержен возникновению киберболезни.

Так же на основе литературы было определено три возможных способа снижения киберболезни: ограничение возможности движений, активный контроль обучаемого, проведение повторных тренировок.

Ограничение возможности движений подразумевает ограничение телодвижения и повышения стабильности осанки за счет использования ограничений, таких как ремни безопасности в автомобилях.

Активный контроль предполагает, что возникновение киберболезни зависит от активности обучаемого в виртуальном пространстве. Известно, что в автотренажерах водители меньше подвержены возникновению киберболезни, чем пассивные наблюдатели «пассажиры».

Повторное воздействие VR. В исследовании выявлено, что повторные 15-минутные тренировки в течение 3 дней дают значительное снижение субъективных и физиологических показателей киберболезни.

Исследование проводилось на 20 испытуемых с применением приложения для VR, в котором была возможность как свободного перемещения, так и пассивного, в роли наблюдателя, а также возможность взаимодействия с виртуальными предметами при помощи контроллера VR. Постуральная стабильность оценивалась с помощью специальной балансировочной пластины, а в качестве элемента ограничения тела использовался страховочный ремень с обвязкой.

Каждый участник был испытан с четырьмя вариантами выполнения упражнений в течение двух дней. Два тестирования были разделены двенадцатичасовыми промежутками (в целях ограничения остаточных симптомов укачивания от тестовой сессии в первый день). В результате участники, которые получили симптомы киберболезни имели различную спонтанную постуральную активность по сравнению с теми, кто не получил симптомов.

Для оценки эффективности взаимодействия можно применять качественные и количественные показатели. Под количественными показателями будем понимать метрики, выраженные численными значениями, например, время выполнения задания, количество ошибок, точность позиционирования виртуальных объектов. К качественным относятся метрики, отражающие субъективные впечатления пользователя о процессе взаимодействия с виртуальной реальностью. Для их сбора необходимо сформировать перечень вопросов в виде опросного листа, позволяющий получить комплексную оценку ощущений и впечатлений пользователя о процессе взаимодействия с виртуальной реальностью.

Актуальной проблемой является сбор и обработка таких опросных листов. В рамках данного исследования рассматривается выбор основных негативных факторов при взаимодействии с виртуальной реальностью, а также механизм сбора и обработки о влиянии этих факторов на процесс обучения.

Методика сбора и обработки данных при взаимодействии с виртуальной реальностью

Для решения поставленной проблемы разработано соответствующее программное обеспечение, методика организации сбора экспериментальных данных и их последующей обработки.

Программное обеспечение тестировалось на фокус-группе, состоящей из студентов различных направлений: инженерных, информационных, гуманитарных. Также в фокус группу включены сотрудники университета и преподаватели.

Рассмотрим методику проведения эксперимента. Она включает три основных этапа.

1. Ввод личных данных испытуемого и авторизация в тренажере (сбор личных данных).
2. Прохождение упражнений в виртуальной реальности (сбор количественных метрик).
3. Прохождение опроса (сбор качественных метрик).

На первом этапе каждый участник из четырех фокус-групп вносит свои личные данные (рисунок 1, справа). Далее он выбирает определённую конфигурацию обучающего сценария.

Виртуальная сцена «Перемещение простых объектов» (рисунок 1, слева). Используется для отработки навыков перемещения в виртуальном пространстве и перемещении и позиционирования простейших объектов.

Параметры виртуального окружения:

- комната с аватаром пользователя;
- тестовый стенд с набором объектов разных размеров. Рядом с объектами – обозначенные цветом области-приемники, в центр которых необходимо поместить объекты.

Метрики:

точность: среднее расстояние между центром объекта и центром области, а также угол поворота объекта;

время: от начала упражнения до размещения последнего объекта;

количество ошибок: предмет размещен вне требуемой области.

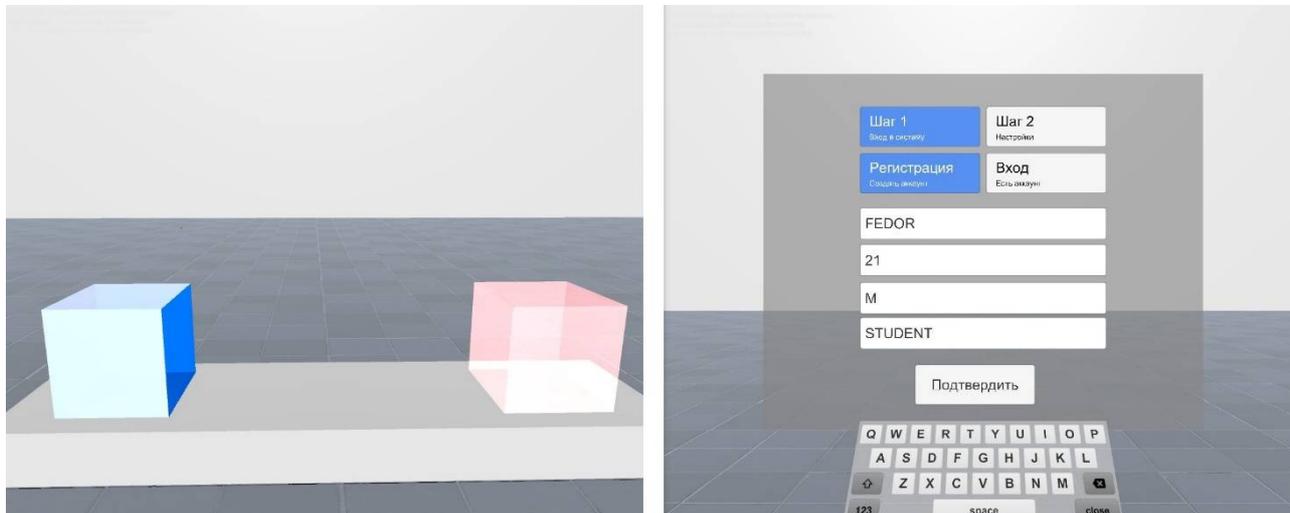


Рис. 1. Виртуальная сцена и окно регистрации

Эксперимент проводится на различных конфигурациях тренажерного комплекса и многократно для получения данных о росте уровня навыков и умениях испытуемых.

Далее осуществляется сбор качественных метрик. Выбраны следующие метрики негативных факторов:

- V1. Испытывали ли Вы тошноту?
 - V2. Испытывали ли Вы укачивание?
 - V3. Испытывали ли Вы головокружение?
 - V4. Были ли дезориентированы в пространстве?
 - V5. Оцените по шкале удобство выполнения действий.
 - V6. Оцените по шкале удобство перемещения.
 - V7. Оцените по шкале удобство обзора в виртуальной реальности.
 - V8. Оцените по шкале свойства объектов (детали, расстояние до объектов, размер).
- Интерфейс для сбора метрик представлен на рисунке 2.

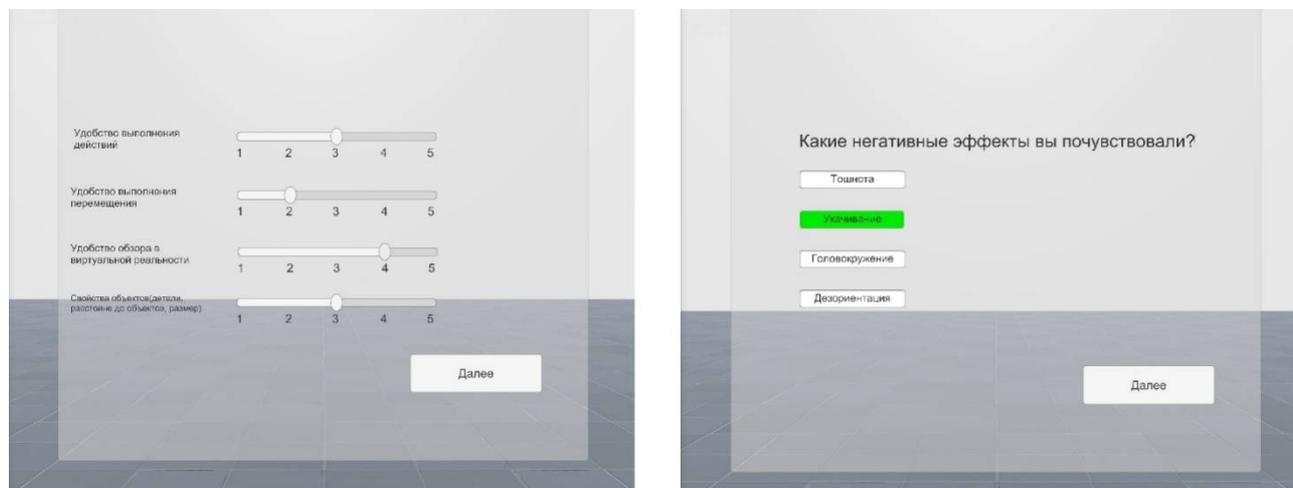


Рис. 2. Форма для сбора метрик негативных факторов

Каждая метрика рассчитывается следующим образом: «сумма значений метрики» разделить на «количество опрошенных в выборке». Таким образом, для негативных эффектов получаем процентное соотношение тех респондентов, что испытывали в ходе обучения негативные ощущения. Следовательно, эффективность процесса освоения средств взаимодействия с виртуальной реальностью отражается в уменьшении количества респондентов, испытывающих негативные ощущения (вопросы В1-В4). По вопросам В4-В8 низкие оценки также обозначают наличие негативного влияния виртуальной реальности.

Результат пройденного опроса записывается в базу данных в таблицу, соответствующую идентификатору текущего юзера.

После прохождения тренажерного комплекса необходимо обработать общий массив информации с целью выявления закономерности влияния виртуальной реальности на состояние людей, имеющих разный возраст, пол и опыт взаимодействия с цифровыми устройствами.

В результате определен набор негативных факторов, которые будут в дальнейшем проанализированы с помощью статистических или интеллектуальных методов. Это позволит предсказывать необходимость и эффективность обучения школьников, студентов и специалистов в виртуальной реальности. Дальнейшие исследования связаны с организацией масштабных испытаний и сбора данных о влиянии негативных факторов на объективные параметры процесса обучения – точность и время выполнения упражнений в виртуальной реальности.

Исследование выполнено в рамках гранта РФФИ № 19-013-00567.

Библиографический список

1. Архипов, А. Е. Проектирование системы визуализации тренажерного комплекса на основе компетентного подхода / А. Е. Архипов, А.И. Попов, А.Д. Обухов // Вопросы журналистики, педагогики, языкознания. – 2020. – Т. 39. – №. 3. – С. 378-390.
2. Krasnyansky M. et al. Methodology of Forming the Readiness of Miners for Work i Extreme Situations Using a Training Complex //International Journal of Emerging Technologies in Learning (iJET). – 2020. – Т. 15. – №. 02. – С. 86-97.
3. Krasnyanskiy M. et al. Modeling of the learning process in adaptive training complexes / A. D. Obukhov, M. N. Krasnyansky, D. L. Dedov, A. A. Siukhin //Journal of Applied Engineering Science. – 2018. – Т. 16. – №. 4. – С. 487-493.
4. Obukhov A. et al. A Mathematical Model of Organizing the Developmental Instruction in the System of Professional Education //Tehničivjesnik. – 2020. – Т. 27. – №. 2. – С. 480-488.
5. Иванова А. В. Технологии виртуальной и дополненной реальности: возможности и препятствия применения //Стратегические решения и риск-менеджмент. – 2018. – №. 3 (108). - С. 88-107.
6. Van Emmerik M. L., de Vries S. C., Bos J. E. Internal and external fields of view affect cybersickness //Displays. – 2011. – Т. 32. – №. 4. – С. 169-174.
7. Risi D., Palmisano S. Effects of postural stability, active control, exposure duration and repeated exposures on HMD induced cybersickness //Displays. – 2019. – Т. 60. – С. 9-17.

УДК 004.896; ГРНТИ 50.47.29

ИНФОРМАЦИОННАЯ ПОДДЕРЖКА ПРИНЯТИЯ РЕШЕНИЙ ПРИ ПРОВЕДЕНИИ ТЕХНОЛОГИЧЕСКОГО ОБСЛУЖИВАНИЯ ХЛЕБОПЕКАРНОЙ ПЕЧИ

Д. А. Ищенко*

**Тамбовский государственный технический университет,
Россия, Тамбов, dimonix68@mail.ru*

Аннотация. Рассмотрены вопросы информационной поддержки принятия решений при проведении технологического обслуживания хлебопекарной печи перед началом и в ходе выполнения технологических операций. Разработана информационно-логическая модель поддержки принятия решений, позволяющая в зависимости от текущего состояния узлов и деталей печи предложить оптимальный вариант проведения технологического обслуживания и наладки печи.

Ключевые слова: информационная поддержка; принятие решений; технологическое обслуживание; хлебопекарная печь.

INFORMATION SUPPORT FOR DECISION-MAKING DURING THE TECHNOLOGICAL MAINTENANCE OF THE BAKING OVEN

D.A. Ishchenko

**Tambov state technical University,
Russia, Tambov, dimonix68@mail.ru*

Abstract. The issues of information support for decision-making during the technological maintenance of the baking oven before and during the execution of technological operations that ensure high quality of manufacturing parts are considered. The information and logical model of decision support has been developed, which allows us to offer the best option for technological maintenance and adjustment of the furnace, depending on the current state of the furnace components and parts.

Keywords: information support, decision-making, technological maintenance, baking oven.

Производство хлеба— древнейшая отрасль пищевой промышленности. Общие принципы современной технологии хлебопечения были созданы в Древнем Египте 5...6 тыс. лет назад. Окончательное рождение замечательного продукта, называемого хлебом, было связано с соединением в единый процесс трех великих технологий древности: выращивание пшеницы хорошего качества, применение жерновов при помоле и использование дрожжей для брожения теста. В дальнейшем эта технология получила распространение в Древней Греции и Римской империи.

В дореволюционной России выпечка хлеба осуществлялась в домашних условиях, в городах работали мелкие, как правило, немеханизированные пекарни. В годы первых пятилеток были приняты решения о механизации хлебопечения, широком строительстве хлебозаводов, создании машиностроительной базы для производства отечественного хлебопекарного оборудования.

Развитие отрасли в «советский» период отличала высокая концентрация и специализация производства. Другая отличительная черта — широкое распространение непрерывного трехсменного режима работы, что обеспечивало более высокую стабильность технологических параметров. Однако, вместе с тем, указанные особенности затрудняли снабжение населения свежим хлебом, тормозили выработку широкого ассортимента изделий [1 - 2].

В настоящее время наметились структурные изменения в обеспечении населения Российской Федерации хлебными изделиями — произошел переход от строительства крупных хлебозаводов к развитию сети предприятий малой мощности.

По количеству предприятий, объему и значимости продукции, стоимости основных производственных фондов хлебопекарная промышленность является одной из ведущих отраслей пищевой промышленности России. Однако в настоящее время по оценкам специали-

стов только 20...35% хлебопекарных предприятий соответствуют современному техническому уровню. Свыше 30% активной части основных производственных фондов имеют возраст от 10 до 20 лет, степень механизации труда не превышает 40%.

Развитие технической базы хлебопекарной отрасли должно быть направлено на:

- рациональное сочетание специализированной и универсальной техники для выработки массовых и специальных сортов, новых видов продукции;
- значительное повышение эксплуатационной надежности и ремонтпригодности машин и аппаратов;
- создание технологического оборудования для небольших пекарен;
- оснащение линий, отдельных участков и машин компьютерной и микропроцессорной техникой.

В соответствии с Федеральной целевой программой стабилизации и развития агропромышленного комплекса научно-техническая политика и области хлебопечения должна быть направлена прежде всего на снижение затрат материальных, энергетических и других видов ресурсов; создание, производство и использование новых видов сырья, в том числе нетрадиционного; повышение производительности труда.

Особое внимание должно уделяться разработке новых технологий и оборудования, уменьшению расхода электроэнергии и материальных ресурсов, снижению доли ручного труда и металлоемкости конструкций. Решение этих задач возможно лишь на основе глубоких знаний технологических процессов и существующего оборудования.

Хлебопекарные печи – это высокотехнологичное оборудование, предназначенное для выпекания изделий из теста. Современные печи обеспечивают высокое качество и равномерность выпечки хлебобулочных изделий. Все они оснащены большим количеством контрольно-измерительных приборов, которые позволяют автоматизировать работу печи в значительной степени.

Сегодня хлебопекарное оборудование представлено большим ассортиментом печей. Выбор хлебопекарной печи зависит, в первую очередь, от вида выпекаемых изделий. Для этого стоит разобраться в классификации печей, которые существуют на рынке хлебопекарного оборудования [3 - 4].

Хлебопекарные печи по виду конструкции делятся следующие виды:

1. Конвекционные печи. В таких печах воздух распределяется равномерно по всей площади камеры при помощи циркуляционного вентилятора. В них также действует система пароувлажнения, а пекарная камера освещается световыми приборами. Такие печи позволяют контролировать процесс выпечки визуально, так как их дверцы изготовлены из двойного стекла. Конвекционные печи достаточно компактны и просты в использовании, поэтому отлично подходят для небольших производственных предприятий. В конвекционных печах можно выпекать изделия из замороженного теста, батоны, отдельные виды кондитерских изделий.

2. Подовые печи обычно состоят из нескольких ярусов (от 1-го до 4-х). В под такой печи загружаются противни определённого размера. Если под керамический, то выпекать изделия из теста можно непосредственно на его поверхности, без использования противня.

В некоторых подовых печах применяют масло в качестве промежуточного теплоносителя (циклотермические подовые печи).

3. Ротационные печи. Ротационные печи оснащены электронной программируемой панелью управления. В них горячий воздух от ТЭНов сдувается вентиляторами. В пекарную камеру можно загружать одну или несколько стеллажных тележек, которые держатся на платформенном или крюковом креплениях. В процессе выпечки стеллажная тележка постоянно вращается, то есть совершает ротацию. Отдельные виды ротационных печей осна-

щены встроенным парогенератором, обеспечивающим непрерывную подачу пара в камеру в процессе выпечки. Это позволяет добиться глянцевой поверхности выпекаемых изделий.

4. Комбинированные печи объединяют в себе подовую печь и конвекционную, поэтому в ней можно выпекать и кондитерские, и хлебобулочные изделия. Такие печи оснащены расстоечной камерой.

Комбинированные печи идеально подойдут для пекарни, в ассортимент которой входят кондитерские и хлебобулочные изделия собственного производства.

5. Туннельные печи используются на достаточно крупных производствах, так как предназначены для непрерывной выпечки всех видов изделий из теста. Хлебобулочные изделия укладываются на металлические пластины, при этом за счёт изменения скорости движения пластин регулируется время выпечки.

По виду энергоносителей, используемых в печах, хлебопекарное оборудование можно разделить на два вида: топливное и электрическое.

Производительность пекарни определяется возможностями хлебопекарного оборудования – это и скорость выпечки, и количество возможного использования противней в печи и многое другое. Выбирать необходимое оборудование следует исходя из ассортимента продукции и планируемого объёма производства. В настоящее время большее предпочтение отдаётся ротационным печам, позволяющим выпекать любые виды хлебобулочных и кондитерских изделий.

Разработка информационно-логической модели поддержки принятия решений

Основным элементом системы поддержки принятия решения при проведении технологического обслуживания хлебопекарной печи перед началом и в ходе выполнения технологических операций является информационно-логическая модель (ИЛМ). В общем виде ИЛМ поддержки принятия решений представляет собой объединение множеств данных и связей между ними в виде правил [5 - 7]. Отдельное производственное правило, содержащееся в базе знаний, состоит из двух частей: условной, состоящей из элементарных предложений, соединённых логическими связками «и», «или», и заключения, включающего в себя одно или несколько предложений, выражающих либо некоторый факт, либо указание на определенное действие, подлежащее исполнению.

Таким образом, ИЛМ может быть представлена следующим кортежем [8 -9]:

$$M = (d_1, \dots, d_i, \dots, d_N, p_1, \dots, p_j, \dots, p_S),$$

где m – оператор ИЛМ; d_1, \dots, d_N – множество данных ИЛМ; p_1, \dots, p_S – множество правил.

В свою очередь правила, входящие в модель, построены по типу: *если ... (условия выполняются), то ... (реализация следствия)*.

Рассмотрим правила, на основе которых должны приниматься решения по формированию перечня мероприятий технологического обслуживания хлебопекарной печи перед началом и в ходе выполнения технологических операций.

Ниже приведен пример правил применительно к хлебопекарной печи типа «РОТОР-АГРО» (рис. 1):

а) *Правила выбора мероприятий при неисправностях электрооборудования:*

– *если «неисправность печи» = «при включении печи кнопкой нет индикации на панели управления. индикатор не работает.», то «выключен вводной выключатель печи» = «включить вводной выключатель»;*

– *если «неисправность печи» = «при включении печи кнопкой нет индикации на панели управления. индикаторне работает», то «нет напряжения питания сети» = «подать напряжение питания»;*

– если «неисправность печи» = «при включении печи кнопкой нет индикации на панели управления. индикатор не работает», то «качество электрической энергии не соответствует ГОСТ 13109-97» = «проверить параметры электрической сети»;

– если «неисправность печи» = «при включении печи кнопкой нет индикации на панели управления. индикатор не работает», то «отключился автоматический выключатель цепи управления» = «выяснить причину и устранить ее, включить автоматический выключатель»;

– если «неисправность печи» = «при включении печи возникает повышенный шум электродвигателей», то «отсутствие одной из фаз питающего напряжения на вводе в печь» = «проверить наличие напряжения на каждой из фаз питающего кабеля при помощи вольтметра или пробника»;

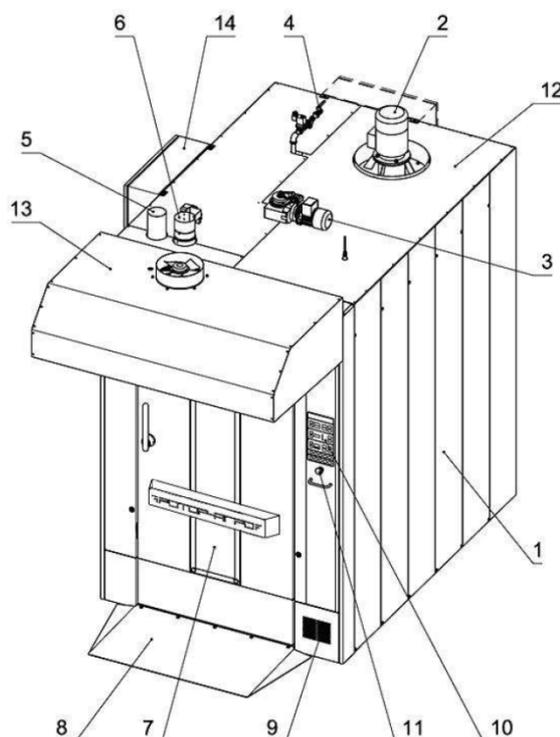


Рис. 1. Схема хлебопекарной печи типа «РОТОР-АГРО»:

1- корпус печи; 2- тепловой вентилятор; 3- привод поворотного стола; 4- трубопровод парогенератора; 5- труба вытяжной вентиляции; 6- труба с приводной заслонкой; 7- дверь; 8- трап; 9- вентилятор охлаждения; 10- панель управления; 11-кнопка аварийного отключения; 12- верхние листы; 13-козырек; 14-шкаф электрооборудования

– если «неисправность печи» = «при запираии двери пекарной камеры поворотом ручки против часовой стрелки до упора, не происходит включения вентилятора, вращения дискастеллажной тележки, блока электронагревателей», то «не отрегулировано положение толкателя конечного выключателя закрывания двери» = «отрегулировать положение конечного выключателя»;

– если «неисправность печи» = «при включении печи не включается двигатель вентилятора или вытяжки или приводтележки.», то «работало соответствующее тепловое реле» = «произвести ручной возврат теплового реле в исходное состояние»;

– если «неисправность печи» = «при разогреве печи ТЭНы отключаются, когда температура в камере выпечки ещё не достигла заданного значения», то «не отрегулирована температура установки защитного датчика-реле температуры, предохраняющего печь от перегрева» = «отрегулировать датчик-реле температуры согласно инструкции»;

– если «неисправность печи» = «при включении печи лампы освещения камеры выпечки не загораются», то «перегорела лампа» = «заменить лампу»;

– если «неисправность печи» = «при включении печи лампы освещения камеры выпечки не загораются», то «сработал автоматический выключатель цепи освещения» = «включить автоматический выключатель»;

– если «неисправность печи» = «запах горелой изоляции, нагрев зажимов электрических контактов», то «ослабили зажимы электрических контактов проводов, пускателей, реле, выключателей, ТЭНов, электродвигателей» = «подтянуть винты и гайки всех зажимов электрических контактов печи».

а) *Правила выбора мероприятий при неисправностях узлов и механизмов печи:*

– если «неисправность печи» = «утечка горячего воздуха (пара) по периметру двери», то «деформация или износ уплотнения двери с корпусом печи» = «заменить уплотнение двери».

– если «неисправность печи» = «повышенный шум при включении вентилятора теплового блока», то «отсутствие смазки в узле двигателя вентилятора» = «снять двигатель и сделать ревизию электродвигателя»;

– если «неисправность печи» = «при вращении поворотного стола появились рывки, неравномерность движения, скрип, посторонние звуки», то «выход из строя подшипника привода вращения тележки» = «разобрать подшипниковый узел, поменять подшипник»;

– если «неисправность печи» = «не хватает крутящего момента для движения поворотного стола», то «не хватает момента для поворота стола» = «отрегулировать момент вращения»;

– если «неисправность печи» = «плохо просматривается пекарная камера», то «загрязнено смотровое стекло» = «промыть стекла».

Программная реализация информационно-логической модели осуществлена в среде экспертной системы CLIPS, основу составляет полноценный объектно-ориентированный язык COOL.

В результате проведенных исследований разработана информационно-логическая модель поддержки принятых решений при проведении обслуживания хлебопекарной печи перед началом и в ходе выполнения технологических операций, позволяющая в зависимости от текущего состояния узлов и деталей печи и предложить оптимальный вариант мероприятий.

Библиографический список

1. Кулешов О.Ю., Седелкин В.М., Кузьмичева Т.В. Математическая модель и методика расчета газодинамики и теплообмена в хлебопекарных печах. - Тр. Росс. нац. конф. по теплообмену, М. 1998.– Т.6. – С. 204-206.
2. Кулешов О.Ю., Седелкин В.М. Разработка численного метода расчета тепломассопереноса в хлебопекарных печах / О.Ю. Кулешов, В.М. Седелкин // Сб. докладов Юбилейной Междунар. науч.-практ. конф. «Пищевые продукты XXI века». – М.: МГУПП, 2001. – Т.2.- С. 61.
3. Хромосенков В.М. Оборудование хлебопекарного производства: Учеб. для нач. проф. образования. - М.: ИРПО; Изд. центр «Академия», 2000. – 320 с.
4. Машины и аппараты пищевых производств. В 2 кн. Кн. 2: Учеб. для вузов/ С.Т.Антипов, И.Т. Кретов, А.Н. Остриков и др.; Под ред. Акад. РАСХН В.А. Панфилова. - М.: Высш. шк., 2001. – 1384 с.
5. Mokrozub V.G., Egorov S.Ya, Nemtinov V. A. Procedural and information-logical models of planning production and repairs of technological equipment for multi-assortment production. - Information technologies in design and production. 2009. No. 2. - pp. 2 - 3.
6. Nemtinov V.A., Zazulya A.N., Kapustin V.P., Nemtinova. Yu.V. Analysis of Decision-Making Options in Complex Technical System Design. - Journal of Physics: Conference Series. 2019. – Vol. 1278 (1). – P. 012018.
7. Nemtinov V., Bolshakov N., Nemtinova Yu. MATEC Web of Conferences: International Conference on Modern Trends in Manufacturing Technologies and Equipment (ICMTMTE 2017), 11 - 15 September, 2017, Sevastopol, 2017, vol. 129, P. 01012.
8. Мокрозуб В.Г., Немтинов В.А., Егоров. С.Я. Информационно-логические модели технических объектов и их представление в информационных системах. - Информационные технологии в проектировании и производстве. 2010. № 3. – С. 68 – 73.
9. Немтинов В.А., Бобылев В.Ю., Немтинова Ю.В., Борисенко А.Б. Информационное обеспечение принятия решений при технологическом обслуживании токарных станков. - Вестн. Тамб. гос. техн. ун-та. 2020. Т. 26. № 2. - С. 220–229.

УДК 378.1; ГРНТИ 14.35.09

ИЗУЧЕНИЕ ИСТОРИЧЕСКИХ ПРОЦЕССОВ С ИСПОЛЬЗОВАНИЕМ ТЕХНОЛОГИИ E-LEARNING

К.В. Немтинов*, А.Ю. Вязинкин*, И.В. Двухжилова*, С.М. Терехов*

*Тамбовский государственный технический университет,
Россия, Тамбов, vyazinkin@yandex.ru

Аннотация. Рассмотрена технология электронного обучения, базирующаяся на программной среде для разработки виртуальных туров 3D Vista Virtual Tour, при изучении исторических процессов памятных мест Тамбовского края, связанных с пребыванием в них известных деятелей России.

Ключевые слова: технология e-learning, исторические процессы, программная среда для разработки виртуальных туров 3D Vista Virtual Tour.

STUDY OF HISTORICAL PROCESSES USING E-LEARNING TECHNOLOGY

K.V. Nemtinov*, A.Yu. Vyazinkin*, I.V. Dvukhzhilova*, S.M. Terekhov*

*Tambov state technical University,
Russia, Tambov, vyazinkin@yandex.ru

Abstract. The article considers the e-learning technology based on the software environment for the development of virtual tours 3D Vista Virtual Tour, when studying the historical processes of memorable places of the Tambov Region associated with the stay of famous Russian figures in them.

Keywords: e-learning technology, historical processes, software environment for the development of virtual tours 3D Vista Virtual Tour.

Одной из форм технологии электронного обучения является веб-квест – релевантная и актуальная форма взаимодействия субъекта познания с символической средой историко-культурной памяти. Преимущество этой формы определяется, в том числе, и возможностью, как индивидуального, так и группового прохождения (решения) квеста, что стимулирует развитие коммуникационных навыков и более интенсивный обмен информацией в процессе познания или освоения нового академического материала [1].

В данной работе при изучении исторических процессов памятных мест Тамбовского края, связанных с пребыванием в них известных деятелей России, авторами предлагается технология электронного обучения, базирующаяся на программной среде для разработки виртуальных туров 3D Vista Virtual Tour. Её разработчики в 2020 году оптимизировали рабочий процесс и функции для всего, что связано с электронным обучением: карты викторины, результаты, геймификация и автоматическая интеграция с LMS, в том числе с одной из самых используемых - системой Moodle. Это позволяет использовать 360° виртуальные туры в качестве основы для более захватывающего, интерактивного и реалистичного обучения.

Реализация виртуального тура выполнена на примере усадьбы Сатиных и Рахманиновых в деревне «Ивановка» Тамбовской области (Россия). При этом выполнена визуализация двух временных периодов. Первый период охватывает настоящее время "Музея-заповедника С.В. Рахманинова" (1873-1943) - русского композитора пианиста, дирижёра, которому удалось соединить направления в музыкальном искусстве России и Западной Европы, создав на этой основе свой оригинальный музыкально-композиторский стиль. Второй период (начало XX века) связан с пребыванием С.В. Рахманинова в этой усадьбе. Переход между временными периодами осуществляется по специальным ссылкам расположенным в разных местах виртуального тура. На рисунке 1 приведена визуализация фрагмента сферической панорамы общего вида усадьбы Сатиных.

Реализация виртуального пространства в виде виртуального музея авторами осуществлялась с помощью программной платформы с открытым кодом OpenSimulator 0.9.0, представляющей из себя серверную платформу для 3D виртуальных миров. Визуализация музея доступна всем пользователям сети Интернет [2 - 4]. Таким образом, создана тематическая

информационная среда, которая способна обеспечить доступ к актуальной информации, и способствует росту интереса молодежи к проблемам национальной культуры, истории, а также социальной (гражданской) культуры жизнедеятельности.

На рисунке 1 представлен фрагмент сферической панорамы вида музея, включая вход в него.

Для создания обучающего квеста в программе 3D Vista Virtual Tour на вкладке «опубликовать» необходимо активировать раздел e-learning [5]. Затем с помощью встроенных инструментов электронного обучения – e-learning – можно настроить все необходимые элементы квеста. На рисунке 2 показан пример визуализации одного из вопросов квеста по модели усадьбы Сатиных.

В конце прохождения квеста пользователь может увидеть его результаты. Он может загрузить данные своих результатов прохождения тренинга в виде файла .csv или сразу же отправить его в систему LMS, в частности систему Moodle. На рисунке 3 приведена визуализация промежуточных результатов прохождения квеста.

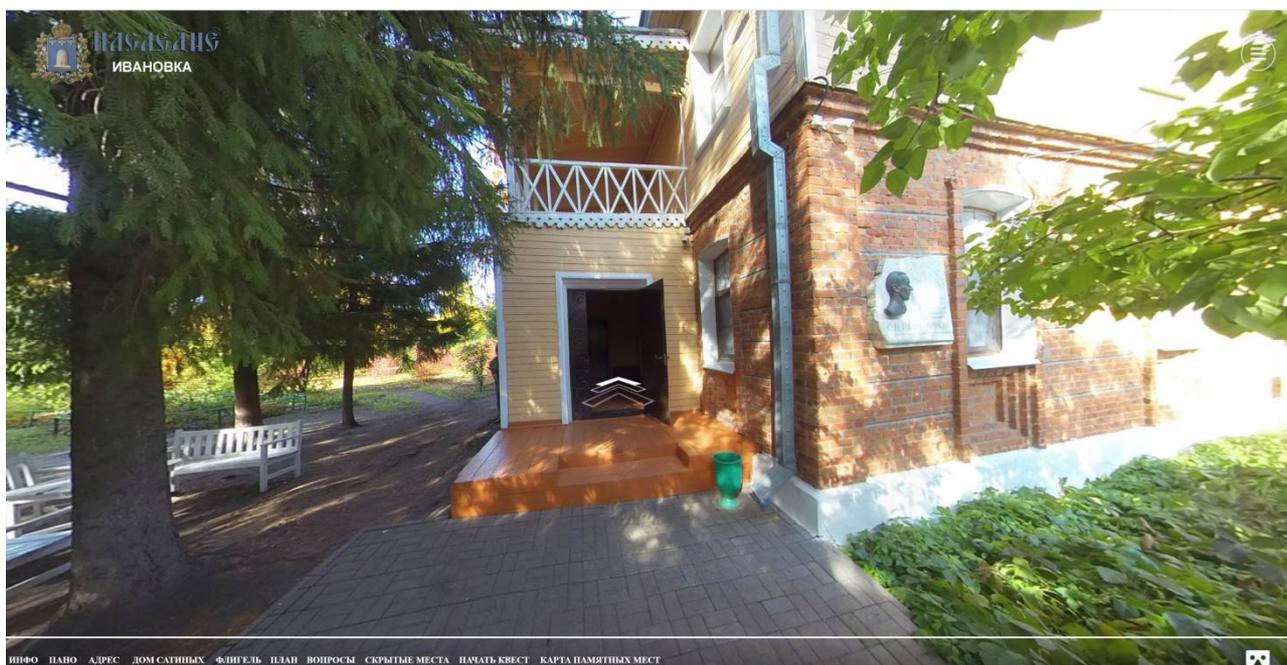


Рис. 1. Визуализация фрагмента сферической панорамы вида усадьбы Сатиных

Виртуальное пространство созданной образовательной среды, безусловно, предоставляет возможность для существования различных форм образовательной коммуникации, исходя из возможностей современных средств обучения и современных информационно-коммуникационных технологий, позволяет обеспечить условия для повышения качества образования, обмена мнениями, взаимного консультирования.

Исследование выполнено за счет гранта Российского научного фонда (проект № 19-18-00044) - «Анализ исторических процессов памятных мест Тамбовского края, связанных с пребыванием в них известных деятелей России, с использованием технологии виртуального моделирования».

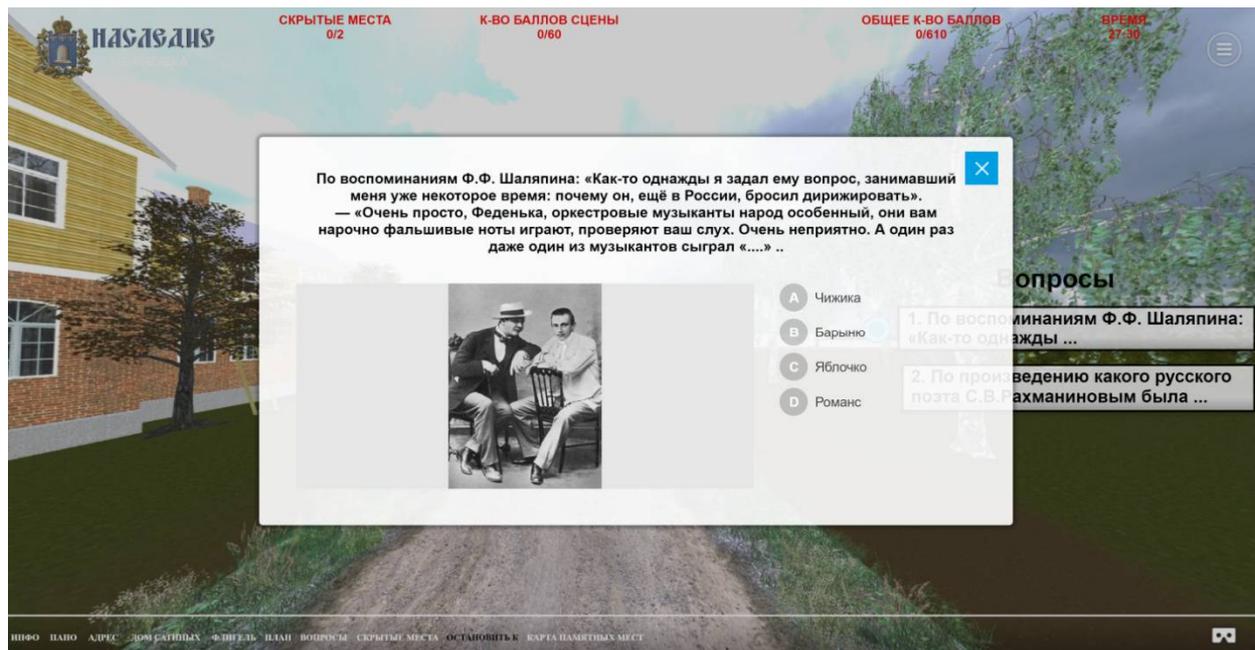


Рис. 2. Визуализация одного из вопросов квеста по модели усадьбы Сатиных

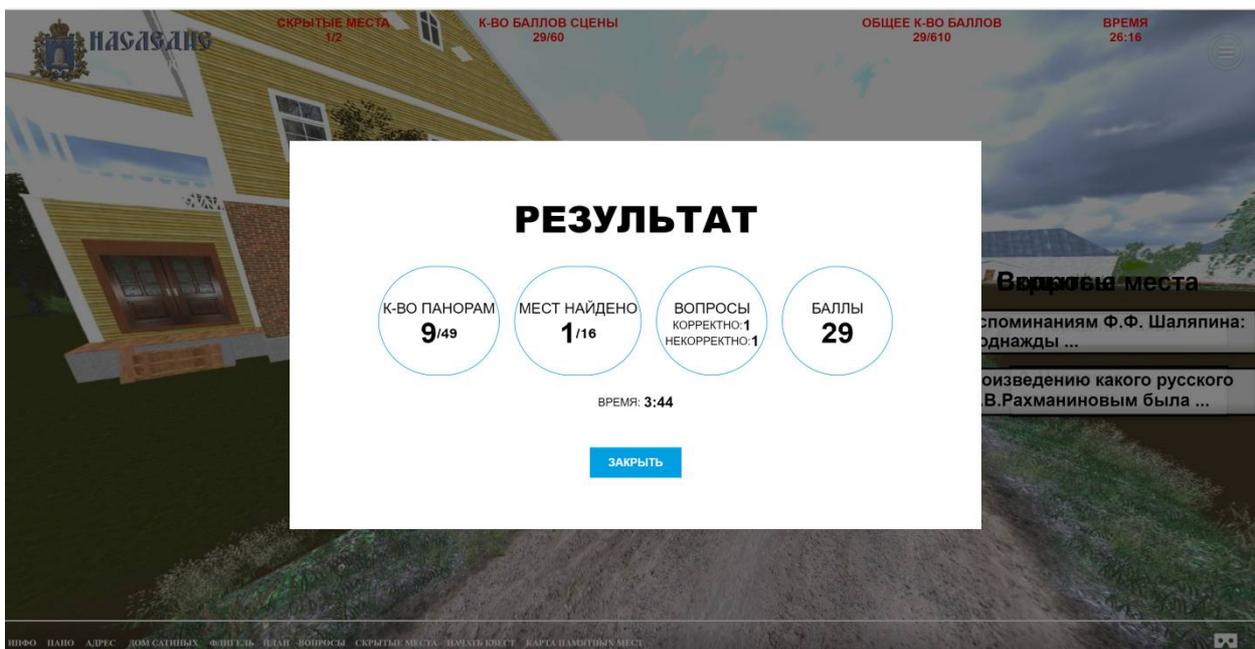


Рис. 3. Визуализация промежуточных результатов прохождения квеста

Библиографический список

1. Tsarapkina Y.M., Yakubova E.Y. Web-quest technology usage in professional self-determination - RUDN Journal of Informatization in Education. 2018. 4 (15). - pp. 373–381.
2. Nemtinov V.A., Gorelov A.A., Nemtinova Y.V., Borisenko A.B. Visualization of a virtual space and time model of an urban development territory - Scientific Visualization. 2016. 1. - pp. 120–132.
3. Nemtinov V.A., Gorelov A.A., Nemtinova Y.V., Borisenko A.B. Implementation of technology for creating virtual spatiotemporal models of urban development history - Scientific Visualization 2018. 3. - pp. 99–107.
4. Viazinkin A.Y., Dvuzhzhilova I.V., Gorelov A.A. Formation of a “Memorial Place”: Commemorative Materials and Virtual Reconstruction (On the Example of the Memorial Place “Ivanovka”) - Advances in Economics, Business and Management Research. 2020. Vol. 114. - pp. 570-573.
5. Virtual Tours in E-Learning, Training & Quizzing. <https://blog.3dvista.com/2020/04/27/virtual-tours-in-e-learning-training-quizzing/>

УДК 658.5; ГРНТИ 82.33.13

РАЗРАБОТКА IDEF0-МОДЕЛИ ПРОЦЕССА ПРОЕКТИРОВАНИЯ АРХИТЕКТУРЫ И ВЫБОРА ОБОРУДОВАНИЯ ЛОКАЛЬНОЙ ВЫЧИСЛИТЕЛЬНОЙ СЕТИ

И.А. Соловьева*, **Д.С. Соловьев****, **В.В. Хлебников*****

**Тамбовский государственный технический университет,
Россия, Тамбов, good.win32@yandex.ru,*

***Тамбовский государственный университет имени Г.Р. Державина,
Россия, Тамбов, solovjevdenis@mail.ru*

****Тамбовский государственный университет имени Г.Р. Державина,
Россия, Тамбов, khlebnikovvv@mail.ru*

Аннотация. В работе рассматривается разработка информационной системы для проектирования локальной вычислительной сети с использованием методологии функционального моделирования IDEF0. Составленная модель предназначена для более подробного отображения информационных процессов, протекающих в разрабатываемой системе.

Ключевые слова: информационная система, локальная вычислительная сеть, IDEF0.

IDEF0-MODEL DEVELOPMENT FOR THE PROCESS OF DESIGNING ARCHITECTURE AND SELECTION OF EQUIPMENT FOR A LOCAL COMPUTER NETWORK

I.A. Solovjeva*, **D.S. Solovjev****, **V.V. Khlebnikov*****

**Tambov State Technical University,
Russia, Tambov, good.win32@yandex.ru,*

***Tambov State University named after G.R. Derzhavin,
Russia, Tambov, solovjevdenis@mail.ru*

****Tambov State University named after G.R. Derzhavin,
Russia, Tambov, khlebnikovvv@mail.ru*

Abstract. The paper considers the IDEF0-model development for the process of designing architecture and selection of equipment for a local computer network. The compiled model is intended for a more detailed display of information processes occurring in the system being developed.

Keywords: information system, local area network, IDEF0.

При проектировании локальной вычислительной сети (ЛВС) возникают трудности с выбором архитектуры и оборудования сети, необходимого для ее работы [1]. В связи с этим возникает необходимость в создании и использовании информационной системы (ИС), которая помогает правильно спроектировать сеть и найти оптимальное сетевое оборудование для правильной ее работы [2].

Функциональное моделирование с использованием методологии IDEF0 отображает более подробные процессы, протекающие в разрабатываемой ИС [3]. Результатом применения методологии IDEF0 является функциональная модель верхнего уровня для ИС проектирования ЛВС, которая представлена на рисунке 1.

Разрабатываемая ИС состоит из 6 подсистем (см. рис. 2), полученных в ходе декомпозиции функциональной модели верхнего уровня:

1. Ввод данных

Подсистема отвечает за ввод данных пользователем, по результатам которых будет проектироваться сеть.

Входными данными подсистемы являются данные, предлагаемые системой по умолчанию: количество зданий; количество этажей (в каждом здании); количество комнат (на каждом этаже); количество компьютеров (в каждой комнате); серверы (количество и тип серверов); параметры сети (цена сети, скорость сети или надежность сети).

На выходе подсистема выдает обработанные входные данные.

Механизмом, осуществляющим операцию ввода, является пользователь системы.

2. Хранение временных данных

Подсистема предназначена для хранения введенных пользователем ИС данных. Данные хранятся во временной базе данных (БД).

Входными данными подсистемы является: количество зданий; количество этажей (в каждом здании); количество комнат (на каждом этаже); количество компьютеров (в каждой комнате); выбранное оборудование.

На выходе подсистема формирует временную БД в оперативной памяти ЭВМ.



Рис. 1. Функциональная модель верхнего уровня разрабатываемой ИС

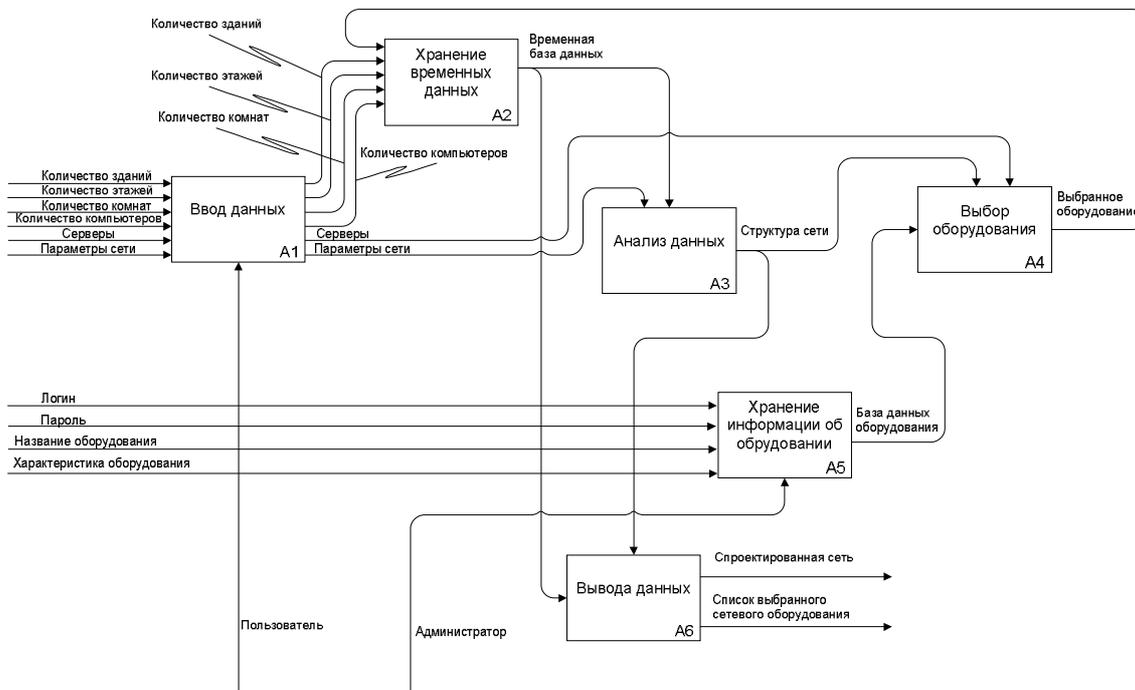


Рис. 2. Диаграмма декомпозиции функциональной модели

3. Анализ данных

Подсистема «Анализ данных», диаграмма декомпозиции которой показана на рисунке 3, отвечает за выбор архитектуры сети. В данном блоке осуществляется анализ введенных пользователем данных. На основе этих данных подсистема разбивает участки проектируемой ЛВС на сегменты и для каждого сегмента сети выбирает тип кабеля. На выходе формируется оптимальная архитектура проектируемой ЛВС, которая передается другим блокам.

4. Выбор оборудования

Подсистема отвечает за выбор сетевого оборудования, которое будет обеспечивать работоспособность проектируемой ЛВС. На вход блока подается БД оборудования. Оборудование выбирается, исходя из указанных пользователем параметров сети и выбранной подсистемой анализа данных архитектуры проектируемой ЛВС. На выходе подсистема формирует список необходимого оборудования для нормального функционирования проектируемой ЛВС. Данный список передается блоку временного хранения данных.

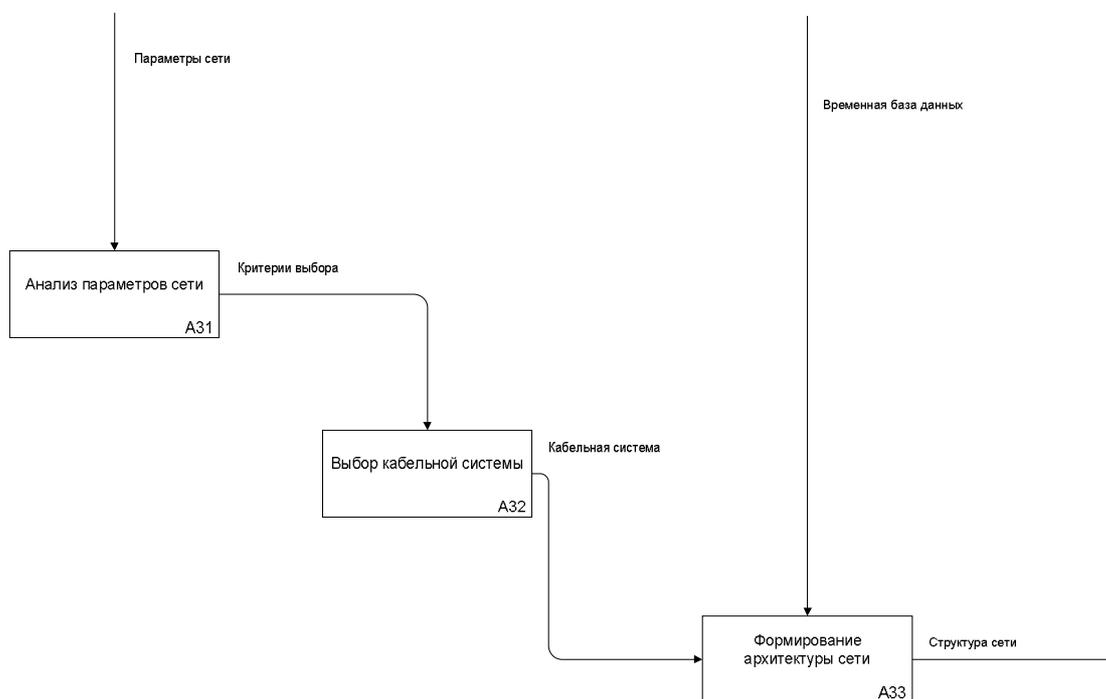


Рис. 3. Декомпозиция блока анализа данных

5. Хранение информации об оборудовании

Подсистема, диаграмма декомпозиции которой показана на рисунке. 4, отвечает за хранение информации о сетевом оборудовании. Функциями данного блока являются: предоставление доступа к БД оборудования; добавление нового и удаление выбранного оборудования из БД.

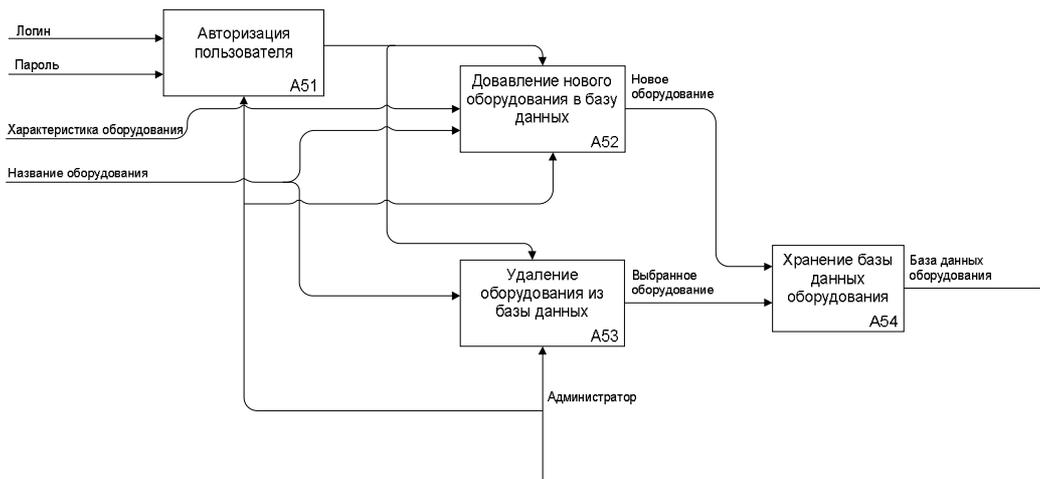


Рис. 4. Декомпозиция блока хранения информации об оборудовании

6. Вывод данных

Подсистема «Вывод данных», диаграмма декомпозиции которой показана на рисунке 5, отвечает за графическое представление проектируемой ЛВС и составлении отчета о выбранном сетевом оборудовании. На вход подается временная БД, которая содержит информацию о количестве зданий, количестве этажей, количестве комнат, количестве компьютеров и выбранном сетевом оборудовании. Данная подсистема сначала рассчитывает координаты расположения компонентов сети. Следующим шагом является отображение данных компонентов и прорисовка структурированной кабельной системы проектируемой сети.



Рис. 5. Декомпозиция блока вывода данных

Данная IDEF0-модель предназначена для разработки соответствующей ИС с целью помощи пользователям проектировании ЛВС с требуемыми характеристиками.

Библиографический список

1. Абросимов, Л.И. Базисные методы проектирования и анализа сетей ЭВМ / Л.И. Абросимов. – М.: Университетская книга, 2015. – 248 с.
2. Ширяев, В.И. Исследование операций и численные методы оптимизации / В.И. Ширяев. – М.: Ленанд, 2017. – 224 с.
3. Методология функционального моделирования IDEF0. Руководящий документ РД IDEF0-2000. – М.: Госстандарт России, 2000. – 75 с.

УДК 621.391; ГРНТИ 49.03.03

РАЗРАБОТКА ПЕННОГО ГАЗОПРОМЫВАТЕЛЯ

А.В. Крылов*

*Тамбовский государственный технический университет,
Россия, Тамбов, krylov@tagat.ru

Аннотация. Рассмотрен мокрый способ очистки технологических выбросов. Рассмотрены преимущества и недостатки пенного газопромывателя. Разработана конструкция пенного газопромывателя.

Ключевые слова: пенный газопромыватель, скруббер, абсорбер, центробежные форсунки, винтовые форсунки.

DEVELOPMENT OF A FOAM GAS WASHER

A.V. Krilov*

Tambov state technical University,
Russia, Tambov, krylov@tagat.ru

Abstract. A wet method for cleaning process emissions has been reviewed. The advantages and disadvantages of the foam gas washer are considered. The design of the foam gas washer has been developed.

Keywords: foam washer, scrubber, absorber, centrifugal nozzles, screw nozzles.

Простым и эффективным способом очистки технологических выбросов от пыли является мокрый способ очистки. Мокрое пылеулавливание основано на контакте запыленного газового потока с жидкостью, которая захватывает частицы и уносит их из аппарата в виде шлама. Процесс пылеулавливания в мокрых газоочистных аппаратах сопровождается процессами абсорбации. Поэтому многие аппараты данного типа могут применяться не только для очистки газов от пыли, но и для очистки от капель жидкости, газообразных составляющих, а также для охлаждения газов. Конденсат жидкости, содержащейся в газах, при охлаждении способствует росту эффективности работы мокрых пылеуловителей [1 -3].

В качестве орошающей жидкости в мокрых пылеулавливающих аппаратах чаще всего применяется вода, а при совместном решении вопросов пылеулавливания и химической очистки газов выбор орошающей жидкости обуславливается процессом абсорбации. Для уменьшения количества отработанной жидкости при работе мокрых аппаратов применяется ее частичная рециркуляция, а иногда и замкнутая система орошения.

Пенные пылеуловители представляют собой аппараты, корпус которых разделен решеткой с равномерно расположенными мелкими отверстиями. Запыленный поток поступает под решетку, очищенный удаляется из верхней части корпуса. Вода поступает на решетку сверху. В зависимости от конструкции пылеуловителя вода с поверхности решетки отводится через отверстия в решетке и частично через слив, либо только через отверстия. Пенные аппараты относятся к низконапорным пылеуловителям. По способу отвода жидкости с решетки их подразделяют на два основных типа: с переливными решетками и с провальными решетками.

Для расчета сечения газопромывателя нам необходимо знать: общий расход очищаемого газа, состав пыли.

Расчет площади сечения корпуса аппарата осуществляется по формуле:

$$S = V/(3600 \cdot v_r),$$

где V - общий расход воздуха, v_r - скорость газа в сечении аппарата из условий устойчивости слоя пены и равна 3 м/с.

Расчет диаметра корпуса аппарата для газопромывателя круглого сечения осуществляется по формуле:

$$D = (4S/\pi)^{0.5},$$

Расчет размера стенки корпуса аппарата для газопромывателя квадратного сечения осуществляется по формуле:

$$A = S^{0.5}.$$

Газопромыватели бывают двух типов сечений: круглый и квадратный. Мы выбираем квадратный, он дает более равномерное распределение загрязненного потока по сечению. Все эти расчеты производим с использованием программы (см. рис. 1).

<u>Расчет сечения газопромывателя</u>	
Введите размер расчетного воздуха, Q=мЗ/ч	3400
Состав очищаемого воздуха	кислотощелочная
Форма сечения газопромывателя	квадратный
Размер сечения корпуса аппарата, м	0,56
Оптимальный расход жидкости, мЗ/ч	0,2
Действительная высота слоя пены аппарата, мм	333,7

Рис. 1. Интерфейс программы

По результатам расчета получаем сечение газопромывателя, и переходим к разработке корпуса газопромывателя.

Основной несущей конструкцией пылеуловителя является корпус. При использовании металлической конструкции необходимо было ее обезопасить в плане коррозионной стойкости. Для этого корпус футеровали пвх2, что усложняло и удорожало производство. Наилучшим решением будет изготовление корпуса из полимерного материала. В зависимости от среды можно подобрать материал: полипропилен для кислото-щелочных сред, поливинилденфторид, поливинилхлорид, полиэтилен для концентрированных хромсодержащих сред. Мы выбираем полипропилен, как более подходящий для наших условий [4 -5].

Полипропилен (ПП) был открыт в 1954 году и его популярность выросла очень быстро из-за способности сохранять повышенную прочность и жесткость при воздействии относительно высокой температуры в течение длительного времени.

Полипропилен существует в пяти основных вариациях таких как: гомополимеры, сополимеры, статистические сополимеры, модифицированные каучуком смеси и специальные сополимеры. Свойства полипропилена зависят от молекулярной массы, способа производства, и участия сополимеров. Полипропилен является одним из самых легких материалов среди пластмасс с плотностью 0,905 г/см³.

Полипропилен имеет ряд преимуществ перед металлом:

- обработка деталей, метод сваривания (экструзионный, газовый метод сварки);
- стойкость полипропилена к агрессивным средам внутри установки по сравнению с металлом; легко очищается и на нем не скапливается пыль и грязь (за счет этого уменьшается периодичность технических осмотров).

- имеет незначительный вес;

- не вступает в реакцию с большинством сред.

Однако полипропилен имеет некоторые недостатки:

- низкая вязкость при низких (минусовых) температурах, приводящая к разрушению при механических воздействиях;

- легковоспламеняющийся;

- чувствительность к ультрафиолету.

На основании вышеизложенного разрабатываем корпус из полипропилена проводим расчеты на прочность стенок, усиленных поясом жесткости .

Расчитываем гидростатическое давление по формуле:

$$P = P_c + \rho g H_{\text{ж}} \times 10^{-6} \text{ МН/м}^2,$$

где P_c = 0 избыточное давление; $g = 9,81 \text{ м/с}^2$; $H_{\text{ж}}$ -высота столба жидкости, ρ -плотность раствора.

Положение по вертикали горизонтальных ребер определяем по формуле:

$$H_1 = 0,23 H_{\text{ж}},$$

для одного ребра примеру.

Расчет давления на стенку корпуса установки осуществляем по формуле:

$$P_1 = P.$$

Расчет толщины стенки без укрепления расчитываем по формуле:

$$S = K * H_1 * (P_1 / \beta_{\text{н.д}}),$$

где K -коэффициент зависящий от способа закрепления стенки, $\beta_{\text{н.д}} = \sigma_T / n_T$. σ_T -предел текучести, n_T -запас прочности.

С учетом округления принимаем толщину стенки.

Все эти расчеты производим с использованием программы (см. рис. 2).

Высота столба жидкости, м	0,3
Количество ребер жесткости	два
Положение по вертикали горизонтального ребра, H1=м	0,0690
Положение по вертикали горизонтального ребра, H2=м	0,1740
Положение по вертикали горизонтального ребра, H3=м	
Гидростатическое давление в нижней части каждого из элементов, p1= Мн/м2	0,0029
Гидростатическое давление в нижней части каждого из элементов, p2= Мн/м2	0,0023
Гидростатическое давление в нижней части каждого из элементов, p3= Мн/м2	0,0012
Гидростатическое давление в нижней части каждого из элементов, p4= Мн/м2	
Минимальная расчетная толщина элементов стенки, s1= мм	0,9
Минимальная расчетная толщина элементов стенки, s2= мм	1,2
Минимальная расчетная толщина элементов стенки, s3= мм	1,0
Минимальная расчетная толщина элементов стенки, s4= мм	
Расчетный момент сопротивления горизонтального ребра ванны W= м3	0,000000001

Рис. 2. Интерфейс программы расчета

По результатам расчета изготавливаем корпус для пенного газопромывателя и проводим гидравлические испытания, т.е. заполняем корпус водой в количестве 300мм от дна. Испытания являются успешными, если деформация стенки после заполнения корпуса водой остается в допускаемых пределах. Корпус удовлетворяет условиям гидравлических испытаний и считаем модернизацию успешной

Одним из недостатков пенных газопромывателей является работа в определенном режиме скорости газа. Устранения этого недостатка возможно за счет внедрения дополнительного орошения в пенном газопромывателе. Запыленный поток будем пропускать через завесу распыляемой жидкости. При этом частицы пыли будут захватываться каплями промывочной жидкости, а очищенный поток удаляется из верхней части корпуса [6].

Для дополнительного орошения будем использовать центробежные форсунки грубого распыла работающие под давлением 0.3...0.4 Мпа и создающие капли требуемого размера. Применение таких форсунок позволяет работать на оборотной воде, содержащей взвеси. Из центробежных форсунок для наших условий подходят спиральные форсунки. Спиральные форсунки – простые, необычные и надежные распылители. Их еще называют пиг тейл (pig tail), поросячий хвост, винтовые форсунки и т.п. Форсунки со спиральным соплом, это универсальный разбрызгиватель для жидкостей, содержащих взвешенную фазу. К ключевым особенностям спиральных форсунок относят широкий и равномерный угол распыла (полный или полый конус), эффективное распыление вязких жидкостей и самое главное – абсолютная устойчивость к засорению. Конструкция формы распыла данных форсунок представляет собой сплошную спираль жидкости, вращающуюся внутри конического пространства.

Высокая устойчивость к закупоркам, делает эти распылители крайне востребованными, когда стоит задача мелко и равномерно распылить большой объем жидкости, в составе которой присутствуют взвешенные частицы (загрязнения). Также данные качества незаменимы в условиях, когда безопасность или надежность системы выходят на первый план.

Во первых определяемся со схемой орошения (см. рис.3). Прямоток, где вода подается по направлению потока воздуха или противоток, где вода подается против направления. У противотока более высокое сопротивление и выше степень очистки, у прямотока наоборот ниже сопротивление и меньше степень очистки.

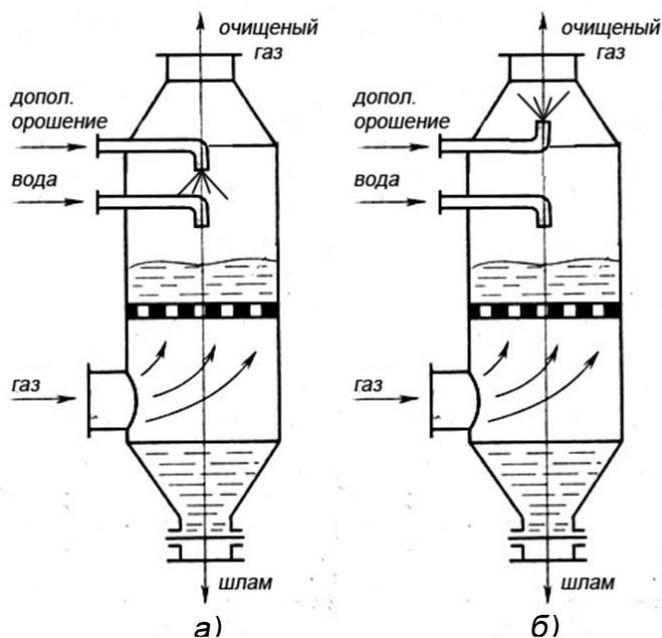


Рис. 3. Схемы орошения: а) противоток, б) прямоток

Определяем схему расположения форсунок. Для этого используем solidworks. Моделируем распыл форсунок в программе (см. рис. 4) и подбираем схему орошения, с наибольшим охватом сечения абсорбера.

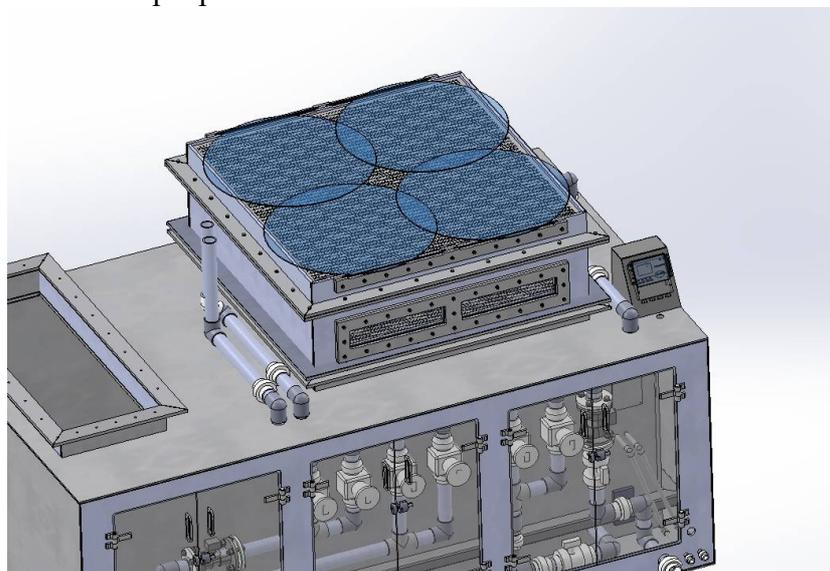


Рис. 4. Иллюстрация подбора оптимальной схемы орошения

Дополнительное орошение проектируем таким образом, чтобы можно было поворотом форсунки на 180 градусов, поменять режим орошения.

Проводим испытания работы абсорбера. Контроль процесса будем вести через смотровое окно. Сначала проводим испытания со схемой противоток и фиксируем результаты. Далее поворачиваем форсунки на 180 градусов и проводим испытания со схемой прямоток. В противотоке мы наблюдаем более равномерное распределение жидкости по сечению аппарата, что означает наибольший контакт жидкости с запыленным потоком и наилучшую очистку. В прямотоке наблюдаем худшие результаты.

В результате опытных испытаний противоток принимаем, как наиболее оптимальную схему для работы. Сами испытания считаем успешными, так как смогли добиться максимального контакта жидкости с запыленным потоком воздуха.

В результате работы был спроектирован и изготовлен газопромыватель. В результате пусконаладочных испытаний выбросы после газопромывателя соответствуют нормам предельно допустимых концентраций.

Библиографический список

1. Мокрозуб В.Г., Немтинов В.А., Егоров С.Я. Информационно-логические модели технических объектов и их представление в информационных системах - Информационные технологии в проектировании и производстве. 2010. №3. С. 68-73.
2. Борисенко А.Б., Карпушкин С.В. Иерархия задач аппаратурного оформления технологических систем многоассортиментных химических производств - Известия Российской академии наук. Теория и системы управления. 2014. № 3. С. 113 – 123.
3. Nemtinov V., Bolshakov N., Nemtinova Y. Automation of the early stages of plating lines design. - MATEC Web of Conferences. 2017. V. 129, № 7. 01012.
4. Nemtinov V.A., Nemtinova Yu.V. On an approach to designing a decision making system for state environmental examination - Journal of Computer and Systems Sciences International. 2005. 44(3). pp. 389-398..
5. Mokrozub V.G., Nemtinov V.A. An Approach to Smart Information Support of Decision-Making in the Design of Chemical Equipment - Chemical and petroleum engineering. 2015. 51(7-8). pp. 31-35.
6. Nemtinov V.A., Nemtinova Y.V. An approach to regulating the relationships between environmental users - Journal of Computer and Systems Sciences International. 2004. 43(5). pp. 800-804.

УДК 004.89; ГРНТИ 50.07.03

МОДЕЛИРОВАНИЕ РАСКРАШЕННЫХ СЕТЕЙ ПЕТРИ С ИСПОЛЬЗОВАНИЕМ ЯЗЫКА SPARQL

В.Ю. Каев, В.Н. Дубинин

*Пензенский государственный университет,
Российская Федерация, Пенза, veld4n@gmail.com*

Аннотация. В данной работе предлагается метод моделирования раскрашенных сетей Петри на основе онтологий и языка SPARQL. Рассматривается онтология раскрашенных сетей Петри, SPARQL-запросы, моделирующие динамику работы, а также вопросы построения приложений для интерпретации сетевых моделей.

Ключевые слова: моделирование, онтология, раскрашенные сети Петри, динамика, SPARQL.

MODELLING OF COLORED PETRI NETS USING SPARQL LANGUAGE

V.I. Kaev, V.N. Dubinin

*Penza State University,
Russia, Penza, veld4n@gmail.com*

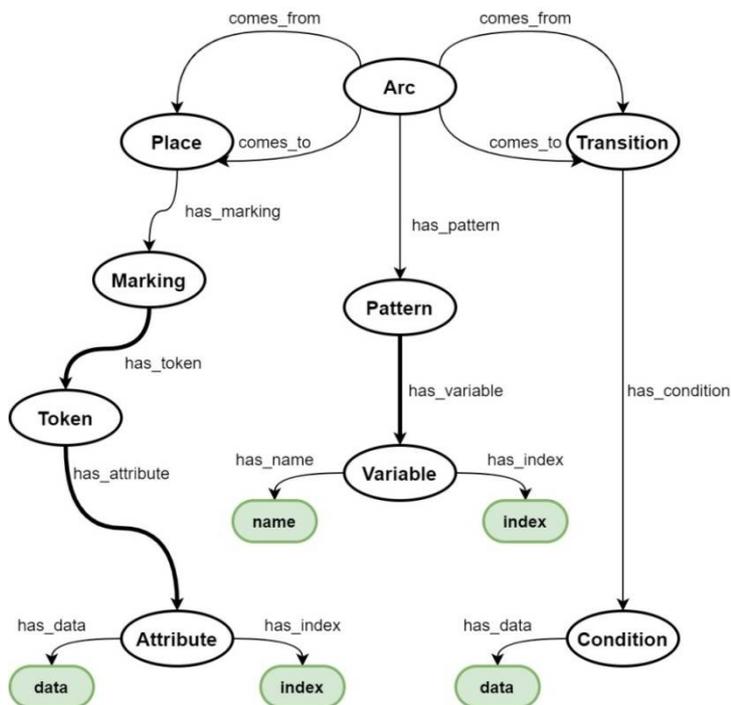
Abstract. This paper proposes a method for colored Petri nets modelling based on ontologies and the SPARQL language. The ontology of colored of Petri nets, SPARQL queries for modelling the dynamics of functioning, as well as issues of building applications to simulate network models are considered.

Keywords: modelling, ontology, coloured Petri nets, dynamics, SPARQL.

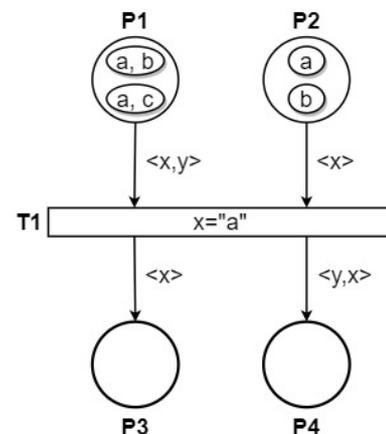
Раскрашенные сети Петри (РСП) широко используются как средство моделирования систем, в которых понятия параллельности и событийного взаимодействия являются ключевыми [1]. Данный класс сетей Петри отличает наличие свойств, присущих высокоуровневым языкам программирования, при сравнительно компактной форме параметризуемых моделей. Это, в частности, делает их подходящими для использования при моделировании, проектировании и реализации распределенных вычислительных, управляющих и коммуникационных систем в промышленном масштабе. Для представления знаний в рамках определенной предметной области, допускающего применение логики, а также позволяющего использовать эти знания повторно, используются онтологии [2]. Они являются основой развивающейся концепции семантического web, которая расширяет возможности машинной интерпретации данных в сети. Онтологическое представление позволяет определить модель РСП в декларативном формате, что упрощает процесс интеграции с другими моделями и приложениями семантической сети. Одним из стандартов семантического web, рекомендуемых консорциумом W3C, является язык запросов SPARQL [3], использующийся для работы с данными в формате RDF. Он предоставляет широкие возможности по извлечению онтологических описаний в форме триплетов и манипуляции ими, что делает его пригодным для моделирования динамических систем.

В данной работе рассматривается моделирование РСП с использованием онтологий и SPARQL-запросов. Используемая онтология РСП на уровне концепций приведена на рисунке 1,а. Простой пример РСП представлен на рисунке 1,б. Онтология РСП включает в себя: классы Place (Позиция), Arc (Дуга), Transition (Переход), Marking (Маркировка позиции), Token (Метка), Attribute (Атрибут), Pattern (Шаблон дуги), Variable (Переменная шаблона) и Condition (Сторожевое условие перехода); объектные свойства: comes_from и comes_to (для входного и выходного узла дуги), has_marking (для маркировки позиции), has_token (для меток), has_attribute (для атрибутов меток), has_pattern (для шаблона), has_variable (для переменных шаблона), has_condition (для условия перехода); свойства по данным: has_data (для данных маркера и условия перехода), has_index (для определения позиции атрибута/переменной в кортеже) и has_name (для задания имени переменной). Атрибуты меток и

переменные шаблонов упорядочены в кортеж. Данная онтология была сформирована в редакторе онтологий Protégé [4].



a)



б)

Рис. 1. Представления раскрашенных сетей Петри: а – онтология на уровне TBox; б – графическое представление (пример)

Сам процесс моделирования представляет собой симуляцию срабатываний переходов посредством выполнения цепочек SPARQL-запросов (транзакций) к онтологии. Фрагменты этих запросов приводятся ниже.

Первый запрос позволяет получить данные по переходам PCП.

```

PREFIX : <http://www.semanticweb.org/baker/ontologies/2020/9/OntoNet-CPN-onlotogy#>
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
SELECT ?transition ?condition_data ?variables
WHERE {
  ?transition rdf:type :Transition.
  ?transition :has_condition ?condition.
  ?condition :has_data ?condition_data
  {
    SELECT ?transition (GROUP_CONCAT(?var_i_name; SEPARATOR=",") as ?variables)
    WHERE {{
      SELECT DISTINCT ?transition ?var_i_name
      WHERE {
        ?arc_i :comes_to ?transition;
          :has_pattern ?patt_i.
        ?patt_i :has_variable ?var_i.
        ?var_i :has_name ?var_i_name.
      }
    }}
  }
  GROUP BY ?transition
}

```

В результате выполнения данного запроса получаем URI переходов, их сторожевые условия на языке SPARQL и список переменных, использующихся в шаблонах входных дуг каждого перехода.

С информацией по переходам становится возможным определение факта разрешенности перехода при определенных комбинациях меток во входных позициях. Общий вид SPARQL-запроса, отвечающего за это, приведен ниже.

```
PREFIX : <http://www.semanticweb.org/baker/ontologies/2020/9/OntoNet-CPN-ontology#>
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
SELECT ?place_i ?token_i ?values <значения переменных...>
WHERE {
  <выборка значений переменных...>
  BIND((<сторожевое условие перехода>) as ?res)
  ?arc_i :comes_to <URI перехода>;
         :comes_from ?place_i;
         :has_pattern ?patt_i.
  ?place_i :has_marking ?marking_i.
  ?marking_i :has_token ?token_i.
  <выделение значений переменных, сгруппированных по меткам>
  <проверка корректного сопоставления значения переменной для всех входных дуг...>
  BIND((<конкатенация значений переменных в строку через запятую>) as ?values)
}
GROUP BY ?place_i ?token_i ?res ?values <значения переменных...>
HAVING (?res = true)
ORDER BY ?values
```

Данный запрос определяет комбинации меток, при которых разрешен переход, а также осуществляет согласование переменных шаблонов входных дуг и значений атрибутов меток из входных позиций. Здесь и далее в угловые скобки заключены части запроса, формируемые динамически на основе данных из предыдущего запроса: идентификатора перехода, сторожевого условия перехода и списка переменных.

Рассмотрим составляющие этого запроса в отдельности на примере простой РСП, приведенной выше. Поиск значений осуществляем для перехода $t1$. Выборка значения и индекса переменной x для нее будет выглядеть следующим образом.

```
{SELECT DISTINCT ?x ?x_index
WHERE {
  ?arc_i :comes_to :t1;
         :comes_from ?place_i;
         :has_pattern ?patt_i.
  ?place_i :has_marking ?marking_i.
  ?marking_i :has_token ?token_i.
  ?token_i :has_attribute ?attr_i.
  ?attr_i :has_data ?x;
          :has_index ?x_index.
  ?patt_i :has_variable ?var_i.
  ?var_i :has_name ?var_i_name;
         :has_index ?var_i_index.
  FILTER(?var_i_name = "x" && ?x_index = ?var_i_index)
}}
```

В результате получаем все возможные значения переменной x , а также их индексы в кортежах атрибутов в метках входных позиций. Выделение известных значений и индексов переменных x и y , сгруппированных по меткам, представлено ниже.

```
{
  SELECT ?token_i ?patt_i
         (COUNT(?attr_i) as ?attr_i_count) (COUNT(?var_i) as ?var_i_count)
         ?x ?x_index ?y ?y_index
```

```

WHERE {
  ?patt_i :has_variable ?var_i.
  ?token_i :has_attribute ?attr_i.
  OPTIONAL {
    SELECT ?token_i ?x ?x_index
    WHERE {
      ?token_i :has_attribute ?attr_i.
      ?attr_i :has_data ?x;
              :has_index ?x_index.
    }
  }
  FILTER(NOT EXISTS {
    ?token_i :has_attribute ?attrx_i.
    ?attrx_i :has_data ?_var;
             :has_index ?_var_index.
    FILTER ((?_var != ?x || ?_var_index != ?x_index) && (?_var != ?y || ?_var_index != ?y_index))
  })
}
GROUP BY ?token_i ?patt_i ?x ?x_index ?y ?y_index
HAVING (?attr_i_count = ?var_i_count)
}

```

Данный фрагмент запроса ограничивает комбинации значений переменных по меткам, используя ранее извлеченные данные об их индексах в пределах кортежей атрибутов. Проверка корректного сопоставления значения переменной x для всех входных дуг приведена далее.

```

FILTER (!BOUND(?x) || NOT EXISTS {
  SELECT ?place2_i ?x (COUNT(?attr2_i_data) as ?x2_count)
  WHERE {
    ?arc2_i :comes_to :t1;
            :comes_from ?place2_i;
            :has_pattern ?patt2_i.
    ?place2_i :has_marking ?marking2_i.
    ?marking2_i :has_token ?token2_i.
    ?token2_i :has_attribute ?attr2_i.
    ?attr2_i :has_index ?attr2_i_index.
    ?patt2_i :has_variable ?var2_i.
    ?var2_i :has_name ?var2_i_name;
            :has_index ?var2_i_index.
    FILTER(?var2_i_name = "x" && ?attr2_i_index = ?var2_i_index)
  }
  OPTIONAL {
    ?attr2_i :has_data ?attr2_i_data;
            FILTER(?attr2_i_data = ?x)
  }
}
}
GROUP BY ?place2_i ?x
HAVING (?x2_count = 0)
})

```

В данном случае проверяется факт наличия аналогичного значения переменной x в других входных позициях, где шаблон дуги содержит эту переменную.

Переход не разрешен, если для него не существует ни одного варианта сопоставления значений переменным шаблонов входных дуг. Общий вид SPARQL-запроса, выполняющего разрешенный переход, представлен ниже.

```

DELETE {
  ?marking_i :has_token ?token_i.
  ?token_i rdf:type :Token;

```

```

      :has_attribute ?token_i_attr.
    ?token_i_attr rdf:type :Attribute;
      :has_data ?token_i_attr_data;
      :has_index ?token_i_attr_index.
  }
  INSERT {
    ?marking_o :has_token ?token_new.
    <формирование меток в выходных позициях>
  }
  WHERE {
    {
      VALUES ?token_i { <URI меток, проходящих через переход...> } # <-- :token1-1 :token2-1
      ?arc_i :comes_to <URI перехода>;
        :comes_from ?place_i;
        :has_pattern ?patt_i.
      ?place_i :has_marking ?marking_i.
      ?marking_i :has_token ?token_i.
      ?token_i :has_attribute ?token_i_attr.
      ?token_i_attr :has_data ?token_i_attr_data;
        :has_index ?token_i_attr_index.
    } UNION {
      ?arc_o :comes_from <URI перехода>;
        :comes_to ?place_o;
        :has_pattern ?patt_o.
      ?place_o :has_marking ?marking_o.
      BIND(UUID() as ?token_new)
      <выборка атрибутов по шаблонам выходных дуг...>
    }
  }
}

```

Данный запрос удаляет метки, определенные в ходе выполнения предыдущего запроса, из выходных позиций, и формирует метки в выходных позициях на основе шаблонов выходных дуг. Идентификаторы новых меток формируются с помощью функции UUID.

Аналогично, рассмотрим составляющие этого запроса. Фрагмент запроса для формирования данных по новым меткам выходных позиций при наличии сопоставленных переменным x и y значений (“a” для x и “b” для y) приведен ниже:

```

?token_new rdf:type :Token;
  :has_attribute ?x;
  :has_attribute ?y.
?x rdf:type :Attribute;
  :has_data "a";
  :has_index ?x_index.
?y rdf:type :Attribute;
  :has_data "b";
  :has_index ?y_index.

```

Здесь в онтологию вставляются (insert) данные о новых метках и их атрибутах. Выборка атрибутов для переменной x производится следующим образом.

```

OPTIONAL {
  SELECT ?patt_o ?x ?x_index
  WHERE {
    ?patt_o :has_variable ?var_o.
    ?var_o :has_name ?var_o_name;
      :has_index ?x_index.
    FILTER(?var_o_name = "x")
    BIND(UUID() as ?x)
  }
}

```

В данном фрагменте, исходя из шаблонов выходных дуг, выбираются сформированные идентификаторы атрибутов, а также индексы для будущих кортежей в составе меток.

Для получения маркировки РСП на текущем шаге моделирования используется следующий SPARQL-запрос.

```
SELECT ?place ?token (GROUP_CONCAT(?data;SEPARATOR=",") AS ?token_data)
WHERE {
  ?place rdf:type :Place.
  ?place :has_marking ?marking.
  ?marking :has_token ?token.
  OPTIONAL {
    ?token :has_attribute ?attr.
    {SELECT ?attr ?data
     WHERE {
       ?attr :has_data ?data.
       ?attr :has_index ?index.
     } ORDER BY ?index
    }
  }
}
GROUP BY ?place ?token
ORDER BY ?place
```

Он позволяет получить позиции РСП и метки, содержащиеся в них.

На практике данный подход был применен при разработке клиент-серверной системы интерпретации РСП под названием *OntoNet*. Она позволяет пользователю загружать на сервер онтологию РСП и моделировать ее выполнение в ручном режиме. В качестве сервера используется *Apache Jena Fuseki Server* [5], взаимодействие организовано по протоколу HTTP. Клиент был разработан на языках *JavaScript/TypeScript*. Он спроектирован с использованием модульной архитектуры, поэтому отдельные его модули могут использоваться в других проектах. Каждый модуль имеет свой интерфейс взаимодействия, предоставляющий широкий диапазон методов для получения состояния сети, определения разрешенных переходов, пошагового моделирования и т.д. Состояние РСП выводится в виде таблицы, отражающей маркировку позиций, однако представление отделено от логики системы и может быть заменено на более продвинутое или специализированное.

Направлением дальнейших исследований является введение модульности и иерархии в «онтологические» РСП, развитие инструментальной системы *OntoNet* для возможности реализации на ее основе распределенных вычислительных и управляющих систем в контексте семантического web. Перспективным является использование *мультимодельного* подхода к построению и реализации сложных систем (например, информационных моделей киберфизических систем), когда одна часть системы представляется с использованием одной модели (например, тех же РСП), другая часть – с использованием другой модели (например, функциональных блоков стандарта IEC 61499), и эти модели интегрируются на базе онтологий.

Библиографический список

1. Jensen K., Kristensen L.M. Coloured Petri Nets. Modelling and Validation of Concurrent Systems // Springer-Verlag Berlin Heidelberg, 2009. – 384 с.
2. Jacksi K. Introduction to Semantic Web. // University of Zakho Department of Computer Science March, 2014.
3. SPARQL 1.1 Update. W3C Recommendation 21 March 2013. [Электронный ресурс]. – Режим доступа: <https://www.w3.org/TR/2013/REC-sparql11-update-20130321/>. – Дата доступа: 05.12.2020.
4. Protégé [Электронный ресурс]. Режим доступа: <http://protege.stanford.edu/>. – Дата доступа: 17.10.2020.
5. Apache Jena Fuseki [Электронный ресурс]. – Режим доступа: <https://jena.apache.org/documentation/fuseki2/>. – Дата доступа: 17.12.2020.

УДК 004.4; ГРНТИ 50.41.25

РАБОТА С ФРЕЙМВОРКОМ SLATE ДЛЯ НАПИСАНИЯ БОГАТЫХ ТЕКСТОВЫХ РЕДАКТОРОВ

А.П. Сальников

Рязанский государственный радиотехнический университет имени В.Ф. Уткина,
Российская Федерация, Рязань, artem7753@mail.ru

Аннотация. В работе рассматриваются базовые принципы работы богатых текстовых редакторов, основные возможности фреймворка SlateJS.

Ключевые слова: богатый текстовый редактор (RichTextEditor, RTE)

WORKING WITH THE SLATE FRAMEWORK FOR WRITING RICH TEXT EDITORS

A.P. Salnikov

Ryazan State Radio Engineering University named after V.F. Utkin,
Russia, Ryazan, artem7753@mail.ru

The summary. The paper discusses the basic principles of rich text editors, the main features of the slate js framework.

Keywords: RichTextEditor, RTE

RichTextEditor (RTE) –при дословном переводе на русский язык означает богатый текстовый редактор. Это отличное решение, которое позволяет пользователю редактировать HTML-разметку, изменяя только текстовые свойства внутри редактора [1].

SlateJS – полностью настраиваемый фреймворк для создания богатых текстовых редакторов. Он позволяет создавать многофункциональные и интуитивно понятные редакторы, подобные используемым в Medium, DropboxPaper или GoogleDocs. Данный фреймворк является полностью модульным, то есть вся логика реализована с помощью плагинов, а не содержится в ядре, так что функционал разрабатываемого редактора ограничен только фантазией программиста.

Если погрузиться вглубь документации и изучить, каким же образом работает сам фреймворк, то можно убедиться, что в основе работы лежит довольно несложная технология. Каждая веб-страница, которая загружается в браузер, имеет свой собственный объект `document`, и у этого объекта есть свойство `designMode`. Оно существует еще с конца 90-х годов. В первый раз оно было реализовано в первых версиях InternetExplorer [3].

```
document.designMode = "on"
```

Если включить это свойство, то есть присвоить ему значение `on`, то вся страница, все содержимое `body` становится редактируемым. Можно поставить курсор в любое место страницы и отредактировать текст. Но этот способ не идеален. Что делать, если не нужно редактировать всю страницу, а только какой-то определенный блок?

Contenteditable –основной браузерный API для создания редактируемых блоков. При переключении этого атрибута в значение `true` все, что находится внутри блока, становится редактируемым [2]. Именно таким образом в последнее время функционирует большинство визуальных редакторов.

Document.execCommand –специальный метод, который предоставляет команды для работы с контентом в редактируемой области. Таких команд фиксированное количество, например, `bold`, `fontSize`, `formatText` и так далее.

В SlateJS для реализации стилизации текста существует несколько типов: `mark`, `inline` и `block`. Отличаются они, по сути, только тем, что элемент будет позиционироваться, как строка, как блок-строка или как блок. Все основные команды для работы с редактором находятся в объекте `editor`. Основные команды это:

- `insertBlock(block)` – вставляет блок в текущее место. Так же есть большое количество команд для вставки по ключу блока, при выделении;
- `wrapBlock (block)` – оборачивает выделенный текст в блок определенного типа;
- `addMark(mark)` – оборачивает выделенный текст в марку определенного типа;
- `toggleMark(mark)` – переключает в выделенном тексте марку определенного типа, то есть если марка есть – выключить, если нет – включить;
- `wrapInline(inline)` – обернуть выделенный текст в блок строчного типа ;
- `unwrapBlock(block)` – удалить блочную обертку из выделенного текста;
- `removeMark(mark)` – удалить марку с выделенного текста;
- `setBlock(block)` – текущая команда не добавляет новый блок в текущее выделение, а заменяет его на выбранный.

После того как мы добавили блок, нам необходимо понять, как его отобразить, для этого существуют методы `renderMark`, `renderBlock` и `renderInline`. В них мы смотрим, какой тип блока к нам пришел, и отображаем в соответствии ему определенную обертку для выделенного текста.

```
const renderMark = (props: RenderMarkProps, editor: CoreEditor, next: () => any) => {
  switch (props.mark.type) {
    case 'bold':
      re-
return <strong { ...props.attributes }>{ props.children }</strong>;
    default:
      return next();
  }
};

const renderBlock = (props: RenderBlockProps, editor: CoreEditor, next: () => any) => {
  switch (props.node.type) {
    case 'header-1':
      return <h1 { ...props.attributes }>{ props.children }</h1>;
    default:
      return next();
  }
};
```

Так же мы можем обрабатывать нажатия клавиатурных клавиш, если добавить в плагин метод `onKeyDown`. Таким образом мы можем проверить, нажата ли определенная клавиша, и, если да, то добавить выделенному тексту определенную марку или тип блока. Функция `next`, которая приходит в параметры, необходима для того чтобы фреймворк не остановился в текущем плагине, а перешел к следующему.

```
const onKeyDown = (event: KeyboardEvent, editor: CoreEditor, next: () => any) => {
    if (event.ctrlKey && event.keyCode === 66) {
        return editor.toggleMark('bold');
    }
    return next();
};
```

При создании объекта редактора, в параметры ему мы передаем массив с используемыми плагинами, они могут быть как самописные, так и установленные извне, как показано в примере ниже.

```
const PLUGINS: any[] = [baseMarksPlugin(), Lists(), headerPlugin()];
```

Так же в SlateJS есть возможность десериализовывать текст, который вставляется в редактор. Данная возможность позволяет сохранить верстку в скопированном тексте, такие как заголовки, списки, таблицы и многое другое. Функция десериализации проверяет при вставке теги в скопированном тексте и создают уже в редакторе соответствующие блоки с определенными типами.

```
const BLOCK_TAGS: any = {
    p: 'paragraph',
    h1: 'header-1',
};
const RULES = [{
    deserialize(el: any, next: any) {
        const block = BLOCK_TAGS[el.tagName.toLowerCase()];
        if (block) {
            return {
                object: 'block',
                type: block,
                nodes: next(el.childNodes),
            };
        }
    }
}];
export const serializer = new Html({ rules: RULES });
```

Библиографический список

1. Flex 3 Cookbook: Code-Recipes, Tips, and Tricks for RIA Developers. Joshua Noble, Todd Anderson. "O'Reilly Media, Inc.", 13 мая 2008 г. – 704с.
2. Хоган Брайан. HTML5 и CSS3. Веб-разработка по стандартам нового поколения. Издательский дом "Питер", 9 нояб. 2011 г. - 272с.
3. Современные технологии в науке и образовании. СТНО 2020. "ПРИМЕНЕНИЕ ТЕКСТОВОГО РЕДАКТОРА ДЛЯ СИСТЕМЫ КОНФИГУРАЦИИ ИНФОРМАЦИОННЫХ САЙТОВ НА БАЗЕ ФРЕЙМВОРКА SLATE", Сальников А.П. . С.31-34.

УДК 62-529; ГРНТИ 55.30.51

МОБИЛЬНАЯ ПОЖАРНАЯ РОБОТОТЕХНИКА КАК НАПРАВЛЕНИЕ ВНЕДРЕНИЯ ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ В МЧС РОССИИ

Е.Ю. Огурцова*, Р.Н. Фадеев**

**Ивановский государственный университет, Шуйский филиал,
Россия, Шуя, ogurcova-elena@mail.ru,*

***Владимирский государственный университет
имени Александра Григорьевича и Николая Григорьевича Столетовых,
Россия, Владимир, fadeevroman.shua@gmail.com*

Аннотация. В данной статье рассматриваются перспективы применения мобильной пожарной робототехники в России. Анализируются разработки пожарных робототехнических комплексов, сравниваются подходы к определению роботизированного средства пожаротушения.

Ключевые слова: роботизированный комплекс, система управления робототехнического средства, беспилотные летательные аппараты.

MOBILE FIRE ROBOTICS AS A DIRECTION OF INTRODUCTION OF INFORMATION TECHNOLOGIES IN EMERCOM OF RUSSIA

E. Yu. Ogurtsova*, R.N. Fadeev**

**Ivanovo State University, Shuya branch,
Russia, Shuya, ogurcova-elena@mail.ru,*

***Vladimir State University named after Alexander Grigorievich and Nikolai Grigorievich Stoletov,
Russia, Vladimir, fadeevroman.shua@gmail.com*

Abstract. This article discusses the prospects for the application of mobile fire robotics in Russia. The engineering of fire-fighting robotic systems is analyzed; approaches to the definition of a robotic fire extinguishing system are compared.

Keywords: robotic complex, robotics control system, unmanned aerial vehicles.

Одним из направлений работы МЧС России является внедрение передовых технологий информационного обеспечения, современных технологий мониторинга, автоматизации деятельности подразделений Федеральной противопожарной службы, использование робототехнических комплексов.

Коллегия МЧС России утвердила Концепцию развития робототехнических систем специального назначения в системе МЧС России до 2030 года. В своем докладе начальник научно-технического управления МЧС России Александр Овсянник отметил, что на балансе подразделений МЧС по всей стране уже есть такие системы, но есть необходимость совершенствования государственной системы стандартов и требований к элементной базе и робототехническим комплексам в целом, а также недостаточная проработка "жизненного" цикла робототехники от их создания, эксплуатации и до утилизации. Еще одно важное направление развития – совершенствование пожарно-спасательных технологий, основанных на применении робототехнических комплексов с адаптивными системами искусственного интеллекта, работающих в единой информационно-управляющей сети.

Работы по созданию многофункциональных роботизированных комплексов, способных автономно либо на дистанционном управлении работать в жестких условиях эксплуатации, активизировались в нашей стране в связи со всплеском техногенных катастроф в 1980-х годах, и, прежде всего, после Чернобыльских событий. Принимая во внимание необходимость тушения пожаров на радиоактивно зараженной местности, на предприятиях химической промышленности, складах взрывчатых веществ и материалов в условиях сильного задымления и загазованности атмосферы сильнодействующими ядовитыми веществами, впервые в отечественной и международной практике в 1986-1988 годах был создан самоходный лафетный ствол СЛС-100-54 «Сойка». Имея возможность дистанционного управления по радиоканалу или по проводам, этот мобильный робототехнический комплекс (РТК) сегодня

считается прародителем всех пожарных роботов в России и определяет их наследственные характеристики – прежде всего, тяжелый и сверхтяжелый класс (вес – десятки тонн), проходимое гусеничное шасси, повышенную защиту, максимально раздвигающую потенциальные границы использования.

Трагический опыт ликвидации техногенных катастроф определил лидерство России в сфере разработки пожарных робототехнических комплексов, которые тяготеют по назначению к тяжелым технолого-разведывательным роботам (ТР), в то время как остальной мир в большей степени ограничивается более лёгкими классами разведывательно-технологических роботов (РТ). Тяжелые комплексы применяются при тушении лесных пожаров, складов боеприпасов, опасных производств, в том числе на нефтегазовых объектах и т.д. Пожарные роботы более легкого класса, как правило, также на гусеничном шасси – довольно маневренны и компактны по размерам, что дает им возможность проезжать в дверные проемы и двигаться по наклонным лестничным пролетам зданий. Имея на борту лафетный ствол с подключением через рукава к внешним источникам воды, а также устройства орошения водой («капельное облако» для собственной защиты от высокой температуры), они способны направлять струю воды или воздушно-механической пены в очаги возгораний на дальность в десятки метров. Их использование оправдано в условиях возникновения чрезмерного риска для пожарных расчетов, из недавних случаев работу пожарных роботов можно было наблюдать, в частности, при тушении пожара в парижском соборе Нотр-Дам (использовались французские пожарные роботы Colossus) [4].

Робототехнический комплекс включает робототехническое средство (РТС) и пульт управления (рисунок 1).



Рис. 1. Общий вид робототехнического комплекса «МРК-01»

Система управления робототехнического средства (СУ РТС) обеспечивает устойчивое управление его базовым шасси, рабочими органами и рабочим оборудованием. Управление осуществляется в супервизорном режиме посредством системы технического зрения, в которую входят две курсовые и одна поворотная бортовые видеокамеры. На пульте оператора размещен монитор.

Конструктивно СУ РТС делится на бортовую и пультовую части. Бортовая часть СУ РТС включает бортовую информационно-управляющую систему (БИУС), электроприводы шасси и рабочего оборудования, измерительные датчики, приемопередающее устройство. БИУС выполнена в виде одноплатного вычислительного модуля, в котором формируются команды управления для приводов, а также обрабатывается информация от измерительных

датчиков состояния и положения рабочего оборудования. Декодировкой управляющих сигналов занимается отдельный блок трансивера.

К 2018 году, не смотря на основательность и продуманность конструкции РТС, нарастающий поток сбоев и отказов со стороны электроники вынудил специалистов рассмотреть вариант глубокой модернизации электроники РТС. Повсеместный переход от аналоговой схемотехники к цифровой и применению в составе систем управления робототехнических средств микроконтроллеров, предопределил направление модернизации РТС и элементную базу.

Исторически Россия является первопроходцем, а в настоящее время – крупнейшем игроком в сегменте мобильного роботизированного пожаротушения [7]. Конкуренцию в мире в этом сегменте России составляет только Китай и только в классах легких и средних пожарных роботов. Экспертами J'son & Partners Consulting на основе ключевых критериев, приведенных к нескольким суммирующим индексам, был проведен рейтинг мобильных пожарных робототехнических комплексов [5]. Принимались во внимание, прежде всего, взрывозащищенность, для возможности тушения пожаров на объектах нефтегазового сектора, а также подвижность и время автономной работы. Также ключевыми метриками были дальность дистанционного управления (а в исключительных случаях – и возможность работы в полностью автономном режиме), наличие дополнительного оборудования (камеры, датчики, навесное оборудование), наличие эксплуатационных сертификатов. В целом, в исследование вошли 69 образцов мобильных пожарных роботов из 22 стран (по числу представленных на рынке моделей доминирует Россия и Китай). Из них была проведена выборка с детальным анализом технических характеристик наиболее зрелых моделей (ключевые критерии: коммерческая эксплуатация и защита от взрыва). В результате отбора в шорт-листе остались 27 моделей из 10 стран. В итоге на первой строчке рейтинговой таблицы оказалась Специальная Пожарная Машина (СПМ), производимая «Омсктрансмаш». Всего в десятке лучших – 4 российских мобильных пожарных робота, 5 китайских и 1 французский.

Стоит отметить, что лучшие модели российских мобильных роботов среднего и легкого класса создавались с использованием опыта зарубежных стран, в частности Австрии, Хорватии и Китая. Поэтому самым большим барьером в развитии отечественной пожарной робототехники на данный период следует признать возможные санкционные ограничения на поставку комплектующих и готовых изделий из упомянутых стран Евросоюза. Это, впрочем, не должно помешать плодотворному сотрудничеству с разработчиками пожарных роботов из КНР.

Российский ГОСТ Р 53326–2009 «Техника пожарная. Установки пожаротушения роботизированные. Общие технические требования. Методы испытаний» определяет:

1. установку пожаротушения роботизированную (УПР) как комплекс автоматических устройств, включающий два и более пожарных робота, систему определения координат загорания и устройство программного управления комплексом, соединенных информационным каналом связи и предназначенный для тушения и локализации пожара.
2. установку пожаротушения роботизированную позиционно-мобильную (УПРМ) как комплекс устройств, способный перемещаться по заранее подготовленной траектории к месту пожара с позиционированием в рабочей точке.
3. робот пожарный (РП) – автоматическое устройство, манипулирующее пожарным стволом в сферической системе координат, на базе стационарного лафетного ствола с дистанционным управлением, с фиксированной или подвижной установкой, с устройством обнаружения загорания и устройством программного управления, предназначенное для тушения и локализации пожара.

В этой части стандарт РФ расходится с западной практикой, в которой по отношению к стационарным средствам пожаротушения обычно не принято использовать понятия «ро-

бот» или «роботизированный». Там роботом считают устройство, движущееся автономно или под управлением человеком на расстоянии, то есть мобильность является определяющим признаком.

За рубежом обычно выделяют три группы стационарных средств пожаротушения: лафетные стволы с дистанционным управлением, устройства и системы пожарного оповещения, автоматизированные системы управления пожаротушением. Аналитики J'son & Partners Consulting сосредоточились на автоматизированных системах управления пожаротушением [5]. Проведенный анализ показал, что по технологическому уровню российские разработки практически ни в чем не уступают зарубежным аналогам. Разница между отечественным и западным рынками стационарной пожарной техники заключается в структуре рынка. Если западный рынок насыщен высокотехнологичными компаниями, производящими подобную технику, то российский рынок имеет значительно более высокий уровень монополизации. Доминирующее положение на нашем рынке занимает лишь один производитель, который активно участвует и в формировании нормативной базы и разработке стандартов в своем сегменте.

Работая на демонополизированном рынке, зарубежные производители автоматизированных систем управления пожаротушением часто декларируют инвариантность своих решений к конкретным типам лафетов и устройств пожарной тревоги, что дает широкий простор для архитектуры самых сложных современных комплексов пожаротушения для любых экстремальных условий применения и с учетом любых пожеланий заказчика.

При рассмотрении перспектив использования беспилотных летательных аппаратов для тушения пожаров следует отметить, что моделей, способных непосредственно участвовать в пожаротушении, во всем мире имеется не более двух десятков [2, 6]. Даже за рубежом, на фоне бурного развития разнообразной беспилотной техники, пожарные беспилотные летательные аппараты не воспринимаются как серьезный тренд. Лишь американский концерн Lockheed Martin пытается поддерживать данное инвестиционное направление, вопреки существующим в США законодательным ограничениям.

Если говорить о России, то можно отметить несколько инициативных проектов пожарных беспилотных летательных аппаратов на стадии прототипа и один концепт, пока не имеющий реального воплощения. На этом фоне выделяется проект тяжелой беспилотной платформы одного из российских производителей, успешное проведение летных испытаний и потенциальное наличие производственной базы у которого позволяет надеяться на появление в нашей стране реально полезного тяжелого пожарного беспилотного летательного аппарата.

Говоря о барьерах, сдерживающих развитие беспилотной пожарной техники в мире, прежде всего, необходимо отметить законодательные ограничения. Пока противоречие между длительной процедурой согласования условий полета беспилотных летательных аппаратов и необходимостью оперативного вылета пожарного беспилотника при возникновении чрезвычайной ситуации, выглядит неразрешимым во всем мире. Кроме этого сказывается и высокая зависимость полета от погодных условий. Еще одним очевидным сдерживающим фактором является нерешенный вопрос с безопасностью при совместном применении в зонах чрезвычайных ситуаций беспилотных летательных аппаратов и пилотируемой авиации. К примеру, американские пилоты пожарных танкеров отказываются летать в клубах дыма рядом с дронами. В Российской Федерации очевидно по тем же причинам в нормативных документах МЧС по состоянию на начало 2021 года отсутствовали указания на возможности пожаротушения с помощью беспилотных летательных аппаратов.

Сейчас нет оснований говорить о сколь-нибудь серьезном уровне использования беспилотных летательных аппаратов в пожаротушении не только в России, но и в мире в целом. Пока участие беспилотников ограничивается лишь мониторинговыми и разведывательными

функциями, которые от случая к случаю выполняют обычные беспилотные летательные аппараты, не специализированные для работы в условиях чрезвычайных ситуаций [1, 3].

Библиографический список

1. Беспилотники помогут спасти леса от пожаров. Мобильные бригады, оснащенные дронами, смогут обнаружить очаги возгораний [Электронный ресурс]. Режим доступа: <https://iz.ru/665872/dmitrii-stulin/bespilotniki-pomogut-spatat-lesa-ot-pozharov>. – Дата доступа: 11.02.2021.
2. Китайцы разработали пожарный дрон [Электронный ресурс]. Режим доступа: <https://nplus1.ru/news/2020/03/23/qilinguav>. – Дата доступа: 11.02.2021.
3. Пожарные дроны, беспилотные авиационные системы для тушения пожаров – перспективы использования [Электронный ресурс]. Режим доступа: <https://firecenter.ru/2426>. – Дата доступа: 11.02.2021.
4. https://www.challenges.fr/entreprise/le-robot-pompier-colossus-l-autre-heros-de-notre-dame_653569. – Дата доступа: 11.02.2021.
5. http://www.iksmedia.ru/company/json_partners_consulting.html. – Дата доступа: 11.02.2021.
6. <http://robotrends.ru/robopedia/katalog-bespilotnikov>. – Дата доступа: 11.02.2021.
7. <http://robotrends.ru/robopedia/katalog-pozharnyh-robotov>. – Дата доступа: 11.02.2021.

УДК 004.922; ГРНТИ 66.01.85

ИНФОРМАЦИОННЫЕ СИСТЕМЫ И ТЕХНОЛОГИИ В ЛЕСНОЙ ПРОМЫШЛЕННОСТИ

Д.З. Шаронова*, Г.С. Миронов*

**Сибирский государственный университет науки и технологий
имени академика М.Ф. Решетнева
Российская Федерация, Красноярск, darya-sharonova@mail.ru*

Аннотация. В статье рассматриваются возможности и перспективы использования информационных коммуникационных технологий в лесной промышленности для оптимизации использования лесных ресурсов и внедрения подхода устойчивого развития.

Ключевые слова: информационные коммуникационные технологии (ИКТ), лесное хозяйство, лесная промышленность, устойчивое развитие.

INFORMATION TECHNOLOGIES IN THE FOREST INDUSTRY

D.Z. Sharonova*, G.S. Mironov*

**Siberian State University of Science and Technology
named after academician M.F. Reshetnev
Russian Federation, Krasnoyarsk, darya-sharonova@mail.ru*

Abstract. The article discusses the possibilities and prospects of using information communication technologies in the forest industry to optimize the use of forest resources and introduce a sustainable development approach.

Keywords: information and communication technologies (ICT), forestry, forest industry, sustainable development.

Информационные коммуникационные технологии (далее – ИКТ) – это не только новые технологии, но и новые способы ведения дел. ИКТ можно рассматривать как три взаимосвязанные темы:

- 1) новые разработки в сфере информации и коммуникации;
- 2) инновации, изменения в организациях и изменения в отраслевой рабочей/деловой практике;
- 3) насколько быстро и насколько широко эти разработки внедряются в обществе и в экономике.

Детали технологии менее важны, чем изменения, которые ИКТ приносят в базовые структуры общества. Например, ИКТ имеют важное значение для того, как общество организует работу и создает экономическое богатство, а также для того, как люди проводят

свой досуг. Они помогают по-новому связать людей, экономику и общество – в этом контексте часто используются слова «глобализация» и «сетевое взаимодействие». Таким образом, анализ в данном исследовании делает акцент на воздействии ИКТ, а не на самой технологии.

Воздействие ИКТ на глобальный лесной сектор можно рассматривать по-разному. Например, в странах, где лесной сектор играет важную роль (например, в России, в Канаде, Финляндии, Швеции, Соединенных Штатах Америки), нередко можно противопоставить новое «общество знаний» или «общество ИКТ» секторам с традиционным подходом, таким как лесная промышленность. В то время как первое рассматривается как представляющее будущее и надежду, второе рассматривается как нечто принадлежащее прошлому.

Действительно, во многих из только что упомянутых стран этот устаревший образ все более затрудняет привлечение новых поколений к изучению предметов, связанных с лесной промышленностью, или к работе в лесной промышленности. Интересно, что в ряде экономически менее развитых стран происходит прямо противоположное.

Лесная промышленность привлекает все больше инвестиций, рабочих мест и интереса в таких странах, как Бразилия, Чили, Китай, Индонезия, Польша. Российским лесопромышленникам есть чему поучиться у иностранных коллег. ИКТ могут стать источником новых возможностей и нового имиджа.

Как и во многих других секторах, ИКТ могут способствовать новым изобретениям и большему процветанию. Поскольку такие возможности пока не присущи существующим структурам российского лесного сектора, необходимо искать новаторские пути сочетания ИКТ и производственных процессов, связанных с лесным хозяйством [1].

Наряду с упомянутыми выше изменениями на макроуровне в различных подсекторах лесной промышленности, происходит также большое количество более конкретных и фундаментальных изменений. Действительно, трудно представить себе проблемы в лесном секторе, которые не затрагиваются ИКТ. С другой стороны, глобальный лесной сектор является настолько крупным образованием, что ИКТ не могут оказывать единообразного и одновременного воздействия на каждую его часть.

Для многих подсекторов ИКТ, как представляется, являются новым двигателем прогресса и открывают новые возможности. Для других это может быть разрушительная или даже «убийственная» технология. Кроме того, во многих случаях воздействие ИКТ еще не может быть четко прослежено. Скорость, с которой эти факторы влияют на сектор, вероятно, будет варьироваться в зависимости от различных географических местоположений и подсекторов.

Воздействие ИКТ, которое является медленным и постепенным, может быть столь же значительным, как и непосредственные изменения, главным образом из-за присущего лесному сектору долгосрочного характера. Например, деревья, посаженные сегодня в естественных бореальных лесах, могут не достичь своего оптимального возраста сбора урожая в течение 70-100 лет. Точно так же после того, как лес полностью вырублен, могут пройти сотни лет, прежде чем он вернется в свое первоначальное состояние. Инвестиции в лесную промышленность обычно делаются на основе временного горизонта 15-30 лет.

Таким образом, вопросы лесного сектора – производство древесины, рынки лесных товаров, сохранение лесов и биоразнообразия – требуют долгосрочного подхода. Вот почему так важен анализ медленных, постепенных тенденций. Оценки и прогнозы этих тенденций будут привлекать внимание к возникающим проблемам, указывать на вероятное воздействие мер вмешательства и направлять разработку инвестиционных и других решений о распределении ресурсов. Новая и меняющаяся операционная среда, обусловленная ИКТ, также создает важные направления для исследований в лесном секторе. В фундаментальных исследованиях могут потребоваться новые или обновленные модели и методы. В прикладных исследованиях необходимы новые эмпирические результаты для

количественной оценки воздействия ИКТ на лесной сектор. Прикладные исследования имеют несомненно важное значение в отношении будущего развития российского лесного хозяйства.

Масштабы лесного хозяйства нашей страны поистине огромны. Нередко леснику приходится управлять лесным участком размером в 100 тыс футбольных полей с запасом 1 тыс различных пород деревьев. Все виды деревьев имеют различные характеристики древесины и роста. Эффективное и устойчивое управление такими сложными лесными экологическими системами является сложной задачей и требует прочной и всеобъемлющей информационной базы для поддержки хорошо обоснованных управленческих решений. Однако до недавнего времени стоимость данных инвентаризации лесов была очень высокой. Огромные объемы лесных данных, таких как географическое положение деревьев, тип древесины, объем древесины, а также информация о качестве, должны были быть записаны буквально вручную [2].

Законы экономики вынудили лесные хозяйства существенно сократить расходы на мониторинг путем преобразования разнообразных природных лесов в легко наблюдаемые и управляемые лесные монокультуры. Минусы такого подхода, однако, были многообразны:

- Преобладающее производство только нескольких различных массовых пород древесины, таких как кедр или сосна, несет высокий риск на конкурентных международных рынках;
- Вредители и болезни гораздо чаще распространялись в однородных лесных насаждениях;
- Развивались системные проблемы экологии, такие как связывание углерода, истощение плодородия почвы, удержание воды и т. д.;
- Места обитания многих растений и животных были уничтожены в больших масштабах.

Решением может быть использование в лесном хозяйстве информационных коммуникационных технологий.

Эффективное и устойчивое управление сложными и разнообразными лесными экосистемами требует большого объема высококачественных лесных данных. Стремительные разработки в области обработки данных и технологий, таких как мобильные карманные компьютеры и беспилотные летательные аппараты, произвели революцию в доступных лесных информационных системах и сборе данных.

Лесные информационные технологии помогли создать благоприятные базовые условия для более устойчивого и эффективного лесопользования. Многие лесные предприятия начали применять «портфельный подход» для управления разнообразными лесными экосистемами с многочисленными потоками доходов от различных древесных пород деревьев, недревесной лесной продукции и экосистемных услуг при минимизации уязвимости к вредителям и болезням, а также рыночных рисков.

Хотя спрос на лесные информационные технологии уже растет, как показал один из наших обзоров рынка, все еще существует большой потенциал расширения возможностей устойчивого лесопользования с помощью лесных информационных технологий. Мы считаем, что информационные технологии эффективно становятся большим рыночным преимуществом, оказывая при этом положительное влияние на окружающую среду.

Как некоторые из этих информационных технологий используются в практике устойчивого лесопользования, показано в следующих двух примерах использования.

Эффективный сбор данных с помощью беспилотных летательных аппаратов. В последние годы число применений беспилотных летательных аппаратов резко возросло, это также относится и к лесному сектору. По сравнению с заказом спутниковых снимков, беспилотный сбор данных не зависит от облачности, является быстрым, гибким и обеспечивает получение четкого и ясного изображения высокого качества.

Дроны – это гибкая платформа для размещения различных датчиков в трехмерном пространстве. Датчики обеспечивают получение изображений и пространственных данных, которые могут быть дополнительно обработаны в виде ортофотокарт или 3D-моделей поверхности полога леса. Полученные результаты могут быть использованы для картографирования, инвентаризации лесов, оценки ущерба лесам, сертификации лесов или для выявления незаконной деятельности, такой как кража древесины, незаконная добыча полезных ископаемых и посягательство на леса [3].

На следующем рисунке показана детальная ортофотосъемка с левой стороны, сливающаяся с цифровой моделью рельефа поверхности купола с правой стороны. Изменения объема полога, вызванные уборочными работами, могут быть количественно оценены автоматически или визуально обнаружены с помощью аэрофотоснимков высокого разрешения, как это видно на следующем изображении.



Рис. 1. Сопоставление ортофотокарты, сделанной с помощью дрона (слева), и спутникового снимка (справа)

С применением специальных алгоритмов изображения, полученные с беспилотников, могут быть использованы в однородных лесных насаждениях для автоматизированного подсчета деревьев и измерения высоты деревьев, как показано на следующем рисунке.



Рис. 2. Изображение поваленного дерева, обнаруженное с помощью дрона

Эффективный сбор данных на основе беспилотных летательных аппаратов прекрасно сочетается с центральным хранилищем данных для всех видов лесных данных – Лесной

информационной системой. Мы считаем, что быстрый сбор данных, например с помощью беспилотных летательных аппаратов, а также централизованное и доступное хранение данных и управление ими могут стать ключом к устойчивому и экономически успешному ведению лесного хозяйства.

Каждое дерево записывается по его географическому положению, типу древесины, объему и качеству. Лесники, аудиторы и покупатели древесины имеют доступ к лесной информации, например, с помощью веб-карты, как показано в следующей видеозаписи. Деревья будут заготавливаться только после твердого обязательства со стороны покупателя древесины. В базе данных Лесной информационной системы управляющий лесом может проверять темпы роста лесов, стоячие запасы, а также соответствие устойчивым ежегодным допустимым рубкам.

Благодаря современным лесным информационным технологиям можно решить конфликт между экономическими целями (получением прибыли) и целями устойчивого развития. В настоящее время необходимо выйти за рамки традиционной парадигмы лесного хозяйства и работать над ответственным лесопользованием, которое сочетает в себе экономическую конкурентоспособность и защиту ценных, трудновосполнимых лесных ресурсов.

Появление цифровых информационно-коммуникационных технологий (ИКТ) создало новые возможности для глобального лесного сектора. Наиболее значительные последствия развития ИКТ до сих пор были связаны с повышением производительности труда и увеличением спроса на бумажную продукцию. ИКТ повысили производительность и снизили производственные издержки как в лесной промышленности, так и в самом лесном хозяйстве. Потребление бумаги заметно возросло в результате внедрения современной офисной техники (персональные компьютеры, копировальные аппараты, принтеры). Внедрение систем глобального позиционирования и спутниковой съемки произвело революцию в мониторинге и управлении лесными ресурсами [4].

Последствия, вероятно, будут более значительными в будущем, чем в прошлом, и во многих случаях качественно иными или даже неожиданными. Таким образом, систематическое рассмотрение этой темы может помочь российскому лесному сектору внести соответствующие коррективы в долгосрочную стратегию своей деятельности.

В 2020 году Федеральным агентством лесного хозяйства (Рослесхоз) была создана единая ИТ-платформа, цель которой – организация взаимодействия бизнеса и государства в лесной промышленности. На сайте организации доступна интерактивная карта «Леса России», в которой находится информация о характеристике лесов по целевому назначению, о территориях лесного фонда, переданных в аренду, о площади лесов и запасах древесины по преобладающим породам и группам возраста деревьев и т.д. [3].

В перспективе предполагается создать общероссийский цифровой ресурс на базе Рослесхоза. С помощью данной системы государство и бизнес будут взаимодействовать в электронном формате. Например, выбор инвесторами лесного участка будет организован с помощью интерактивной карты, а права на использование участка будут продаваться на открытом электронном аукционе. В 2023 году планируется запуск «Цифрового леса», в котором будут объединены информационные системы управления лесным хозяйством ЛесЕГАИС и информационная система дистанционного мониторинга Рослесхоза.

В настоящее время в тестовом режиме работают подсистемы «Учет лесного фонда», «Лесопользование» и «Администрирование платежей». Лесной сектор только начал осознавать вероятное долгосрочное воздействие ИКТ и понимать их потенциальную величину. Взгляды на характеристики, количество и сроки этих воздействий, как правило, существенно различаются во всем лесном секторе. Такие расхождения во взглядах можно отчасти объяснить отсутствием научных исследований по данной теме и отсутствием

соответствующих данных. Таким образом, ИКТ создают новые направления не только для глобального лесного сектора, но и для научных исследований в области лесного хозяйства.

Библиографический список

1. Распоряжение Правительства РФ от 20.09.2018 N 1989-р «Об утверждении Стратегии развития лесного комп' лекса Российской Федерации до 2030 года» [Электронный ресурс]. URL: <http://www.garant.ru/products/ipo/prime/doc/71960006/>
2. Распоряжение Правительства РФ от 28.07.2017 N 1632-р «Об утверждении программы «Цифровая экономика Российской Федерации». [Электронный ресурс]. URL: <http://static.government.ru/media/files/9gFM4FHj4PsB79I5v7yLVuPgu4bvR7M0.pdf>
3. Федеральное агентство лесного хозяйства [Электронный ресурс] URL: <http://rosleshoz.gov.ru/>
4. Поконов А.А. Использование современных информационных технологий в отечественной промышленности (на примере лесного комплекса Российской Федерации) [Электронный ресурс] URL: <https://cyberleninka.ru/article/n/ispolzovanie-sovremennyh-informatsionnyh-tehnologiy-v-otechestvennoy-promyshlennosti-na-primere-lesnogo-kompleksa-rossiyskoy>

УДК 004.415.2; ГРНТИ 20.01.07

ОСОБЕННОСТИ КОНСОЛИДИРУЮЩЕЙ БИЛЛИНГОВОЙ СИСТЕМЫ РАСЧЕТНО-ПЛАТЕЖНОГО КОМПЛЕКСА «АБОНЕНТ+»

Д.В. Аникеев

*Рязанский государственный радиотехнический университет имени В.Ф. Уткина,
Российская Федерация, Рязань, d.anikeev@abonentplus.ru*

Аннотация. В работе рассматривается реализация консолидирующей биллинговой системы на примере консолидирующей биллинговой системы расчетно-платежного комплекса «Абонент+». Перечислены основные особенности реализации консолидирующей функции системы. Описаны подходы, обеспечивающие распределенный характер биллинговой системы.

Ключевые слова: консолидирующие биллинговые системы (CBS), очереди сообщений, протокол AMQ, консолидация биллинговых систем с прямым выставлением счета (DBS), аппроксимация функций расчета DBS, применение документ-ориентированных СУБД для целей реализации распределенного кэширования.

CONSOLIDATING BILLING SYSTEM'S FEATURES OF «ABONENT+» CALCULATION AND PAYMENT SOFTWARE SYSTEM

D.V. Anikeev

*Ryazan State Radio Engineering University named after V.F. Utkin,
Russia, Ryazan, d.anikeev@abonentplus.ru*

Annotation. The paper discusses consolidating billing system of «Abonent+» calculation and payment software system as an example of consolidating billing systems. Given main features that provide it to be a consolidating billing system. Described approaches that provide it's distributed features.

Keywords: consolidationg billing systems (CBS), message queues, AMQ protocol, consolidation of direct billing systems (DBS), approximation of direct billing systems' calculation functions, use of document-oriented databases for distributed caching.

Консолидирующие биллинговые системы представляют собой комплексные программные решения, предназначенные для расчетов с потребителями (абонентами) за предоставляемые им те или иные виды услуг, например жилищно-коммунальные [1]. Консолидация предполагает предоставление абоненту единого счета на оплату услуг, расчет и учет которых осуществляется в различных, зачастую территориально распределенных, биллинговых системах. Кроме формирования единого счета на оплату CBS также обеспечивают возможность расчета в режиме реального времени сумм, представляемых к оплате, на основе передаваем-

мых абонентом в момент оплаты показателей потребления (показатели снимаются с устройств учета объемов потребления различных коммунальных ресурсов либо абонентом вручную, либо автоматически). Сведения о произведенной оплате CBS распределяют между биллинговыми системами, осуществляющими учет всех услуг, предоставляемых абоненту.

CBS по сравнению с биллинговыми системами с прямым выставлением счета являются более сложными в реализации, так как оперируют большими объемами информации (интегрируют данные по всем услугам, оказываемым абоненту, а не только один класс услуг, предоставляемый конкретным поставщиком). Архитектура CBS является смешанной: распределенный характер таких систем обусловлен множеством входящих в них DBS, а централизованный — необходимостью консолидации сведений в рамках одного абонента [1].

Примером CBS является «Расчетно-платежный комплекс «Абонент+» [3]. Программный комплекс включает в себя устанавливаемые для целей расчета поставщиками услуг DBS «Расчетная система «Абонент» (далее — РС) и CBS «Платежная система «Абонент» (далее — ПС). Интеграция РС и DBS реализована посредством двух протоколов обмена: протокола выгрузки из DBS данных о жилых помещениях для целей консолидации и последующего формирования единых счетов и протокола загрузки в DBS сведений о платежах, принятых посредством ПС.

Протокол выгрузки данных позволяет загружать в ПС в режиме онлайн сведения о жилых помещениях вне зависимости от типа DBS. Для описания жилого помещения используется иерархическая структура в формате JSON [4]. Также ПС поддерживает импорт в режиме оффлайн сведений о жилых помещениях, представленных в виде наборов данных в форматах CSV, XML, JSON.

Протокол загрузки сведений о платежах предназначен для расщепления консолидированных платежей, принятых на основе ранее выставленного единого счета. Данные в DBS загружаются в режиме онлайн в виде XML-документов с использованием протокола SOAP [5]. Как и в случае с протоколом выгрузки данных, ПС поддерживает обмен сведениями о платежах в режиме онлайн с использованием наборов данных в форматах CSV, XML, JSON.

К отличительным особенностям CBS «Платежная система «Абонент» следует отнести:

- 1) применение кортежей [6] в качестве базового типа параметров консолидации биллинговых данных;
- 2) приведение комплексных функций расчета, применяемых в DBS, к кусочно-линейным функциям [7] для осуществления в реальном времени расчета суммы к оплате по выставленным счетам;
- 3) формирование очередей сообщений на основе AMQP [8] для обеспечения высокой отказоустойчивости протокола выгрузки данных из DBS;
- 4) использование NoSQL БД [9] для целей реализации функции приема платежей в территориально распределенных кассовых пунктах.

Применение кортежей в качестве базового типа параметров консолидации дало возможность унифицировать структуру консолидирующих параметров, что в свою очередь позволило автоматизировать процесс консолидации. В качестве базового параметра консолидации используется адрес жилого помещения абонента. Адрес представляется в виде кортежа (x_1, \dots, x_n) , где $n \in [1, (k + m)]$, $k \in [1, 7]$, $m \in [1, 8]$. k обозначает адрес абонента без указания информации о доме и определяется структурой федеральной информационной адресной системы (ФИАС) [10]. Диапазон m характеризует обслуживаемое жилое помещение (поддерживаются номера домов, домов с корпусом, квартир и комнат с литерами и без). В отдельных случаях (несколько обслуживаемых абонентов, проживающих в одном жилом помещении) базовая структура консолидирующего параметра может быть расширена за счет дополнительных идентифицирующих параметров (например, номер свидетельства о собственности). Для определения коэффициента подобия жилых помещений, выгруженных из

разных DBS, CBS «Платежная система «Абонент» на первом шаге вычисляет частный коэффициент подобия соответствующих элементов характеризующих кортежей (которые в свою очередь также являются кортежем символов), а на втором шаге — осуществляет свертку частных коэффициентов на основе эмпирически заданной весовой функции. В случае, если полученное после второго шага значение общего коэффициента подобия близко к единице, сведения об абонентах консолидируются автоматически, иначе сопоставление данных осуществляется оператором системы.

Приведение комплексных функций расчета, к кусочно-линейным позволило унифицировать специфику расчета, реализованную в различных DBS. Аппроксимация осуществляется путем определения периодов стабильности (участков, на которых исходная функция изменяется линейной). Выявленные периоды стабильности в кусочно-линейной функции определяются контрольными точками (границы периодов), базовым смещением и углом наклона. Большая часть вариантов расчета, предусмотренных современным законодательством РФ [11], является множеством линейных преобразований базовых параметров расчета (нормативов, тарифов и т.п.), а потому может быть представлена в виде совокупности линейных функций при условии их неизменяемости на выявленных периодах стабильности. Такой подход с достаточно высокой точностью и при высокой скорости расчета позволяет вычислять не только суммы к оплате с учетом передаваемого абонентом объема потребления, но и размер неустойки за просроченную задолженность.

Формирование очередей сообщений повысило отказоустойчивость протокола выгрузки данных из DBS как за счет гибкой настройки параметров работы брокеров сообщений под конкретное серверное окружение, так и за счет построения кластеров из нескольких отдельных сервисных процессов. Помимо прочего за счет типизации очередей в ПС имеется возможность выделения классов сообщений на основе времени их обработки: сообщения с длительным временем обработки могут помещаться в отдельные («медленные») очереди, а в случае выявления регулярной ошибки обработки того или иного сообщения оно помещается в отдельную очередь для ручного анализа оператором CBS.

Использование NoSQL БД позволило организовать кэширование выставленных единых счетов за предоставленные услуги. Кэширование значительно уменьшило время отклика ПС, так как процесс подготовки сведений для приема платежей выполняется не более одного раза вне зависимости от количества поступающих запросов на оплату (или передачу показаний приборов учета). Кэширование в CBS реализуется путем разделения всех сведений о выставленном счете на два класса: статические (сведения о жилом помещении, функции расчета объема потребления и т.п.) и динамические (сведения о ранее произведенных в системе платежах). Статические данные подготавливаются на этапе выгрузки информации из DBS. Динамические данные представлены в виде списка приращений (дельта). После выгрузки сведений из DBS динамические данные очищаются. Использование NoSQL БД помимо повышения быстродействия CBS позволяет при высокой скорости вставки (по сравнению с реляционными базами данных нет необходимости осуществлять проверку целостности кэша) и чтения (данные о жилом помещении загружаются всегда в полном объеме, а их поиск осуществляется по известным на этапе формирования ключам) обеспечить энергонезависимость системы (в отличие от резидентных СУБД перестроение кэша не требуется при краткосрочном отключении питания серверов) и реализовать сеть доставки контента (Content Delivery Network – CDN) [12] благодаря встроенному механизму репликации.

Вышеописанные особенности «Расчетно-платежного комплекса «Абонент+» позволяют ежемесячно выставлять порядка миллиона единых счетов более чем 3 миллионам абонентам из 130 различных DBS, располагающихся в нескольких регионах РФ, и обрабатывать 1,5 миллиона консолидированных платежей.

Библиографический список

1. Shen, Y. Research of the Convergent Billing System Based on the Whole Business / Y. Shen, Z.-S. Gu, S. Wang, L. Zhang. — Shanghai: International Conference on Management of e-Commerce and e-Government. — 2013. — с. 304-307.
2. Таненбаум Э., ван Стеен М. Распределенные системы. Принципы и парадигмы / Э. Таненбаум, М. ван Стеен. — СПб.: Питер. — 2003. — с. 880.
3. ООО «Абонент+» [Электронный ресурс]. — 2021. Режим доступа: <https://abonent.plus/>. — Дата доступа: 16.02.2021.
4. RFC 7159. The JavaScript Object Notation (JSON) Data Interchange Format [Электронный ресурс]. — 2007. — Режим доступа: <https://www.w3.org/TR/2007/REC-soap12-part0-20070427/>. — Дата доступа: 16.02.2021.
5. SOAP Version 1.2 Part 0: Primer (Second Edition) [Электронный ресурс]. — 2007. — Режим доступа: <https://tools.ietf.org/html/rfc7159>. — Дата доступа: 17.02.2021.
6. Кормен Т., Лейзерсон Ч., Ривест Р., Штайн К. Алгоритмы: построение и анализ — 2-е изд. / Т. Кормен, Ч. Лейзерсон, Р. Ривест, К. Штайн — М.: Вильямс. — 2005. — с. 1296.
7. Никольская И.Л. Факультативный курс по математике. / И.Л. Никольская — М.: Просвещение. — 1991. — с. 272-274.
8. AMQP Working Group 0-9-1 [Электронный ресурс]. — 2008. — Режим доступа: <https://www.amqp.org/specification/0-9-1/amqp-org-download>. — Дата доступа: 17.20.2021.
9. Маркин А.В. Постреляционные базы данных. MongoDB: учебное пособие — 2-е изд. / А.В. Маркин — М.: Ай Пи Ар Медиа. — 2020. — с. 383.
10. О федеральной информационной адресной системе и о внесении изменений в Федеральный закон "Об общих принципах организации местного самоуправления в Российской Федерации" / Федеральный закон от 28 декабря 2013 г. N 443-ФЗ. — 2013.
11. О предоставлении коммунальных услуг собственникам и пользователям помещений в многоквартирных домах и жилых домов / Постановление Правительства РФ от 06 мая 2011 г. № 354. — 2011.
12. Buuya R., Pathan M., Vakali A. Content Delivery Networks / R. Buuya, M. Pathan, A. Vakali. — Springer Science & Business Media. — 2008. — с. 417.

УДК 004.891.3; ГРНТИ 20.15.05

ПРЕДСКАЗАНИЕ САХАРНОГО ДИАБЕТА С ПОМОЩЬЮ НЕЙРОСЕТИ

В.А. Калашников, М.В. Саватеев, А.И. Мартышкин, Д.А. Трокоз

*Пензенский государственный технологический университет,
Российская Федерация, Пенза, alexey314@yandex.ru*

Аннотация. Универсальность нейронных сетей позволяет использовать данную технологию почти во всех сферах жизни. Нейронные сети снизили популярность в здравоохранении, облегчая работу специалистов и позволяют предупреждать проблемы, которые могут возникнуть из-за ошибки человека. В данной работе рассматривается возможность использование нейросети для предсказания сахарного диабета.

Ключевые слова: здравоохранение, предсказание заболеваний, диабет, полносвязные сети прямого распространения, машинное обучение, нейронная сеть.

PREDICTION OF DIABETES MELLITUS WITH NEURAL NETWORK

V.A. Kalashnikov, M.V. Savateev, A.I. Martyshkin, D.A. Trokoz

*Penza State Technological University,
Russia, Penza, alexey314@yandex.ru*

The summary. The versatility of neural networks allows the use of this technology in almost all spheres of life. Neural networks have gained popularity in healthcare, making it easier for professionals to work and preventing problems that can arise from human error. This paper discusses the possibility of using a neural network to predict diabetes mellitus.

Keywords: healthcare, disease prediction, diabetes, fully connected feedforward networks, machine learning, neural network.

На сегодняшний день в большинстве стран, чтобы стать врачом, необходимо потратить много лет на обучение. Сам же процесс становления очень и очень сложен. Тем време-

нем развитие технологий искусственного интеллекта позволяет применять нейросети буквально во всех сферах деятельности человека. В сфере здравоохранения нейросети позволяют минимизировать фактор врачебной ошибки, разгрузить рабочее время специалистов и поднять качество предоставляемых услуг за счет быстрого и высокоточного проведения анализа данных пациентов.

Нейросети в области здравоохранения применяются в следующих сферах:

1. Прогнозирование заболеваний на базе данных о состоянии здоровья и лабораторных анализах.
2. Классификация изображений, например, сканы, рентген, компьютерная томография и прочее с целью выявления каких-либо дефектов.

В данной работе рассматривается нейронная сеть, которая предсказывает у пациента сахарный диабет на базе второстепенных данных о состоянии здоровья человека.

Экспериментальные данные для обучения сети, взятые из медицинского архива, это обезличенная информация обо всех пациентах женского пола не моложе 21 года.

В наборе экспериментальных данных, состоящих из анонимных записей, имеется восемь параметров. Так же есть информация о диагностировании у пациентов сахарного диабета (соответственно 1 – «диагностирован» или 0 – «здоров»). Восемь остальных параметров также имеют численные значения:

- 1) число беременностей;
- 2) концентрация глюкозы в плазме через 2 часа после введения в пероральном глюкозотолерантном тесте;
- 3) диастолическое артериальное давление;
- 4) толщина кожной складки в районе трицепса;
- 5) концентрация инсулина в сыворотке крови;
- 6) индекс массы тела;
- 7) функция, описывающая генетическую предрасположенность к диабету;
- 8) возраст.

Индекс массы тела рассчитывается по формуле 1.

$$I = m/h^2, \quad (1)$$

где I – индекс массы тела;
 m – масса тела в килограммах;
 h – рост в сантиметрах;

Для моделирования сети использовался язык программирования Python, а также набор библиотек Kerasи Tensorflow. Сама нейронная сеть представляет собой полносвязную сеть прямого распространения, состоящую из нескольких слоев разной плотности. Модель полносвязной сети представлен на рисунке 1.

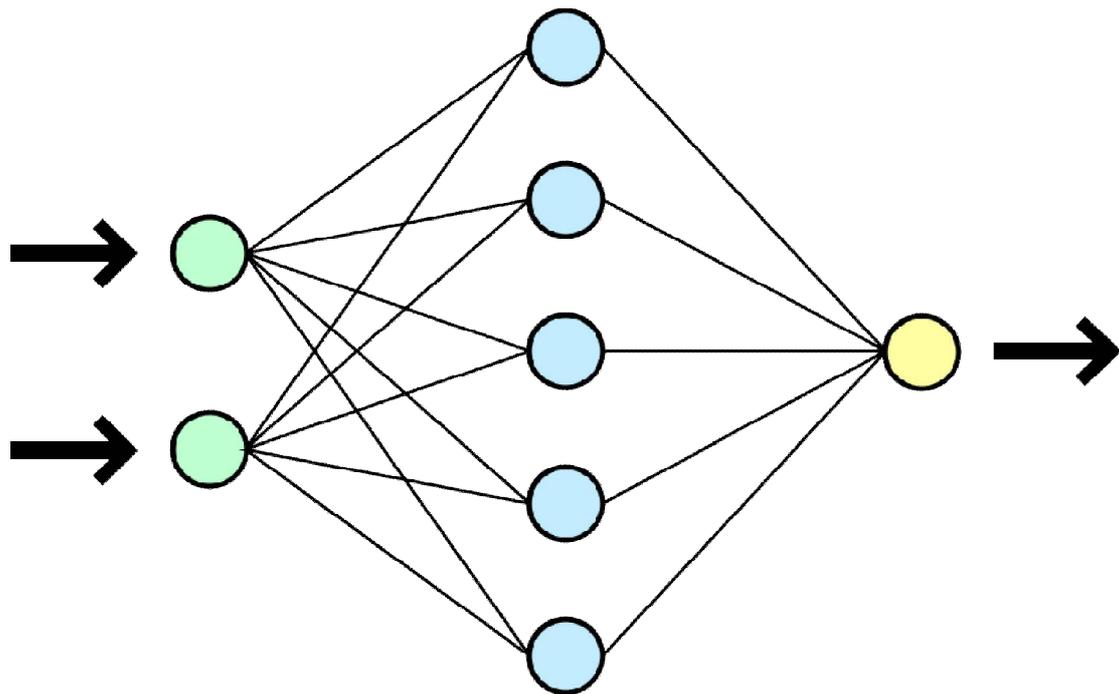


Рис. 1. Модель полносвязной нейронной сети прямого распространения

В качестве функции активации для всех слоев, кроме выходного, используется функция ReLU. Для выходного слоя используется сигмоидная функция для определения конечной вероятности риска заболевания. Графики функций представлены на рисунке 2.

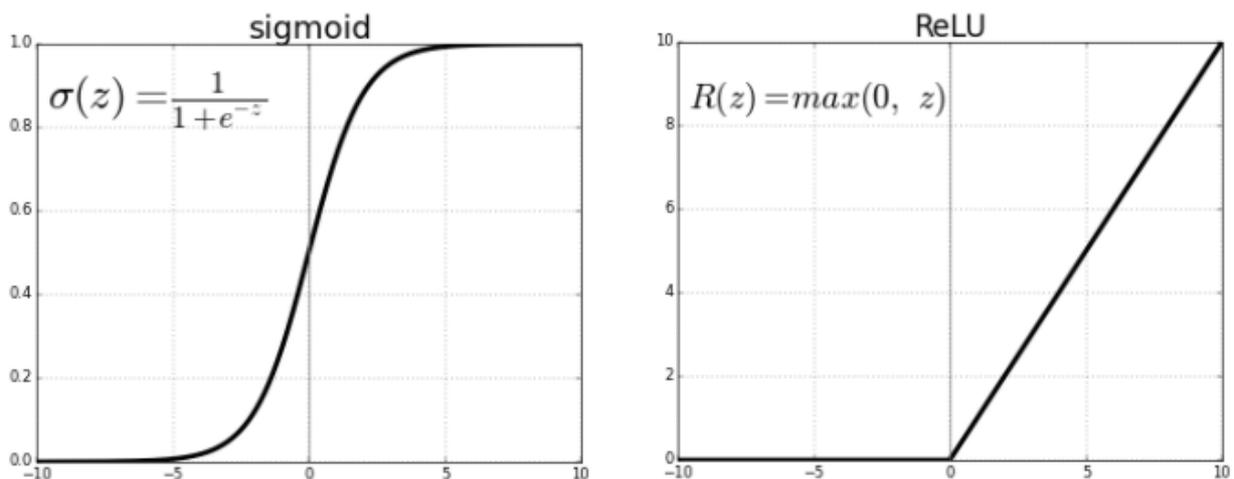


Рис. 2. График функции сигмоида и ReLU

Для обучения нейронной сети использовалось более 800 записей, о которых говорилось выше. Обучалась сеть в течение 1000 эпох (итераций обучения). После обучения начался процесс оценки точности предсказаний на новой тестовой выборке. Полностью обученная сеть предсказывала диабет с точностью 97,27%. Данный пример демонстрирует потенциал использования технологий машинного обучения в предсказании заболеваний по косвенным данным.

Выводы

В рамках данной работы была разработана и обучена нейросеть для предсказания у пациентов сахарного диабета. После обучения сеть смогла предсказать диабет с точностью 97,27%. Данная нейросеть способна за короткий срок проанализировать огромный массив данных и заблаговременно предупредить заболевания по косвенным данным, что в потенциале способно улучшить качество медицинских услуг путем раннего предупреждения болезни.

Библиографический список

1. Тарик Р., Создаем нейронную сеть. – Санкт-Петербург: ООО «Альфа-книга», 2017.
2. Джоши П., Искусственный интеллект с примерами на Python. – Санкт-Петербург: ООО «Диалектика», 2019.
3. Джулли А., Пал С., Библиотека Keras – инструмент глубокого обучения. Реализация нейронных сетей с помощью библиотек Theano и TensorFlow. – Москва: ДМК Пресс, 2018.

УДК 681.5.01; ГРНТИ 28.15.15

АВТОМАТИЗИРОВАННАЯ СИСТЕМА УПРАВЛЕНИЯ СБЫТОМ ГОТОВОЙ ПРОДУКЦИИ

Б.С. Дмитриевский, А.А. Терехова

*Тамбовский государственный технический университет,
Россия, Тамбов, terehova.aa@mail.tstu.ru*

Аннотация. В статье рассмотрена система автоматизированного управления сбытом готовой продукции на предприятии. Представлены функциональная схема работы информационной системы, описаны основные таблицы используемых баз данных. С помощью структурной схемы описан процесс потоков данных в автоматизированной информационно-управляющей системе.

Ключевые слова: система управления, сбыт продукции, информационная система, документооборот.

AUTOMATED CONTROL SYSTEM SALES OF FINISHED PRODUCTS

B.S. Dmitrievsky, A.A. Terekhova

*Tambov State Technical University,
Russia, Tambov, terehova.aa@mail.tstu.ru*

Abstract. The article considers the system of automated sales management of finished products at the enterprise. The functional scheme of the information system is presented, the main tables of the databases used are described. Using a block diagram, the process of data flows in an automated information and control system is described.

Keywords: management system, product sales, information system, document flow.

Управление на предприятии является одним из важнейших условий организации его эффективной работы. В свою очередь, процесс сбыта является средством достижения поставленных целей предприятия и завершающим этапом выявления вкусов и предпочтений покупателей.

Именно сбыт продукции для предприятия особо важен, так как его объем определяет ключевые показатели работы предприятия. Кроме того, именно в процессе сбыта окончательно определяется результат работы предприятия, направленный на расширение объемов деятельности и получение максимальной прибыли [4].

Процесс планирования является частью управления сбытом готовой продукции. Эта деятельность опирается на выявление и прогнозирование спроса, анализ и оценку имеющихся ресурсов и перспектив развития хозяйственной конъюнктуры. Отсюда вытекает необхо-

димось связывания планирования с маркетингом и контролем с целью осуществления непрерывной корректировки ключевых показателей производства и сбыта, которые зависят от спроса на рынке.

Целью разработки информационной системы предприятия является решение комплекса задач по автоматизации управления и планирования сбыта готовой продукции с использованием современных информационных технологий.

Внедрение системы позволит решить следующие задачи:

- повышение эффективности деятельности отдела сбыта;
- повышение оперативности получения аналитической информации;
- повышение скорости выписки документов при реализации готовой продукции;
- автоматизация анализа и прогнозирование объемов продаж;
- автоматизация анализа эффективности деятельности менеджеров по продаже.

Основными недостатками ручного метода управления и планирования сбыта являются чрезмерный объем информации, высокая трудоемкость ее обработки и низкая оперативность выдачи отчетности.

Отсюда возникает явная необходимость в автоматизации данных процессов [2].

На рисунке 1 представлена функциональная схема работы информационной системы в виде диаграммы IDEF0 нулевого уровня дитализации.

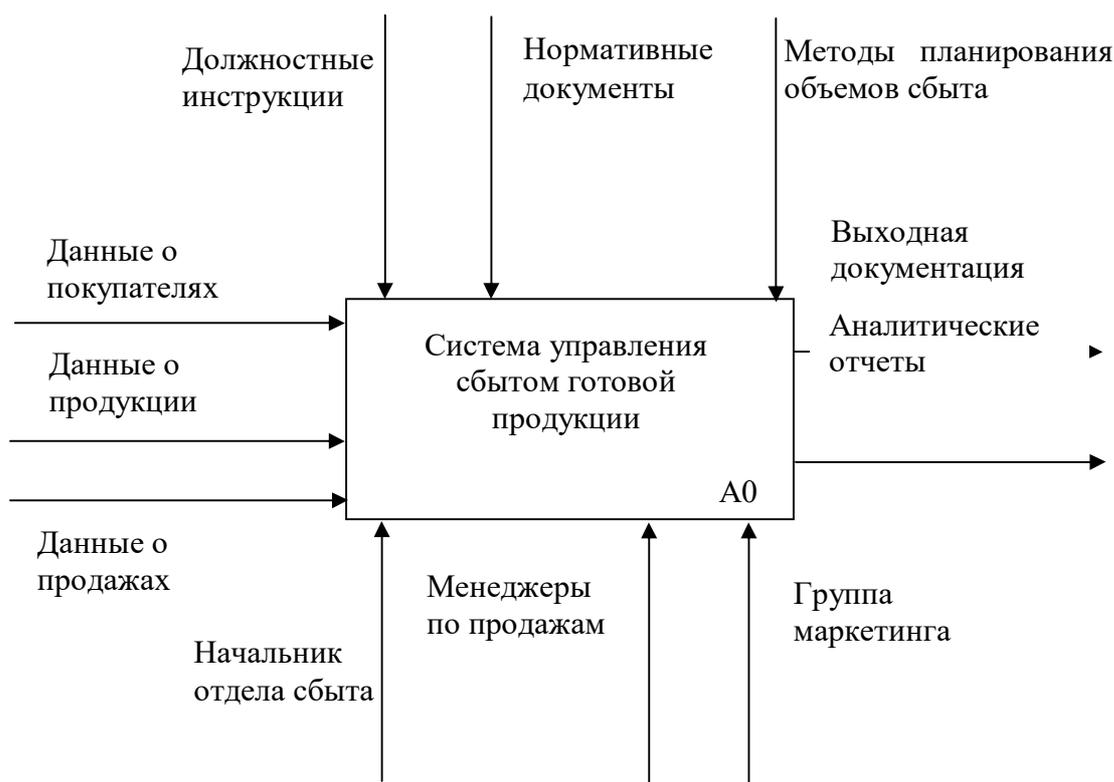


Рис. 1. Диаграмма IDEF0 нулевого уровня детализации

В информационной системе предполагается использование модели данных.

Разработанная программа использует следующие таблицы базы данных:

1. Номенклатура – справочник товаров.
2. Контрагенты – справочник покупателей.
3. Договоры – справочник договоров.

4. Места хранения – справочник мест хранения товара.
5. Менеджеры – справочник менеджеров по продажам.
6. Заявки – документы «Заявка покупателя».
7. Реализации – документ «Реализации продукции».

Таблицы баз данных созданы с использованием конфигуратора программного комплекса 1С:Предприятие в виде таких объектов, как справочники и документы.

Справочниками называются средства для работы со списками однородных данных. Именно при помощи справочников осуществляется ввод стандартной информации в формы документов, а также ее просмотр и корректировка, если таковая необходима. В большинстве своем справочники являются электронными аналогами каталогов. Каждая карточка - это строка справочника, а сведения, заносимые в карточку, являются реквизитами справочника [1].

Название и структура каждого определенного справочника формулируется и составляется при его создании в специализированном конфигураторе. У любого справочника существует два обязательных реквизита, которые создаются автоматически — "Код" и "Наименование". Другие реквизиты справочника определяются в конфигураторе конкретно для каждого создаваемого справочника, и могут содержать какую-либо дополнительную информацию [3].

Входными экранными формами являются:

- форма документа «Заявка покупателя»;
- форма документа «Складская реализация продукции».

Выходные формы:

- счет покупателю на русском и английском языках;
- товарная накладная;
- товарно-транспортная накладная;
- счет-фактура;
- форма отчёта «Анализ продаж»;
- форма отчёта «Анализ заявок покупателей»;
- форма отчёта «Прогнозирование продаж»;
- форма отчёта «Анализ эффективности работы менеджеров»;
- форма отчёта «Сегментация контрагентов на группы по объемам продаж»;
- форма отчёта «Сегментация товаров на группы по объемам продаж».

Схема документооборота на предприятии при сбыте готовой продукции представлена на рисунке 2.

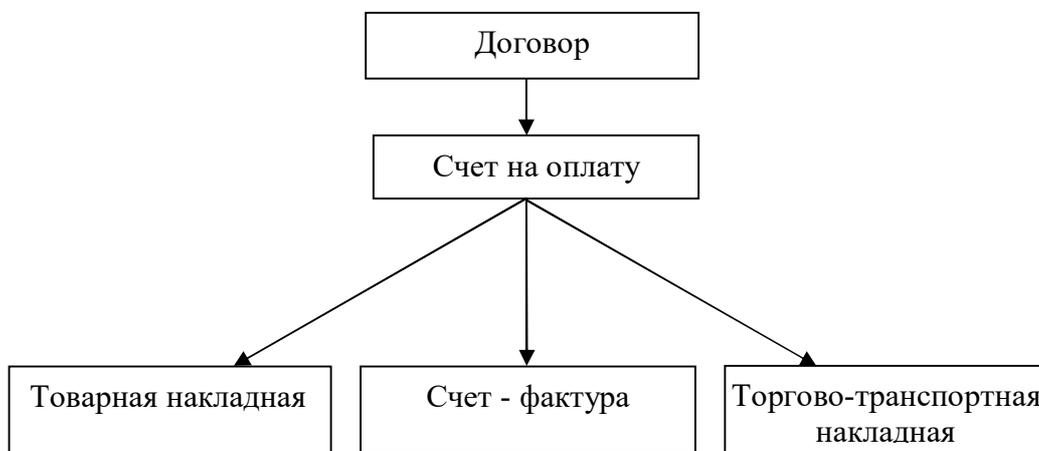


Рис. 2. Структура документооборота при сбыте готовой продукции

В процессе работы с предложенной системой автоматического управления менеджер по продажам принимает заявку покупателя, данные о покупателе заносятся в базу данных покупателей, формируется документ «Заявка покупателя», в который вносятся данные по покупателю, по товарам. Покупателю выписывается счёт на оплату.

В момент реализации происходит заполнение документа «Реализация» либо на основании заявки покупателя автоматически, либо данные по товару и покупателю заносит менеджер. Формируются реализационные документы такие как товарная накладная, счёт-фактура, товарно-транспортная накладная.

Структура потоков данных в предложенной автоматизированной информационно-управляющей системе представлена на рисунке 3.

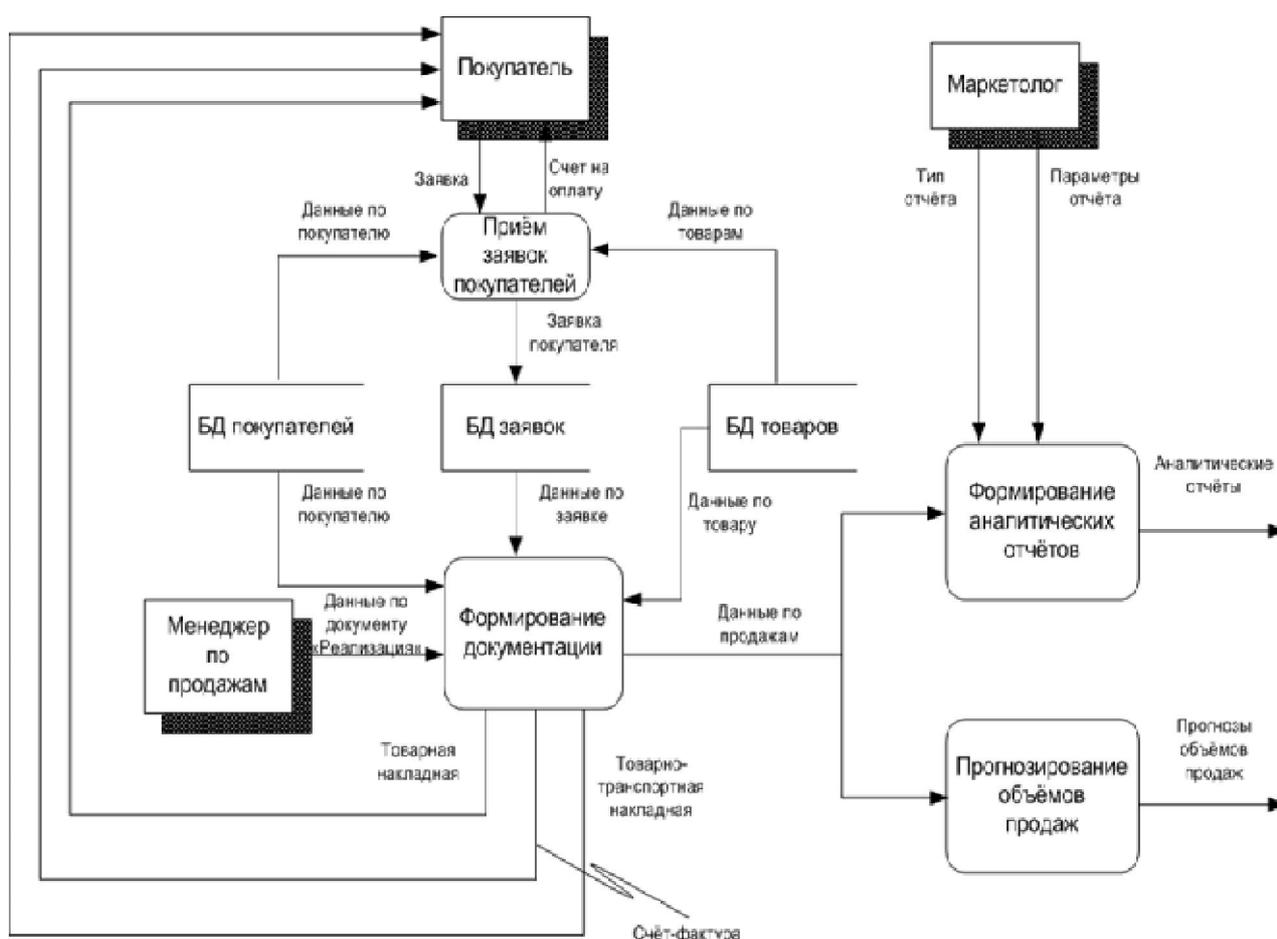


Рис. 3. Структура потоков данных в автоматизированной информационно-управляющей системе

На основании данных по продажам маркетологи получают необходимые аналитические отчёты и прогнозы объёмов продаж.

Сценарий работы предложенной системы автоматического управления сбытом готовой продукции на предприятии представлен на рисунке 4.

Разработанная подсистема управления сбытом готовой предназначена для формирования документации по реализации, а также аналитических отчётов по объёмам, динамики и прогнозам продаж.

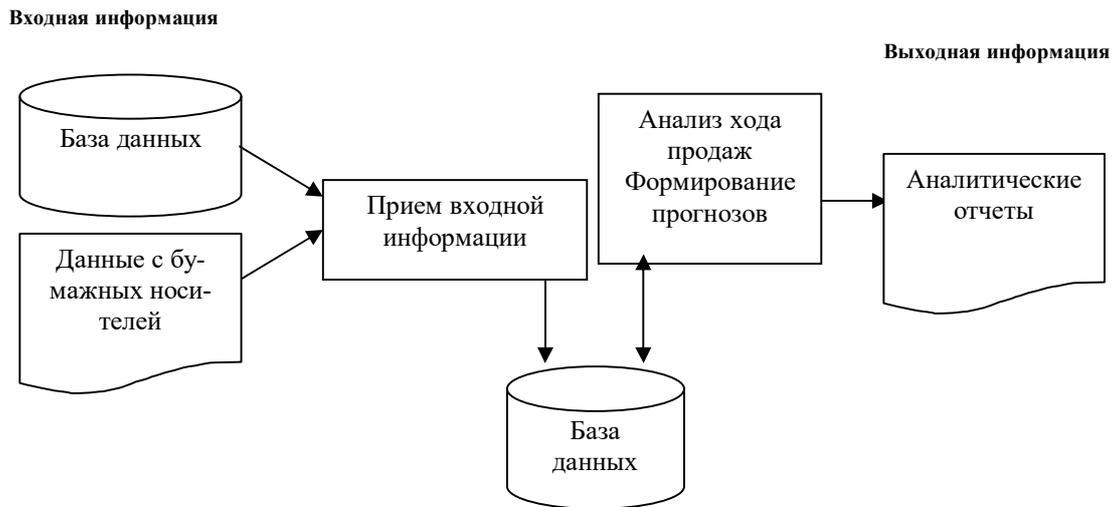


Рис. 4. Сценарий работы автоматизированной системы управления

Заключение

В статье были рассмотрены и решены следующие вопросы:

- сущность ведения сбытовой политики и планирования сбыта готовой продукции;
- анализ структуры и функций отдела сбыта и группы маркетинга;
- автоматизация ввода и получения аналитической информации по реализации продукции;
- разработка программного обеспечения для автоматизации основных этапов управления сбытом и его планирования.

Введение системы позволит повысить эффективность работы отдела сбыта и группы маркетинга, повысит оперативность получения информации, степень достоверности обработки информации, степень ее защищенности.

Внедрение данного проекта позволит решить следующие задачи:

- повышение эффективности деятельности отдела сбыта;
- повышение оперативности получения аналитической информации;
- повышение скорости выписки документов при реализации готовой продукции;
- автоматизация анализа и прогнозирование объемов продаж;
- автоматизация анализа эффективности деятельности менеджеров по продаже.

Библиографический список

1. Соловьёва, Ю.В. Внутрифирменное планирование : учебное пособие / Ю.В. Соловьёва, М.В. Черняев. — Москва : Дашков и К, 2019. — 210 с
2. Жуков, Б.М. Инновационное обеспечение гибкого управления предприятием. Теория, инструментарий, реализация : монография / Б.М. Жуков. — Краснодар : Южный институт менеджмента, 2010. — 410 с.
3. Дмитриевский, Б.С. Программно-алгоритмический комплекс защиты и управления предприятием / Б.С. Дмитриевский, В.Г. Матвейкин, В.И. Медников, С.Г. Семержинский // Программные продукты и системы. - 2017. - № 2. - С. 307-313.
4. Барданосова, А.А. Направление планирования сбыта нового продукта / А.А. Барданосова, К.П. Марченко, Н.Н. Тельнова // Экономика и социум. – 2016, №3. – с. 1738 – 1742.

УДК 004.94: 519.688; ГРНТИ 50.41

ЧИСЛЕННЫЙ МЕТОД РАСЧЕТА ДАННЫХ ПОТРЕБЛЕНИЯ КОММУНАЛЬНОГО РЕСУРСА ДЛЯ ПРОВЕДЕНИЯ OLAP АНАЛИЗА

М.Г. Костиков

*Рязанский государственный радиотехнический университет имени В.Ф. Уткина,
Российская Федерация, Рязань, m.kostikov@abonentplus.ru*

Аннотация. В работе рассматривается численный метод расчета данных потребления коммунального ресурса для проведения OLAP анализа и проводится сравнительный анализ алгоритмов определения объемов потребления коммунальной услуги, в случае отсутствия показаний приборов учета.

Ключевые слова: численный метод, объем потребления коммунальной услуги, интерполяция, приборы учета.

NUMERICAL METHOD OF PREPARING DATA ON CONSUMPTION OF UTILITY RESOURCE FOR OLAP ANALYSIS

M.G. Kostikov

*Ryazan State Radio Engineering University named after V.F. Utkin,
Russia, Ryazan, m.kostikov@abonentplus.ru*

Annotation. The paper discusses a comparison of algorithms for determining the volume of consumption of public services in the absence of meter readings.

Keywords: utilities consumption, interpolation, metering devices..

Рост уровня автоматизации и информатизации практически во всех сферах деятельности человека позволил накопить огромное количество информации о соответствующих бизнес-процессах. Анализ этих данных, позволяет не только составлять отчеты, но также определять тренды, выявлять зависимости, а также строить прогнозы развития процессов в будущем. Комплексный анализ и внедрение специализированных подходов бизнес-аналитики (OLAP) позволяют существенно повысить точность управления бизнес-процессами, а следовательно и эффективность всей организации в целом, независимо от ее сферы деятельности. Не является исключением и сфера ЖКХ. Обязательства ресурсоснабжающих организаций (РСО), связанные с необходимостью непрерывного обеспечения граждан коммунальными ресурсами, и наличие выраженных сезонных трендов заставляют поставщиков услуг очень тщательно относиться к планированию собственных ресурсов, которое невозможно без глубокого анализа. Однако, прежде чем строить гипотезы на основе исходных данных, необходимо убедиться, что эти данные корректны. В соответствии с принципами ETL, перед непосредственным использованием данных в аналитической системе они должны быть подготовлены и очищены от ошибок, так как их наличие неминуемо будет приводить к искажению гипотез. С точки зрения предметной области ЖКХ это особенно важно, так как связано с тем, что данные аккумулируются из множества источников, в том числе, и непосредственно от абонентов, являющихся конечными потребителями коммунальных услуг. Данные могут быть разрежены, поступать не регулярно, или не поступать совсем, содержать ошибки и т. д. Целью данной работы является на примере двух ключевых параметров: объема потребления и показаний индивидуального прибора учета разработать подход, позволяющий восстановить истинные исходные данные, необходимые для работы с аналитическими системами.

Основным субъектом деятельности в сфере жилищно-коммунального хозяйства является абонент. Он является конечным потребителем коммунальных ресурсов, производит их оплату, в соответствии с расчетом, выполненным в РСО или расчетном центре. Непосредственно под расчетом понимается выполнение ежемесячных начислений и разовых перерасчетов по различным КУ в натуральном и денежном выражении, расчет пени и т.д [1].

Каждый абонент характеризуется некоторым набором количественных и качественных характеристик. Под количественными характеристиками абонента, понимаются харак-

теристики, непосредственно влияющие на размер начисления. К ним относятся число зарегистрированных граждан и животных, размер занимаемой жилой и нежилой площади и т.д. Качественная характеристика – величина, определяющая характер потребления КУ [2]. К ним относятся такие логические (номинальные) характеристики, как режимы работы оборудования, признак наличия или отсутствия горячего водоснабжения и т.д.

Можно выделить несколько типовых вариантов расчета. Если помещение, связанное с лицевым счетом абонента, не оборудовано индивидуальным прибором учета, для него выполняется расчет нормативного потребления. Подобный расчет является ежемесячным и зависит от некоторой нормативной величины потребления коммунального ресурса, приходящейся на количественную характеристику абонента, установленную, как правило, на уровне субъекта федерации [3].

$$V_i = S_i \times N, \quad (1)$$

где S_i - общая площадь i -го помещения;

N – норматив потребления коммунального ресурса;

V_i - объем приходящийся на i -е помещение (абонента).

В случае если индивидуальный прибор учета имеется, величина потребления определяется как разница между текущими и предыдущими показаниями.

$$V_i = \sum_{j=1}^n C_{ij} - P_{ij}, \quad (2)$$

где n – количество индивидуальных приборов учета по заданному ресурсу;

C_{ij} - текущие показания по j -му прибору учета i -го абонента;

P_{ij} - предыдущие (последние учтенные) показания по j -му прибору учета i -го абонента.

Фактический расчет позволяет более точно отразить реальное потребление абонентов. И ресурсоснабжающие организации и абоненты в этом напрямую заинтересованы [4]. В тех случаях, где возможно применение индивидуальных приборов учета, переход к фактическому расчету всячески стимулируется, в том числе, вводом повышающих коэффициентов k для нормативного расчета:

$$V_i = (1 + k) * S_i \times N. \quad (3)$$

Для некоторых коммунальных ресурсов, в первую очередь, для электроэнергии в настоящее время предлагается в качестве индивидуальных приборов учета использовать счетчики, обладающие возможностями автоматической передачи показаний в РСО (системы АСКУЭ). Такие приборы учета способны значительно повысить прозрачность потребления ресурсов и снять с абонента необходимость самостоятельно передавать показания. Однако, на текущий момент доля таких приборов учета не значительна, а в соответствии с текущими планами модернизации полная замена приборов учета произойдет не ранее 2030 года. Доля автоматизированных счетчиков других коммунальных ресурсов – еще ниже. Соответственно, на текущий момент, а так же в ближайшем будущем, основным источником показаний будут являться сами абоненты или контролеры. Если показания контролеров можно считать достоверными, хотя человеческий фактор полностью исключать нельзя, то абонент может ошибаться, умышленно занижать или завышать показания, или не передавать их совсем. Все эти факторы могут приводить к тому, что объем потребления агрегированный в информационной системе может отличаться от реально потребленного даже в случае работы с индивидуальными приборами учета. Контрольные обходы частично выправляют ситуацию, однако

они довольно редки и возможности выполнять их ежемесячно у РСО нет. В случае отсутствия показаний индивидуальных приборов учета, Постановление Правительства, регламентирующее расчет, предполагает использование в расчете средних значений потребления.

$$V_{i,t} = \frac{1}{m} \sum_{k=1}^m V_{i,t-k}, \quad (4)$$

где t – порядковый номер месяца, за который выполняется расчет;

m – период, за который определяется средний объем, как правило 6 или 12 месяцев.

Данный алгоритм предполагает в течение некоторого начального периода начисление скользящего среднего арифметического объема, а затем использование норматива.

Очевидно, не смотря на то, что с точки зрения законодательства рассчитывать объемы и суммы необходимо именно таким образом, и именно эти значения должны фигурировать в бухгалтерских отчетах, с точки зрения аналитики и исследования трендов эти значения не являются истинными и могут вносить искажения, в частности, особо сильное влияние может быть оказано на сезонный тренд. Для достижения достоверности аналитического прогноза правильным решением будет являться отказ от использования средних и, тем более, нормативных значений и применение методов интерполяции для восстановления значений в каждом расчетном периоде.

В соответствии с правилами предоставления коммунальных услуг, расчетным периодом является месяц, но показания приборов учета фиксируются на определенную дату. В течение расчетного периода допускается произвольное количество фактов съема показаний. При этом объем потребления определяется как разность между текущими показаниями, и показаниями предыдущей фиксации. Очевидно, что такой объем непосредственно связан с двумя датами съема показаний, которые в свою очередь могут относиться к разным расчетным периодам. Для составления аналитических отчетов необходимо этот объем разложить на составляющие, относящиеся к отдельно взятым расчетным периодам. Способствовать реализации этого требования может учет показаний непосредственно на границе месяцев. Исходя из того, что на практике этого достичь практически невозможно, решением этой проблемы также является использованием методов интерполяции.

Учитывая тот факт, что на момент составления прогноза имеются точные значения показаний в определенные временные точки, и в общем виде эти точки на временной оси распределены неравномерно, в качестве метода восстановления значения точки в определенный промежуток времени может использоваться интерполяционная формула Лагранжа (5).

$$L_n = \sum_{i=0}^n y_i \frac{(x-x_0)(x-x_1)\dots(x-x_{i-1})(x-x_{i+1})\dots(x-x_n)}{(x_i-x_0)(x_i-x_1)\dots(x_i-x_{i-1})(x_i-x_{i+1})\dots(x_i-x_n)}. \quad (5)$$

Для проведения эксперимента использовался следующий подход. Для анализа были взяты реальные данные о потреблении коммунальных ресурсов абонентов с индивидуальными приборами учета, осуществляющих регулярную передачу показаний. Таким образом, средних и нормативных объемов потребления у таких абонентов нет, потребление и показания таких абонентов заранее определены. Однако, так как формулы для расчета среднего и нормативного известны, все данные для расчета имеются, появляется возможность получить расчетные значения величины потребления абонентами ресурса в случае отсутствия передачи показаний. Эти же значения предполагается рассчитать, применив методы интерполяции. Суть эксперимента сводится к моделированию ситуации, в которой передача показаний прекращается, расчетные значения будут определяться по правилам бухгалтерского учета и с помощью применения численных методов интерполяции. Результаты вычислений будут сравниваться с эталонными значениями. Для моделирования используются данные за по-

следние два года. Так как функция величины показаний от времени является монотонно возрастающей, для сравнения результатов нагляднее перейти к зависимости величины потребления (объему) от периода учета, определяемой по формуле 2. Результаты моделирования по абоненту с медианным потреблением ресурса представлены на рисунке.

Анализ показывает, что средние значения за первые 3 месяца близки к эталонным значениям, но при этом превышают их. Причиной может являться тот факт, что при расчете среднего учитывается объем за последние 6 месяцев, в данном случае и месяц, предшествующий повышению тарифов. Как правило, абоненты завышают объем, потребленный в этом периоде, что сказывается на расчете среднего в дальнейшем.

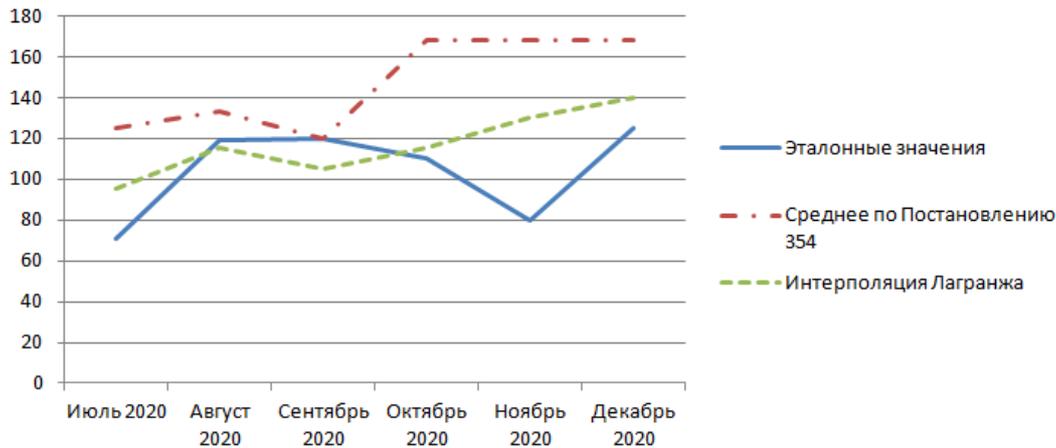


Рис. Зависимость величины потребления (КВт) от периода учета

Переход на нормативы по истечению 3 месяцев приводит к явному завышению объема. При этом данные, полученные с помощью интерполяции, оказываются существенно ближе к эталонным. Это объясняется, в том числе, выбором исходных точек для выполнения расчета, в периоды не содержащие экстремального потребления. Таким образом, для реальной оценки потребления коммунального ресурса, в случае отсутствия передачи показаний приборов учета, использование данных, полученных методами интерполяции, позволяет работать с меньшей погрешностью, чем работа с данными, полученными в соответствии с нормативными документами, регламентирующими расчет. Следовательно, такой подход, особенно в сочетании с методами работ [4 - 9], позволяет строить более точные аналитические прогнозы.

Библиографический список

1. Анисеев С. В., Костиков М. Г., Маркин А. В. Математическая модель определения активных услуг на основе матричной алгебры // Вестник Рязанского государственного радиотехнического университета. 2015. №54. С. 139-144.
2. Костиков М. Г. Моделирование операции расчета стоимости коммунальной услуги при использовании двухкомпонентных режимов потребления // Вестник Рязанского государственного радиотехнического университета. 2019. №68. С. 90-97.
3. Костиков М. Г. Миронов В. В. К расчету стоимости жилищно-коммунальных услуг при использовании общедомовых приборов учета в муниципальных расчетных системах // III международный научно-технический форум СТНО -2020. Сборник трудов. Том 3. – РГРТУ, 2020. С. 111-114
4. Костиков М. Г. Миронов В. В. Принципы реализации расчета стоимости жилищно-коммунальных услуг при использовании общедомовых приборов учета в информационной системе // Инженерно-строительный вестник Прикаспия: научно-технический журнал / Астраханский государственный архитектурно-строительный университет. Астрахань : ГАОУ АО ВО «АГАСУ», 2020. № 4 (34). С. 133–139.
5. Головастов С.В., Коробов А.Е., Миронов В.В. Математические методы исследования теплообмена в тепловых трубах. // Вестник РГРТУ. 2014. № 1 (Вып. 47). С. 120-125.

6. Миронов В.В. Датчики, системы контроля и диагностики воздействия космического мусора на космические аппараты // Датчики и системы. ИПУ РАН. 2014, № 9. С. 2-9.

7. Миронов В.В., Миронова К.В. Проектирование обработки корпоративных данных на основе системы бизнес-анализа QlikView /Современные технологии в науке и образовании /Сборник трудов III межд. науч.-техн. и науч. метод. конф. – Рязань, BookJet, 2018. Т. 5. С. 101 – 108.

8. Аникеев С.В., Миронов В.В. Методологические основы построения программного комплекса синтеза проектных решений в задачах внедрения информационных систем расчетов с населением за жилищно-коммунальные услуги /Современные технологии в науке и образовании /Сборник трудов (индексирован в РИНЦ) III межд. науч.-техн. и науч. метод. конф. – Рязань, BookJet, 2018. Т. 4. С. 224 – 228.

9. Аникеев С.В., Костиков М.Г., Миронов В.В. Информационная система «Абонент +» и опыт ее внедрения на предприятиях ЦФО // Информатизация образования и науки, 2020, № 4 (48), С. 156 – 173.

УДК 004.043; ГРНТИ 20.53.19

АВТОМАТИЗИРОВАННЫЙ РАСЧЕТ СХЕМЫ ОБРАЩЕНИЯ ТКО

А.В. Куликова, А.В. Маркин

*Рязанский государственный радиотехнический университет имени В.Ф. Уткина,
Россия, Рязань, anna.pudova1997@gmail.com*

Аннотация. Рассматривается разработка программного комплекса, позволяющего выполнять расчет нормативных объемов твердых коммунальных отходов (ТКО) и на основе полученных данных с помощью графовой системы управления базами данных Neo4j модифицировать схему обращения с ТКО.

Ключевые слова: Neo4j, схема обращения с ТКО, граф.

AUTOMATED CALCULATION OF THE MSW CIRCULATION SCHEME

A.V. Kulikova, A.V. Markin

*Ryazan State Radio Engineering University named after V. F. Utkin, Ryazan,
Russia, anna.pudova1997@gmail.com*

Annotation. The development of a complex of information systems is considered that allows calculating the standard volumes of solid municipal waste (MSW) and, based on the data obtained, using the Neo4j graph database management system, to modify the MSW management scheme.

Keywords: Neo4j, the scheme of treatment of MSW, graph.

Цель работы

На сегодняшний день одним из актуальных направлений внедрения информационных технологий является автоматизация территориального планирования. Территориальное планирование – это планирование размещения объектов городской и сельской местности для государственных и муниципальных нужд, зон с особыми условиями использования территорий [1].

Согласно постановлению Правительства РФ требуется в каждом регионе разработать и утвердить правила обращения с отходами и организовать общую структуру их вывоза и утилизации с помощью единого оператора по обращению с ТКО [2]. Поскольку понятие единый региональный оператор существует относительно недавно, текущее расположение контейнеров представляет собой совокупность схем, используемых ранее различными операторами.

Целью настоящей работы является автоматизация территориального планирования в сфере обращения ТКО с помощью графовой СУБД Neo4j.

Обзор существующих решений

В настоящее время при территориальном планировании используются следующие методы [2]:

- 1) экспертных оценок;
- 2) нормативный;
- 3) экономико-статистический; программно-целевой;
- 4) экстраполяция;
- 5) экономико-математический;
- 6) балансовый;
- 7) районирование и зонирование территорий.

Выполнение этих методов осуществляется с помощью различных систем, например [3]:

- 1) сервис Zig-Zag (2019 год);
- 2) программный модуль АСУ «Управление отходами», предназначенный для использования на Android устройствах компании РОСИНФА;
- 3) программное обеспечение «экофера-учет отходов» общества с ограниченной ответственностью «КомЭко» (2019 год) и другие.

Однако их применение не позволяет оценить существующее расположение объектов и формировать эффективные рекомендации для его изменения.

Постановка задач

Поскольку поставленная цель перекликается с несколькими известными прикладными задачами, решаемыми с помощью методов теории графов, именно они выбраны в качестве механизмов модификации схемы обращения с ТКО. Для достижения поставленной цели требуется решить следующие задачи:

- 1) обзор методов территориального планирования и механизмов их реализации;
- 2) обзор и анализ методов теории графов и программных средств обработки информации на их основе;
- 3) выявление источников исходных данных, их получение и обработка;
- 4) разработка программного комплекса автоматизированного построения графовой модели схемы обращения с ТКО;
- 5) построение графа существующей схемы обращения с ТКО;
- 6) разработка модели схемы обращения с ТКО на основе методов теории графов;
- 7) совмещение существующей и расчетной графовых моделей с географической картой;
- 8) разработка рекомендаций по расположению объектов ТКО.

Архитектура комплекса

Основными компонентами разрабатываемого комплекса являются (рис. 1):

- 1) источники исходных данных:
 - региональный оператор - файл с текущим расположением и характеристиками контейнеров ТКО, загружаемый в базу данных СУБД Neo4j [4]. Графовая база данных объектов ТКО, хранящая информацию о географических координатах и характеристиках контейнеров;
 - ресурсоснабжающие организации - база данных с характеристиками домов и нормативами для расчета. Реляционная СУБД Firebird и база данных числа жителей и домов. Информация по абонентам и домам обновляется каждый раз при запуске расчета объема ТКО и количества контейнеров в приложении;
 - географические координаты домов с API-сервисов и web-сайтов. Документо-ориентированная СУБД MongoDB и база данных адресов и географических координат домов. Данные обновляются раз в неделю. Триггер по загрузке данных с API сервисов и сайтов запущен в планировщике задач Windows.

2) система мониторинга и обработки географических координат и адресов объектов из внешних систем. Система позволяет решить следующие задачи:

- получать данные с сервисов;
- выполнять парсинг координат;
- реализовывать геокодирование;
- сохранять данные в базу данных СУБД MongoDB;

3) система расчета нормативного объема ТКО и необходимого количества контейнеров, а также проектирования схемы обращения. Система позволяет решить следующие задачи:

- рассчитывать граф текущего расположения объектов ТКО;
- рассчитывать нормативный объем ТКО по городу и на каждый дом;
- рассчитывать количество контейнеров по городу и на каждый дом;
- разрабатывать модель схемы обращения с ТКО на основе методов теории графов;
- предоставлять рекомендации по расположению объектов ТКО.

Структура комплекса представлена на рисунке 1.

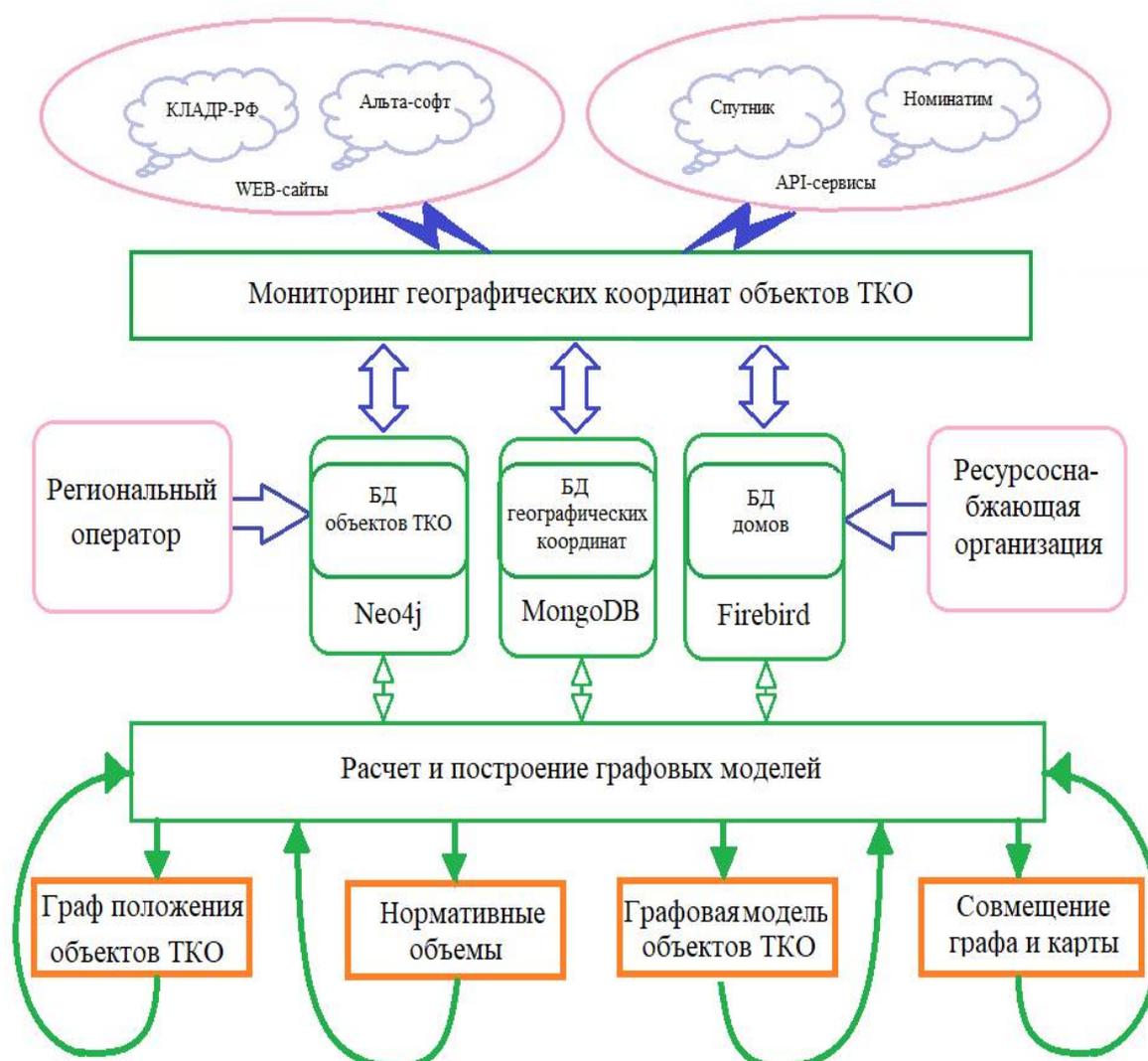


Рис. 1. Структура комплекса

На рисунке 1 розовым и фиолетовым цветом обозначены источники информации. Зеленым цветом выделены компоненты комплекса, оранжевым – основной результат работы

комплекса. Разработанный комплекс позволяет решать выше поставленные задачи. Комплекс поддерживает следующие виды обменов:

- 1) между компонентами комплекса и ресурсоснабжающей организации осуществляется онлайн-обмен. Ресурсоснабжающая организация предоставляет данные по характеристикам домов и количества жителей. Данные для расчетов всегда поддерживаются актуальными;
- 2) между компонентами комплекса и региональным оператором – ROLAP (реляционные OLAP) технологии;
- 3) между компонентами комплекса и API-сервисами получение данных по шлюзу.

На основе этих данных и информации от ресурсоснабжающих организаций по домам построен граф расположения домов г. Рязани и прикрепленным к ним контейнерам. Для каждого микрорайона выделен свой цвет узлов.

На рисунке 2 представлена детализация до одного дома и закрепленных за ним контейнеров. Желтый узел – это дом, красные – контейнеры.

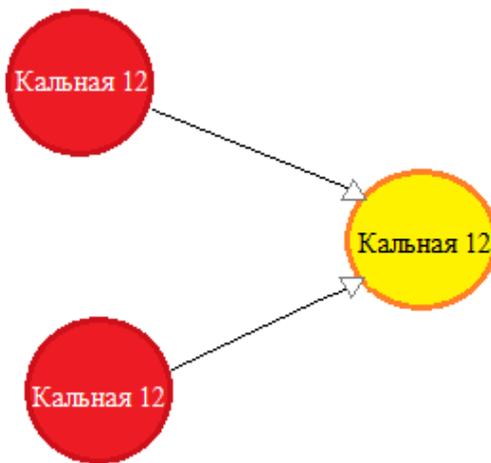


Рис. 2. Детализация схемы обращения с ТКО

Для расчета количества контейнеров и объема ТКО для одного дома используются следующие математические выражения [1, 2].

Объем накопления ТКО в сутки, м³:

$$C = (P * N_{\Gamma} * K_{\text{н}}) / 365, \quad (1)$$

где P – количество жильцов многоквартирного дома;

N_{Γ} – годовая норма накопления ТКО на одного проживающего в доме, м³/чел.;

$K_{\text{н}}$ – нормативный коэффициент накопления отходов 1,25.

Количество контейнеров, шт.:

$$M = (C * T * K_{\text{р}}) / (V * K_{\text{з}}), \quad (2)$$

где $V_{\text{к}}$ – суточный объем мусора, м³;

T – максимальное время накопления ТБО, дни;

$K_{\text{р}}$ – корректировочный коэффициент, учитывает заполнение бака повторно мусором, оставшимся после выгрузки. $K_{\text{р}} = 1,05$;

$K_{\text{з}}$ – коэффициент, накопления емкости отходами не до верха, а на три четверти (0,75).

На рисунке 3 приведен фрагмент схемы ТКО по рассчитанным объемам и характеристикам для того же дома, что и на рисунке 2. Как следует из графа, по рассчитанным объемам (рис. 3) для дома 12 на улице Кальная нужно добавить контейнер, что позволит избежать потенциального переполнения баков и дополнительных рейсов мусоровозов.

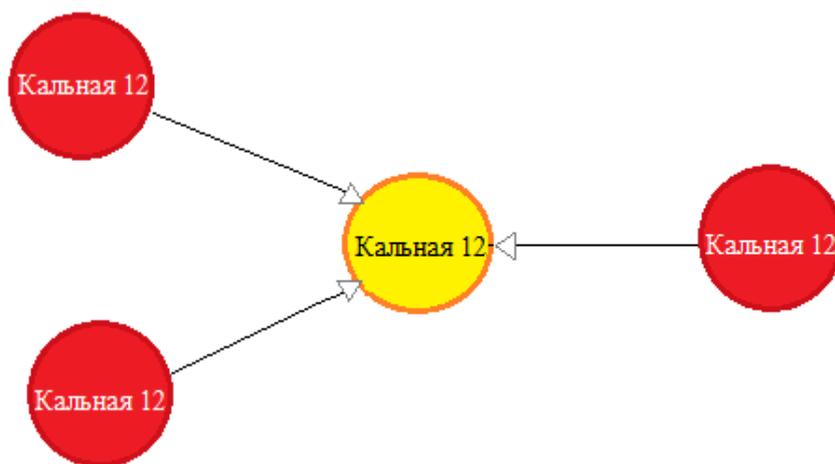


Рис. 3. Рассчитанный граф

На текущий момент не решена только задача 7 из вышеперечисленных. В дальнейшем планируется выполнить совмещение графа и карты города, а также выполнить изменение рассчитанного расположения контейнеров с помощью методов теории графов научной библиотеки СУБД Neo4j. Для модификации схемы обращения с ТКО выбраны следующие методы теории графов:

- 1) центральности и ранжирования, позволит найти дороги и перекрестки с наибольшей пропускной способностью;
- 2) кластеризация, выполнит разделение узлов по микрорайонам города;
- 3) нахождение кратчайшего пути, позволит определить наиболее короткий маршрут мусоровоза.

Первый тип – методы центральности и ранжирования. Например, по атрибутам узлов с помощью метода максимального потока можно получить перекресток с наибольшей пропускной способностью. На рисунке 4 представлен запрос и результат его выполнения.

```

CALL
algo.closeness('Перекресток',
'Улица', {write:true,
stats:true,
writeProperty:'centrality'})
YIELD nodes, loadMillis,
computeMillis, writeMillis
Перекресток БД с наибольшей
пропускной способностью
  
```

а.имя	а.centrality
"Мин. Соцзащиты"	0.33980582524 271846

Рис. 4. Результат нахождения перекрестка с наибольшей пропускной способностью

Второй тип – кластеризация. На слайде приведены примеры сообществ графа, разделенных запросом к БД по расстоянию между узлами. Признак (атрибут) можно указать любой из имеющихся в БД (рис. 5).

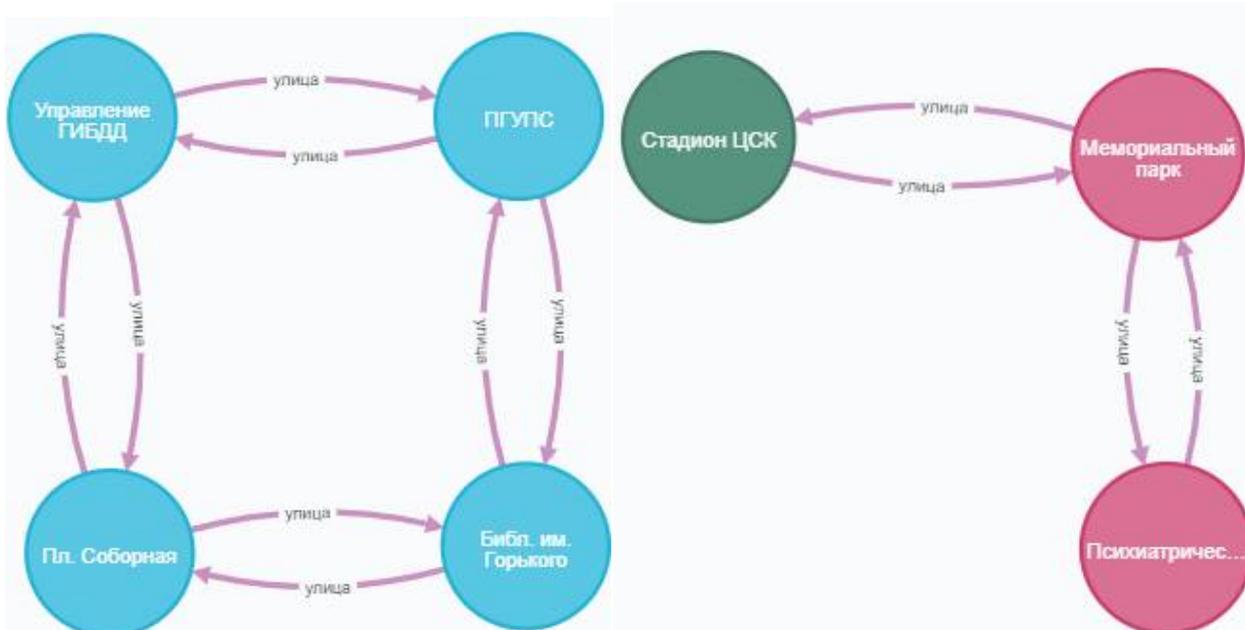


Рис. 5. Результат кластеризации

Третий тип – нахождение кратчайшего пути. Найти кратчайший путь между узлами Гостиница Первомайская и министерство Соц. Защиты (рис. 6).

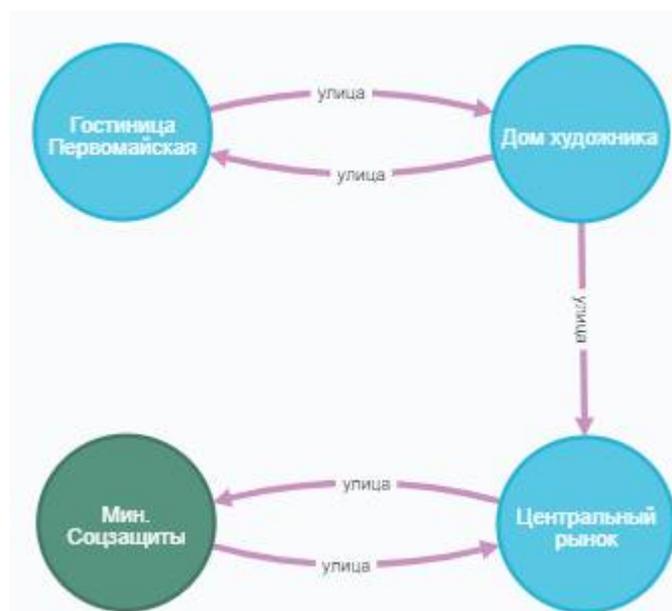


Рис. 6. Результат нахождения кратчайшего пути

Заключение

Представленный программный комплекс позволит рассчитывать граф текущего расположения объектов ТКО и автоматизировать построение и модификацию полученной модели.

Библиографический список

1. Градостроительный кодекс Российской Федерации от 29.12.2004 N 190-ФЗ (ред. от 02.08.2019) (с изм. и доп., вступ. в силу с 13.08.2019) [Электронный ресурс] URL:http://www.consultant.ru/document/cons_doc_LAW_51040/ (дата обращения: 26.01.2021).
2. Постановление Правительства РФ от 22 сентября 2018 г. [Электронный ресурс] URL: <https://base.garant.ru/72060036/> (дата обращения: 06.02.2021).
3. Роспатент открытый реестр [Электронный ресурс] URL: <https://rospatent.gov.ru/ru> (дата обращения 30.01.2021).
4. Neo4j [Электронный ресурс] URL:<https://neo4j.com/>(дата обращения: 25.01.2021).

УДК 004.043; ГРНТИ 20.53.19

РАЗРАБОТКА ТЕХНОЛОГИИ ПОДГОТОВКИ ОТЧЕТНОЙ ДОКУМЕНТАЦИИ В СИСТЕМАХ С РАСПРЕДЕЛЕННЫМ КЭШИРОВАНИЕМ ДАННЫХ

Г.А. Копцев, С.В. Аникеев

Рязанский государственный радиотехнический университет имени В.Ф. Уткина, Россия, Рязань, koptsev.gleb@gmail.com

Аннотация. Рассматривается разработка программного решения, позволяющего выполнять подготовку отчетной документации на основе данных, полученных посредством обмена с системой, осуществляющей распределенное кэширование данных.

Ключевые слова: отчетная документация, система, кэширование.

DEVELOPMENT OF TECHNOLOGY FOR PREPARING REPORTING DOCUMENTATION IN SYSTEMS WITH DISTRIBUTED DATA CACHING

G.A. Koptsev, S.V. Anikeev

Ryazan State Radio Engineering University named after V. F. Utkin, Ryazan, Russia, koptsev.gleb@gmail.com

Annotation. The development of a software solution is considered that allows you to prepare reporting documentation based on data obtained through exchange with a system that implements distributed data caching.

Keywords: reporting documentation, system, caching.

Цель работы

В современном развивающемся мире эффективное функционирование каждой организации, независимо от ее профиля, находится в прямой зависимости от уровня оперативности обработки информации и создания документации, скорости взаимодействия между структурными подразделениями и контрагентами. Поэтому автоматизация, высокий уровень которой достигается за счёт использования прикладного программного обеспечения, является одной из первостепенных задач современных компаний.

Существует множество решений для автоматизации работы предприятий в самых различных областях жизнедеятельности. Одним из таких решений является расчетная система «Абонент» в сфере ЖКХ.

Расчетная система «Абонент» предназначена для решения расчетно-учетных задач, связанных с поставкой населению энергоресурсов и оказанием услуг ЖКХ [1].

Система имеет широкий спектр функциональных возможностей, одной из которых является способность создания отчетной документации. С течением времени в процессе эксплуатации этой функции пользователи начали сталкиваться с рядом неудобств, и появилась потребность в переработке технологии подготовки отчетной документации.

Целью настоящей работы является повышение уровня производительности и комфорта для пользователя в процессе подготовки отчетной документации.

Обзор существующего решения

В настоящее время работа с отчетной документацией в клиенте расчетной системы Абонент осуществляется посредством взаимодействия с реляционной базой данных с помощью SQL-запросов. Такой подход в процессе активной эксплуатации и наполнения базы данных большим количеством информации сталкивается с проблемой масштабируемости реляционных БД. В большинстве случаев формирование отчета требует выполнения ресурсоемких запросов, которые сильно влияют на производительность системы в целом и приводят к частой блокировке других транзакций, что делает параллельную работу системы с помощью других клиентов расчетной системы крайне медлительной.

На рисунке 1 представлена текущая схема взаимодействия пользователя с системой.



Рис. 1. Модель взаимодействия пользователя с РС

Постановка задач

Для решения проблемы уменьшения производительности базы данных разработчиками расчетной системы Абонент было разработано решение, кэширующее имеющиеся данные в базе в оперативную память и представляющее их в виде объектной модели, позволяющее получать информацию посредством HTTP-запросов. Такой подход позволяет переработать технологию подготовки отчетной документации и уйти от ряда недостатков.

Таким образом, необходимо разработать решение, позволяющие использовать преимущества сервиса распределенного кэширования в технологии подготовки отчетов.

Разрабатываемый проект представляет собой веб-приложение, которое решает следующие задачи:

- 1) отображение доступных для формирования отчетов;
- 2) получение наборов данных для формирования отчетов;
- 3) возможность использования различных типов пользовательских параметров;
- 4) генерация отчета.

Модель взаимодействия разрабатываемого решения с сервисом распределенного кэширования данных представлена на рисунке 2.

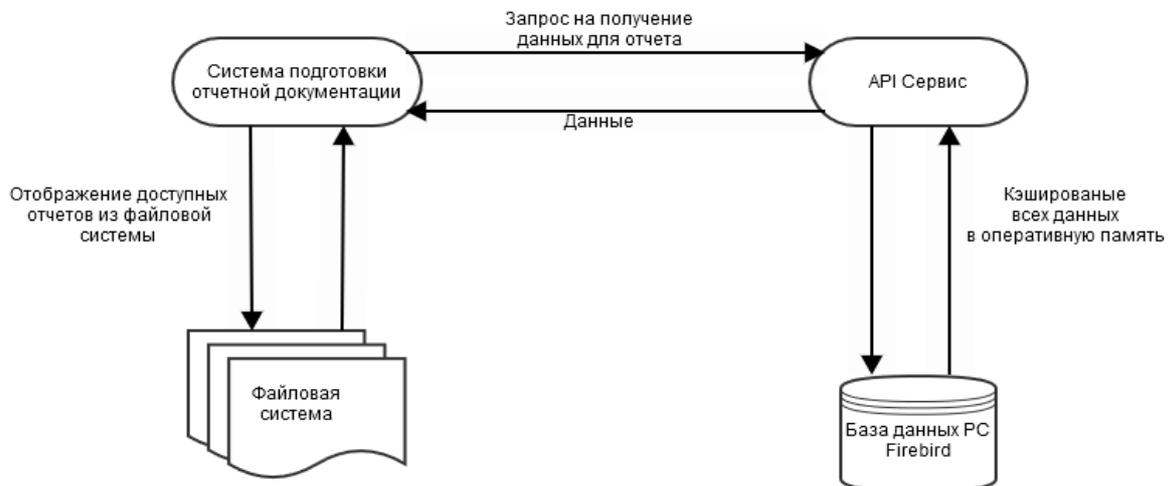


Рис. 2. Модель взаимодействия

Отображение доступных для формирования отчетов

Источником доступных для формирования отчетов является файловая система. В качестве исходной точки указывается каталог в файловой системе, а приложение анализирует наличие в подкаталогах специальных файлов – манифестов отчетов (рисунок 3).

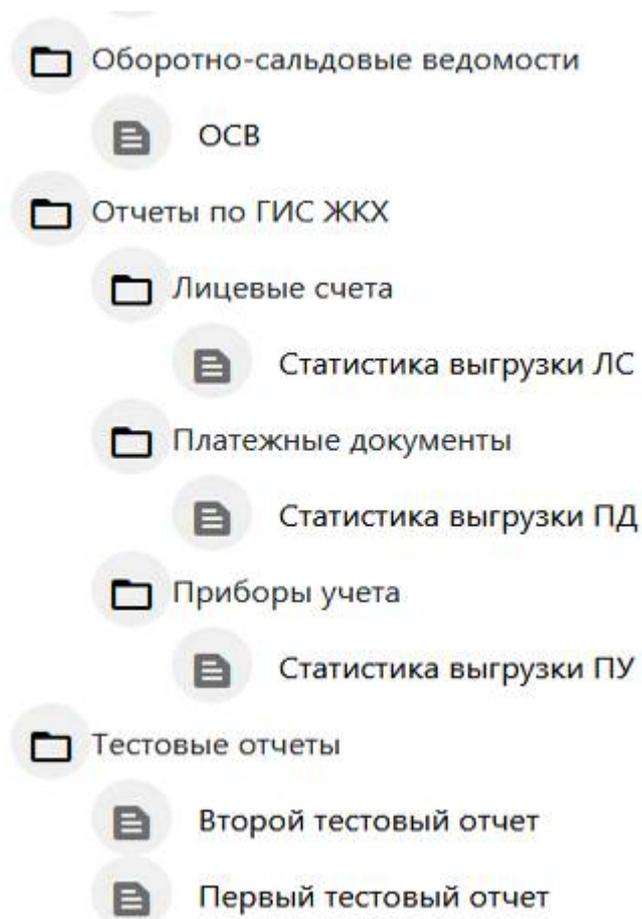


Рис. 3. Иерархия манифестов в файловой системе

Манифест отчета содержит всю информацию, необходимую для формирования документа и содержит (рисунок 4):

- 1) название;
- 2) путь к шаблону отчета;
- 3) информация по получению наборов данных и их параметры.

```

{
  "Name": "Новый отчет",
  "TemplatePath": "ReportTree\\Тестовые\\testTemplateJson.frx",
  "DataSets": [
    {
      "Query": "Abonents.Values.Skip({0}).Take({1}).ToArray()",
      "Name": "SimpleDataset",
      "Params": [
        {
          "Name": "Skip",
          "Required": true,
          "Value": null,
          "ParamType": 0,
          "Query": null,
          "KeyValueList": null
        },
        {
          "Name": "Take",
          "Required": true,
          "Value": null,
          "ParamType": 0,
          "Query": null,
          "KeyValueList": null
        }
      ]
    },
    {
      "Query": "Abonents.Values.Where(a => a.Pays.Any(p => new string[]{{0}}.Contains(p.BundleOfPays.Analytics[BundleOfPaysAnalytics.SourceId]))).ToArray()",
      "Name": "ComplexDataset",
      "Params": [
        {
          "Name": "SourceDocCD",
          "Required": true,
          "Value": null,
          "ParamType": 5,
          "Query": "Params.SourceDocs.Values.ToDictionary(k => k.SourceDocCD, v => v.SourceDocNM)",
          "KeyValueList": null
        }
      ]
    }
  ]
}

```

Рис. 4. Структура манифеста отчета

Такой подход позволяет пользователем создавать собственные иерархии отчетов, создавая и размещая папки с манифестами по своему усмотрению. Поддерживается любая уровень вложенности.

Отчеты являются кликабельными, и при нажатии на них из манифеста подгружается список пользовательских параметров для ввода (рисунок 5).

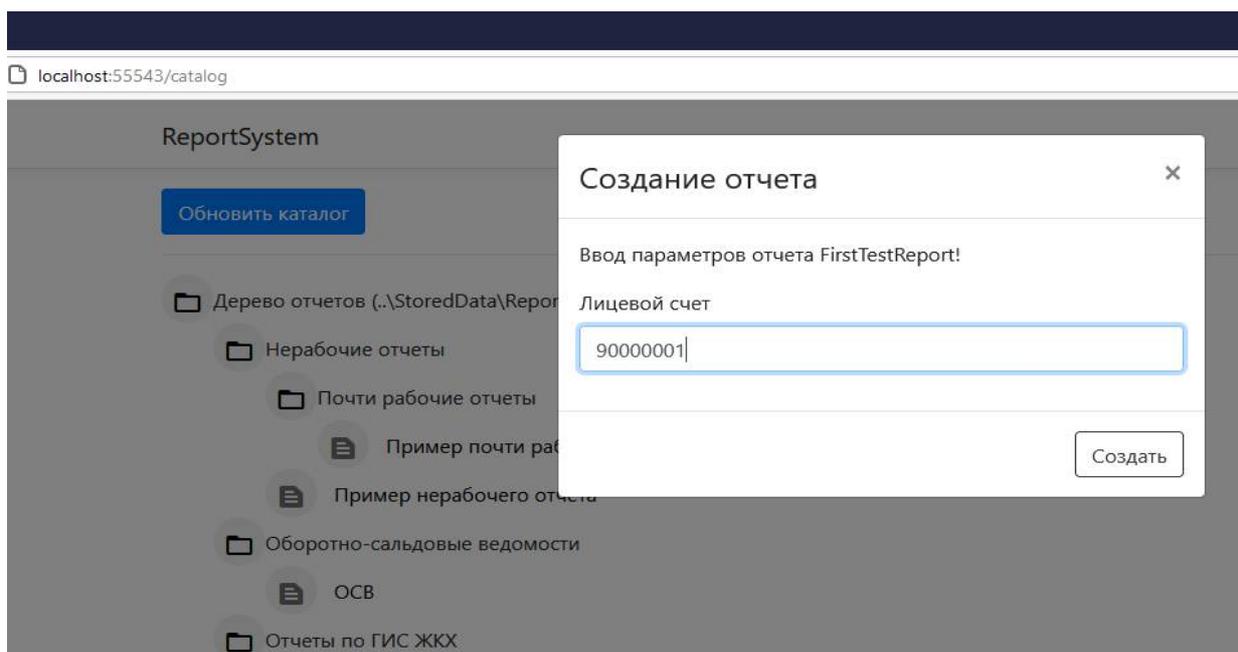


Рис. 5. Ввод параметров

Получение наборов данных для формирования отчетов

В манифестах, рассмотренных ранее, присутствует информация для получения наборов данных: запрос и пользовательские параметры.

Как упоминалось ранее, сервис распределенного кэширования данных хранит данные в оперативной памяти в виде объектной модели, поэтому запрос для набора данных фактически представляет собой обращение на языке C# к глобальному массиву с данными. В запросах используется технология LINQ – интегрированная в язык программирования технология запроса к источнику данных [2]. Перед непосредственной отправкой HTTP-запроса к сервису системой подготовки документации запрос наполняется введенными пользователем параметрами через интерфейс сайта. На приемной стороне запрос обрабатывается и в качестве ответа возвращается JSON с необходимыми данными.

Полученные наборы данных регистрируются в качестве источников данных при генерации отчета. В качестве генератора отчетов используется Fastreport.

Возможность использования различных типов пользовательских параметров

Система позволяет использовать параметры различных типов данных. В зависимости от выбранного типа поле для ввода параметров примет соответствующий вид (рисунок 6).

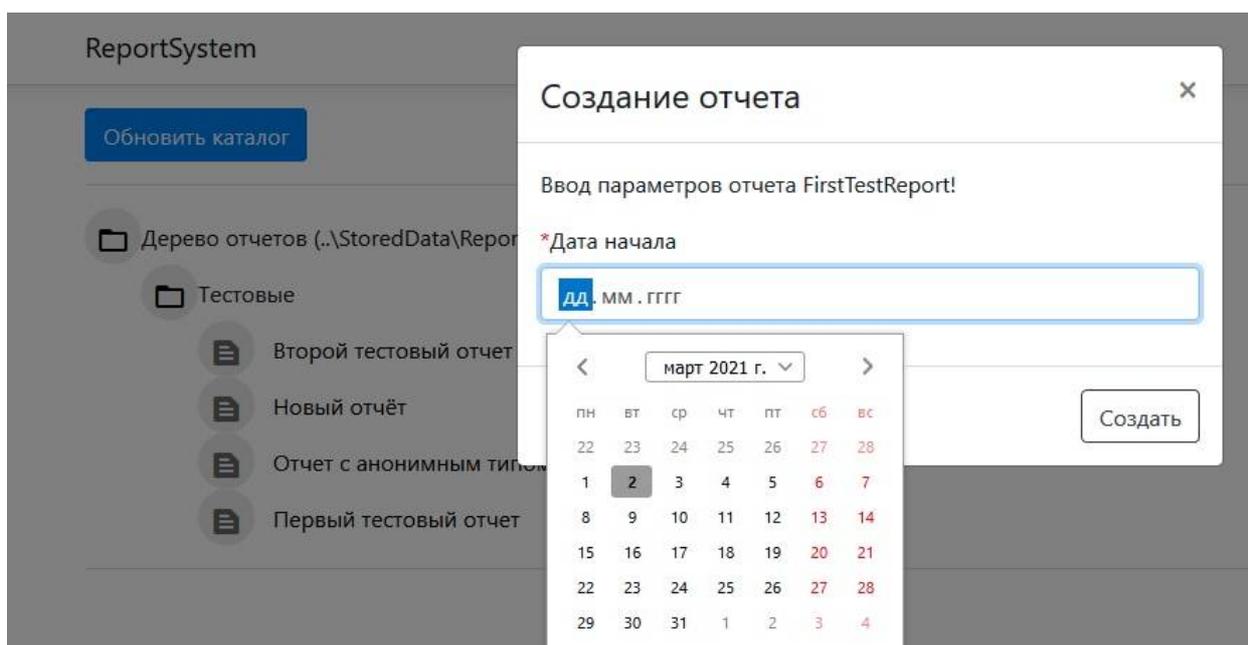


Рис. 6. Ввод параметра типа «Дата»

Помимо простых параметров, которые требуют лишь ввода значения, предусмотрено использование комплексных параметров – выбор значения из списка. В качестве таких параметров могут выступать отдельные таблицы базы данных расчетной системы Абонент. Для таких параметров, как и для наборов данных, требуется указать LINQ-запрос, с помощью которого, обратившись по HTTP-протоколу к сервису распределённого кэширования данных, можно получить необходимый список параметров. Результатом отработки LINQ-запроса для параметра должен являться массив типа “ключ-значение” (рисунок 7).

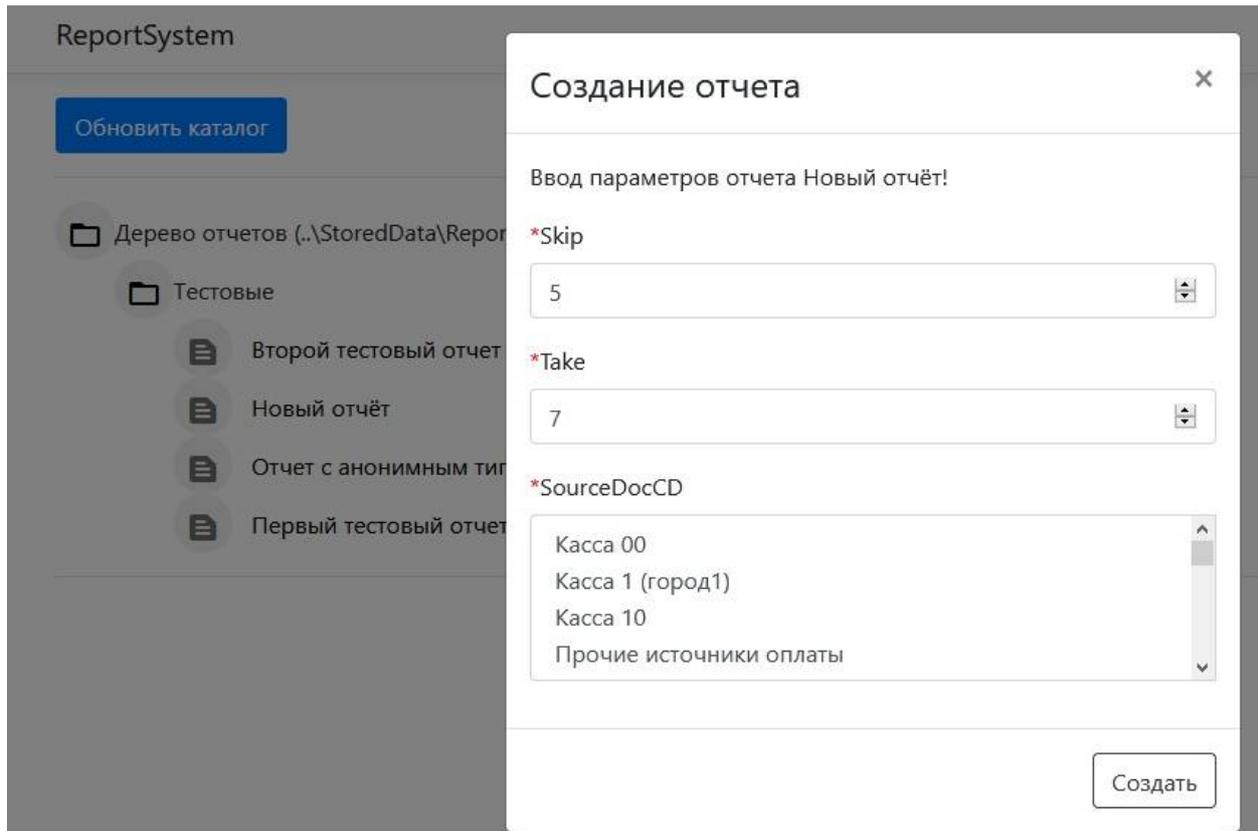


Рис. 7. Ввод составного параметра

При загрузке окна ввода параметров приложение анализирует наборы данных в манифесте на наличие комплексных параметров и в случае их наличия формирует HTTP-запросы на получение данных для них.

Заключение

Таким образом, поставленная задача выполнена, было разработано средство, значительно ускоряющее процесс подготовки отчетной документации в расчетной системе Абонент и реализующее функции, рассмотренные ранее.

Библиографический список

1. Абонент+ Расчетно-платежный комплекс [Электронный ресурс] URL: https://www.abonent.plus/product_01 (дата обращения: 06.02.2021).
2. LINQ (C#). [Электронный ресурс] URL: <https://docs.microsoft.com/ru-ru/dotnet/csharp/programming-guide/concepts/linq/> (дата обращения: 06.02.2021).

УДК 004.043; ГРНТИ 20.53.19

РАЗРАБОТКА ИНФОРМАЦИОННОЙ МОДЕЛИ ВЕДЕНИЯ ПРЕТЕНЗИОННО-ИСКОВОЙ РАБОТЫ В СФЕРЕ ЖКХ

Д.А. Андреев, С.В. Аникеев

*Рязанский государственный радиотехнический университет имени В.Ф. Уткина,
Россия, Рязань, andreevda97@mail.ru*

Аннотация. Рассматривается модель информационной системы, позволяющей эффективно выполнять поэтапное ведение претензионно-исковой работы с потребителями жилищно-коммунальных услуг.

Ключевые слова: сети Петри, претензионно-исковая работа, дебиторская задолженность, ЖКХ.

DEVELOPMENT OF AN INFORMATION MODEL JUDICIAL AND PRE-TRIAL WORK IN THE FIELD OF HOUSING AND COMMUNAL SERVICES

D.A. Andreev, S.V. Anikeev

*Ryazan State Radio Engineering University named after V. F. Utkin, Ryazan,
Russia, andreevda97@mail.ru*

Annotation. The article considers a model of an information system that makes it possible to effectively perform step-by-step management of claims and claims work with consumers of housing and communal services.

Keywords: Petri Nets, judicial and pre-trial work, accounts receivable, housing and communal services.

Цель работы

Претензионно-исковая работа – основополагающая доля успешного ведения деятельности в организации ЖКХ, которая предполагает подготовку судебных и досудебных документов. В связи со спецификой работы выделяется отдельная юридическая должность. Без эффективной деятельности по возврату дебиторской задолженности нельзя представить нормальное функционирование организации ЖКХ. Проблемы данного характера незамедлительно отражаются на качестве услуг, а соответственно и качеству жизни граждан [1].

Целью проекта является разработка автоматизированной системы учета исков для повышения эффективности ведения претензионно-исковой работы с потребителями жилищно-коммунальных услуг. Достижение цели обеспечивается за счет возможности оперативного выполнения большинства однотипных задач по подготовке и формированию комплекта судебных и досудебных документов в едином сервисе.

Обзор существующих решений

В настоящее время большинство распространяемых программных продуктов нацелено в основном на работу с юридическими лицами и составление документов для арбитражных судов, а также на повышение прозрачности и управляемости процесса, при этом, степени его автоматизации недостаточно (таблица 1). В результате обнаружены два коммерческих решения, которые применяются для выполнения работы с дебиторской задолженностью по физическим лицам – клиентам организаций ЖКХ.

Таблица 1. Результаты сравнительного анализа существующих решений

	RKIT Group	Юрайт	Avis Expert	DigitalDesign	Платформа	NetSystems
Ведение работы с потребителями ЖКУ	–	–	–	–	+	+
Веб-приложение	–	–	–	–	–	+
Единый реестр претензий и исков для разных подразделений	+	+	–	+	+	–
Авто-обмен с биллинговой системой	–	–	–	+	+	–
Калькулятор расчета госпошлины и задолженности	–	+	–	–	+	+
Претензионно-исковая работа с потребителями ЖКУ	–	–	–	–	+	+
Формирование документов с автоматической подстановкой полей	–	–	–	+	+	+
Предлагает пользователю эффективную меру воздействия	+	–	–	+	+	–
Возможность оповещения должников	–	–	–	–	–	+
Аналитические отчеты	+	+	+	+	+	+

Недостатками использования существующих решений является невозможность их дальнейшей индивидуальной доработки для достижения максимальной эффективности при ведении претензионно-исковой работе с потребителями жилищно-коммунальных услуг.

Основным отличием разрабатываемой автоматизированной системы учета исков будет являться автоматическая компоновка и формирование юридических документов, в зависимости от особенностей учета, состава семьи, количества жилищно-коммунальных услуг и подобных характеристик, что в значительной степени ускорит процесс составления пакета документов для предъявления в государственные органы и суды. При этом, работа с должником будет иметь поэтапный характер, где применяемая мера воздействия зависит от состояния дебиторской задолженности контрагента и ранее выполненных действий.

Модель поэтапного ведения претензионно-исковой работы

Претензионно-исковая работа не ограничивается разовой подготовкой документов, существуют различные этапы досудебной и судебной работы. После подачи заявления в суд образуется множество вариантов исхода, например, отмена судебного приказа или отказ во взыскании, после которых необходимо выполнять соответствующие операции и формировать новый комплект документов. Таким образом, в разрабатываемой системе должна быть заложена модель поэтапного ведения работы.

Сети Петри – это инструмент исследования систем. Теория сетей Петри делает возможным моделирование системы математическим представлением её в виде сети Петри, то есть представляет собой механизм формализации процесса моделирования [2].

Сеть Петри формально определяется как четверка вида:

$S = \langle P, T, E, Mo \rangle$, где

P — конечное множество позиций (иначе – мест или условий);

T — конечное множество переходов (событий);

E – конечное множество дуг между позициями и переходами;

Mo – начальная маркировка (разметка).

Графическим изображением сети Петри является двудольный ориентированный граф с двумя типами вершин: вершины P_i (из множества P) изображают обычно окружностями, а вершины T_i (из множества T) – прямоугольниками. Дуги графа могут быть направлены от окружностей только к прямоугольникам, а от прямоугольников – только к окружностям, так что любая позиция может быть входной или (и) выходной позицией одного или нескольких переходов. С помощью сетей Петри можно моделировать широкий класс систем. Наиболее удобны СП при моделировании систем, в которых протекают параллельные взаимодействующие процессы, причем параллельность моделируется естественным и удобным образом [3].

Для построения информационной модели поэтапного выполнения претензионно-исковой работы удачнее всего использовать сеть Петри ввиду параллельного характера обработки и ведения дел должников. В организациях ЖКХ данную работу могут вести одновременно, как юридические, так и абонентские отделы. При этом должники могут быть разбиты на группы по этапам работы, на которых применяются наиболее эффективные меры воздействия.

Принцип поэтапного ведения работы с должниками по жилищно-коммунальным услугам представлен на рисунке 1.

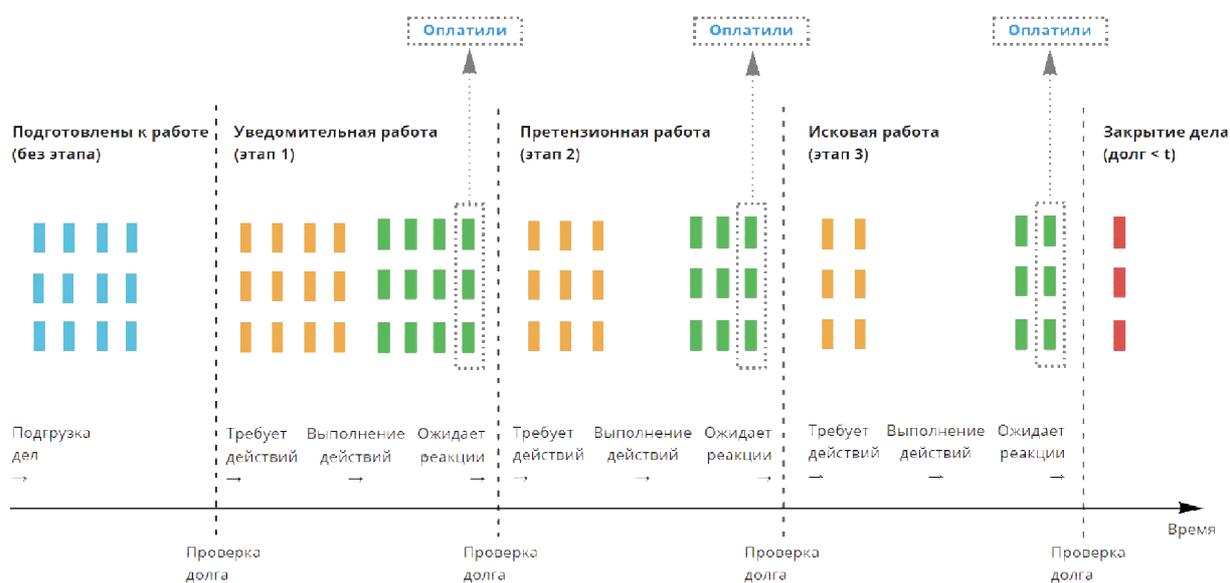


Рис. 1. Концептуальная схема поэтапного ведения работы

В разрабатываемой системе предполагается заведение дел по должникам с дальнейшим переходом по этапам работы, в которых производится выполнение соответствующих действий досудебной и судебной деятельности, своевременно контролируются сроки и поступления оплаты по задолженности. На основе концептуальной схемы ведения претензионно-исковой работы строится информационная модель в виде сети Петри с заданными исходными параметрами (рисунок 2).

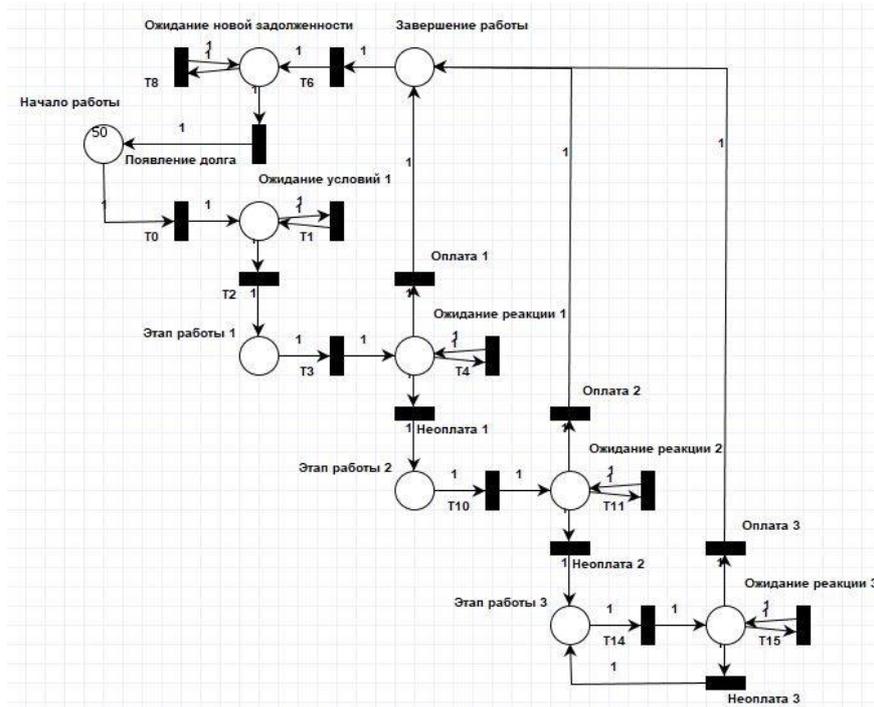


Рис. 2. Сеть прохождения этапов работы с должником

Основные характеристики построенной сети.

1. Не является безопасной – так как не безопасна каждая её позиция (число фишек в ней не равно 1 либо 0).
2. Сеть обладает свойством сохранения – так как общее число фишек в сети является постоянной величиной.
3. Сеть является строго сохраняющей, так как число входов на каждом переходе равно числу выходов.
4. В сети отсутствуют тупики (переходы, которые не могут быть запущены) – следовательно можно говорить об активности всех переходов.

В результате моделирования с исходными параметрами сети выявлено, что при равной вероятности оплаты и неоплаты должника данная сеть успешно справляется с заданной нагрузкой (отсутствуют взаимоблокировки и невыполнимые операции). После запуска 10000 итераций для 50 должников распределение фишек в позициях стало следующим:

- "Ожидание условий 1" – 13;
- "Начало работы" – 11;
- "Этап работы 1" – 10;
- "Завершение работы" – 8;
- "Ожидание новой задолженности" – 5;
- "Этап работы 2" – 2;
- "Этап работы 3" – 1.

Вероятность перехода на следующий этап уменьшается с увеличением количества мер воздействия на должника. Таким образом, на начальных этапах работы всегда остается больше маркеров, чем на последних. Наибольшая концентрация должников остается в позициях "Начало работы", "Этап работы 1", "Ожидание условий 1", "Завершение новой задолженности" и "Завершение работы".

Заключение

Поэтапная модель ведения претензионно-исковой работы позволяет увеличить эффективность мероприятий, направленных на возврат дебиторской задолженности в организациях ЖКХ. В дальнейшем, на основе информационной модели будет реализована автоматизированная система, которая позволит применять комплекс мер воздействий на должников по ЖКХ-услугам, что в значительной степени повлияет на финансовую стабильность организаций ЖКХ, а в следствии чего на уровень и качество жизни всего населения.

Библиографический список

1. Как автоматизировать претензионно-исковую работу // Практический журнал юриста «ЮристВзаконе» [Электронный ресурс] URL: <https://uristvzakon.ru/kak-avtomatizirovat-pretenzionno-iskovuyu-rabotu.html> (дата обращения: 26.01.2021)
2. А.А. Лескин, П.А. Мальцев, А.М. Спиридонов «Сети Петри в моделировании и управлении» Л.: Наука, 1989. – 133 с.
3. Волкова В. Н., Денисов А. А. Теория систем и системный анализ : учебник для вузов : /.— М. : Юрайт, 2010 . – 678 с.

УДК 004.032.26

РАЗРАБОТКА ПОДСИСТЕМЫ ФУНКЦИОНАЛЬНОГО ПРЕОБРАЗОВАНИЯ ЧАСТОТЫ В КОД НА ОСНОВЕ МНОГОСЛОЙНОЙ ИСКУССТВЕННОЙ НЕЙРОННОЙ СЕТИ

А.В. Конькова, С.В. Челебаев

*Рязанский государственный радиотехнический университет имени В.Ф. Уткина,
Российская Федерация, Рязань, ana.konkova@yandex.ru*

Аннотация. В работе рассматривается процесс измерения неэлектрической величины, состоящий из последовательности двух операций: первичного преобразования неэлектрической величины в электрическую и вторичного преобразования электрической величины в цифровой позиционный код. Разработана структура нейросетевого преобразователя. Реализован алгоритм обучения многослойной нейронной сети. Выполнен анализ погрешности преобразования. Выполнено моделирование работы подсистемы на программируемой логической интегральной схеме. Проведен анализ аппаратных затрат для параллельной и последовательной реализации нейронов сети.

Ключевые слова: нейронные сети, обучение, алгоритм обратного распространения, аппроксимация данных, вторичное преобразование.

DEVELOPMENT OF A SUBSYSTEM OF FUNCTIONAL FREQUENCY TO CODE CONVERSION BASED ON A MULTILAYER ARTIFICIAL NEURAL NETWORK

A.V. Konkova, S.V. Chelebaev

*Ryazan State Radio Engineering University named after V.F. Utkin,
Russia, Ryazan, ana.konkova@yandex.ru*

The summary. The paper discusses the process of measuring a non-electric quantity consisting of two operations: primary conversion of non-electrical quantity into electrical and secondary conversion of electrical quantity into digital position code. The structure of the neural network converter has been developed. A multilayer neural network learning algorithm has been implemented. The analysis of the conversion error has been carried out. Simulation of the subsystem operation on a programmable logic device has been made. The analysis of hardware costs for parallel and sequential implementation of network neurons has been made.

Keywords: neural networks, training, backpropagation algorithm, data approximation, secondary transformation.

В промышленности широко используются информационно-измерительные системы (ИИС). Для сбора информации об объекте исследования в ИИС применяются датчики. Датчики преобразуют информацию о теплоэнергетических (температура, давление и т.д.), механических (перемещение, скорость, линейные размеры), физических (плотность, влажность, мутность) и химических (концентрация) параметрах в электрические сигналы, изменение параметров которых пропорционально изменению параметров контролируемых процессов. Выходными сигналами датчиков могут быть амплитуда или частота. Для дальнейшей обработки результатов измерений необходимо использовать вторичный преобразователь, позволяющий трансформировать аналоговую величину в цифровой код (рисунок 1).

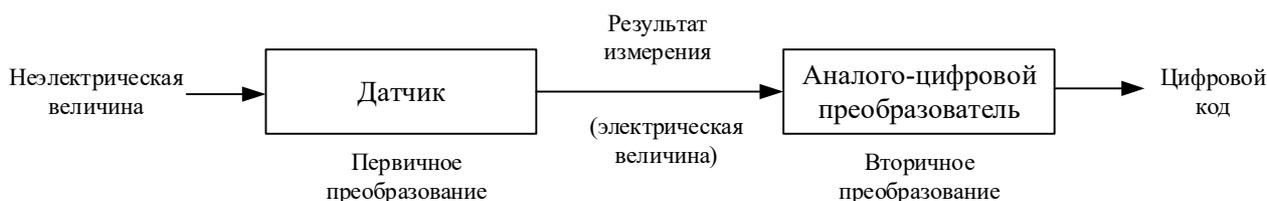


Рис. 1. Процесс измерения неэлектрической величины

Как правило, тарировочная характеристика, описывающая зависимость электрической величины от измеряемой неэлектрической, нелинейна (рисунок 2). Для точной обработки снятых измерений возникает необходимость решения задачи линеаризации, которая заключается в восстановлении искомой линейной зависимости. Для этого применяют функциональные преобразователи информации, использующие технологию искусственных нейронных сетей [1].

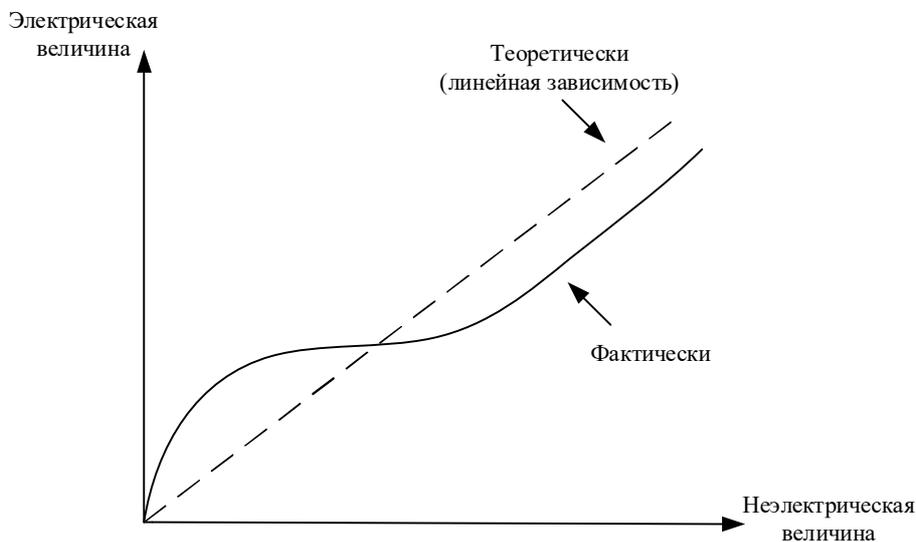


Рис. 2. Пример нелинейной зависимости

Представляются сложные функциональные зависимости обычно с помощью аппроксимации на основе технологии искусственных нейронных сетей. На рисунке 3 представлена искусственная нейронная сеть, используемая в качестве нейросетевой структуры функционального преобразователя [2, 3].

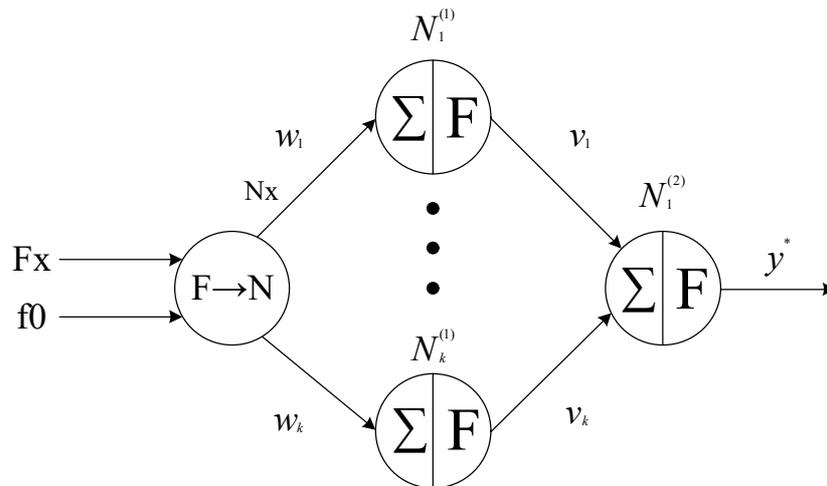


Рис. 3. Структура нейросетевого преобразователя

Для того, чтобы нейронная сеть была способна выполнить поставленную задачу, её необходимо обучить. Целью обучения сети является такая подстройка её весов, чтобы приложение некоторого множества входов приводило к требуемому множеству выходов. На рисунке 4 представлена блок-схема алгоритма обратного распространения ошибки [4]. Основная идея этого метода состоит в распространении сигналов ошибки от выходов сети к её входам, в направлении, обратном прямому распространению сигналов в обычном режиме работы.

В качестве примера нелинейной зависимости выбрана функция $y = \cos(\pi x/2)$ на интервале $x \in [0; 1]$. Исследование было проведено для нейронной сети с 10-ю скрытыми слоями, которая обучалась на основе алгоритма обратного распространения. Исходные данные: объём выборки – 257, количество эпох обучения – 10000000.

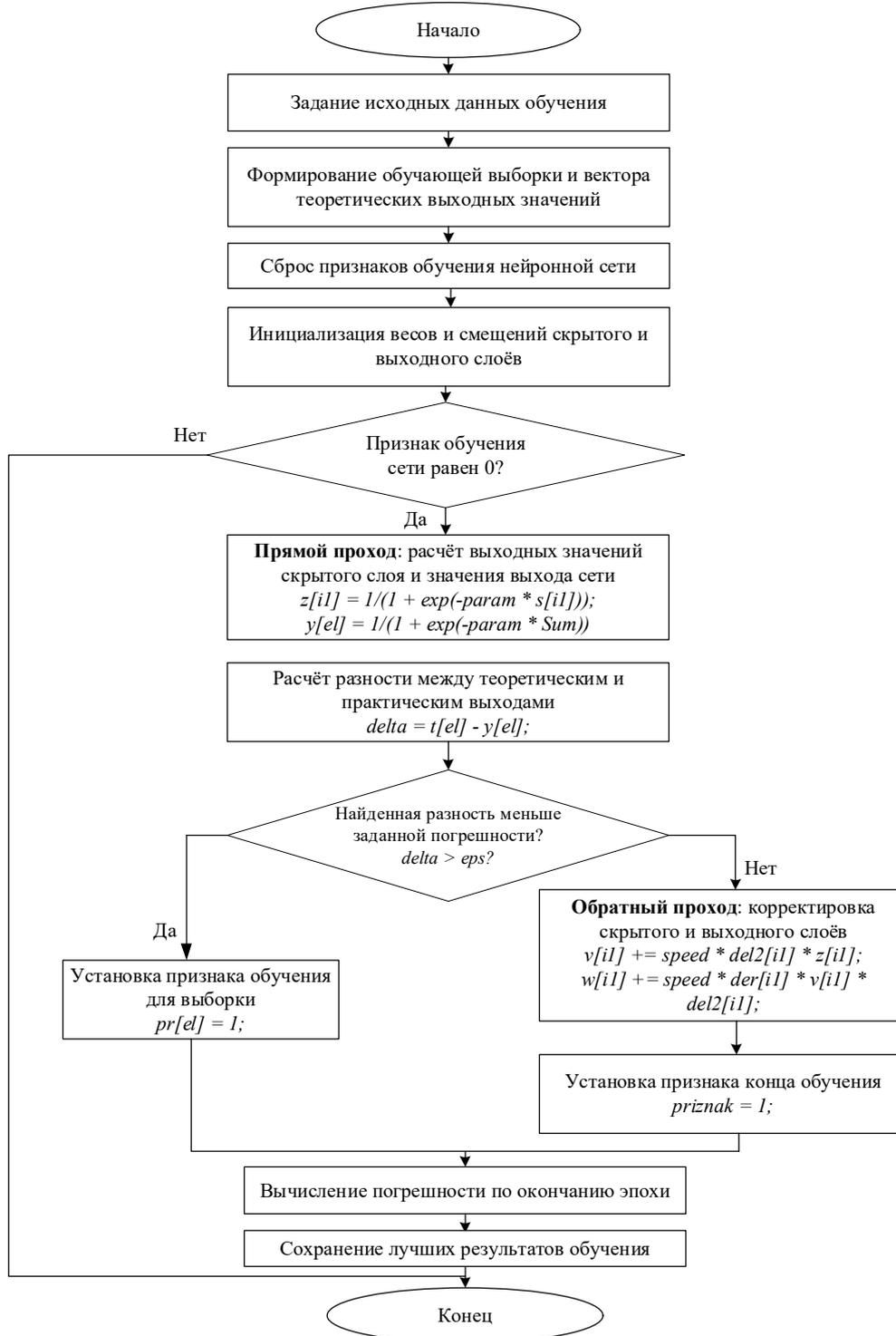


Рис. 4. Блок-схема алгоритма обратного распространения ошибки

В ходе выполнения анализа влияния параметра функции на значение погрешности преобразования был получен график, представленный на рисунке 5.

Из рисунка 5 можно сделать вывод, что при увеличении параметра активационной функции также уменьшается максимальная погрешность преобразования.

На рисунке 6 представлен график зависимости максимальной погрешности от объёма обучающей выборки.



Рис. 5. График зависимости максимальной погрешности преобразования от параметра активационной функции

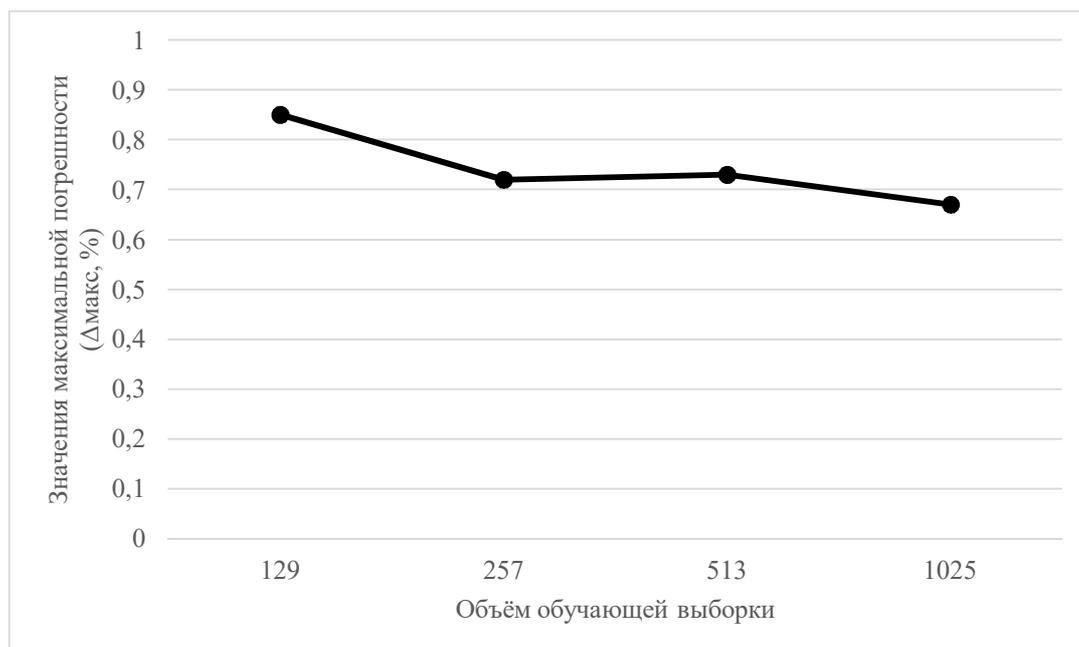


Рис. 6. График зависимости максимальной погрешности преобразования от объёма обучающей выборки

Согласно рисунку 6, увеличение объёма выборки ведёт к незначительным изменениям величины погрешности.

График влияния количества нейронов скрытого слоя на величину максимальной погрешности представлен на рисунке 7.

На основе данных рисунка 7 можно сделать вывод, что при выборе количества нейронов в скрытом слое больше 10, изменение величины погрешности преобразования незначительное.

В качестве микросхемы для реализации подсистемы функционального преобразования была выбрана микросхема семейства Cyclone II. Это серия недорогих микросхем. Их цена варьируется в диапазоне десятков долларов.

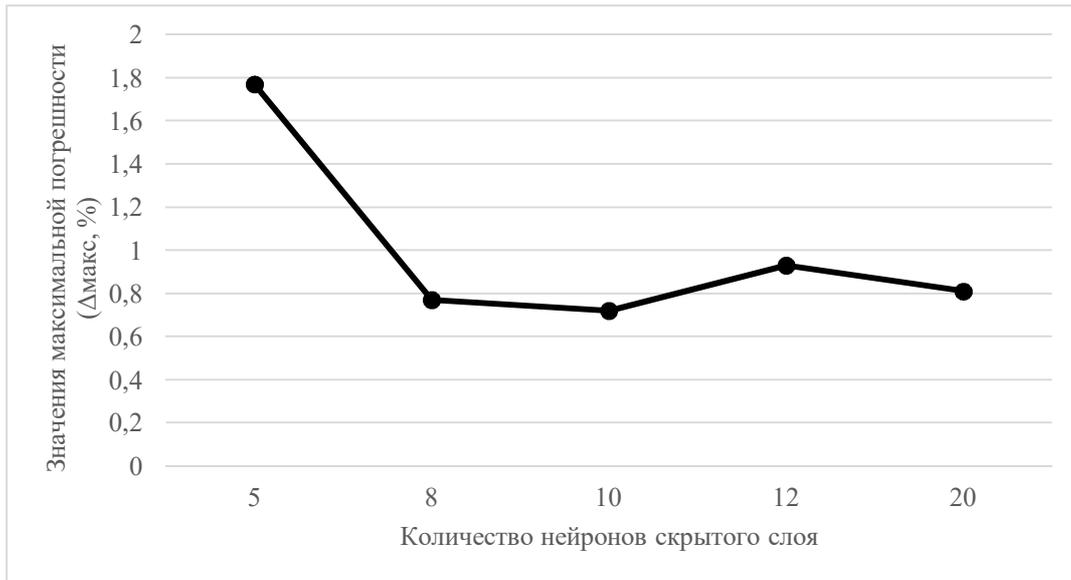


Рис. 7. График зависимости максимальной погрешности преобразования от количества нейронов скрытого слоя

Для моделирования работы нейронной сети могут быть использованы как последовательная, так и параллельная вычислительная схемы. Применение последовательной схемы обусловлено тем, что частота поступает на вход сети медленно, а вычисление значений выходов нейронов производится непосредственно в коде.

В таблицах 1 и 2 представлены аппаратные затраты на реализацию суммирующего устройства сети и устройства для вычисления выходов скрытого слоя сети соответственно.

Таблица 1. Аппаратные затраты при реализации суммирующего устройства 1-го слоя

	Параллельная схема	Последовательная схема
Количество комбинационных функций	178	84
Количество D-триггеров	185	246
Количество аппаратных умножителей	20	4

Таблица 2. Аппаратные затраты при реализации устройства вычисления выходов 1-го слоя сети

	Параллельная схема	Последовательная схема
Количество комбинационных функций	2,372	433
Количество D-триггеров	392	424
Количество аппаратных умножителей	32	4

Видно, что аппаратные затраты при использовании последовательной схемы намного меньше, чем при использовании параллельной. Также на основе анализа аппаратных затрат можно сделать вывод, что последовательная реализация нейронов цифрового слоя сети позволит реализовать многоканальную информационно-измерительную систему на основе программируемой микросхемы с низкой стоимостью.

Библиографический список

1. Гализдра В.И., Бабаев Ш.Б. Нейронные сети и аппроксимация данных // Научные и образовательные проблемы гражданской защиты. 2011. № 3. С. 35-43.
2. Солдатова О. П. Нейроинформатика: учебное пособие. Самара. 2013. 130 с.
3. Локтюхин В.Н., Челебаев С.В. Нейросетевые преобразователи импульсно-аналоговой информации: организация, синтез, реализация / Под общей редакцией А. И. Галушкина. — М.: Горячая линия–Телеком, 2008. — 144 с.
4. Уоссермен Ф. Нейрокомпьютерная техника: Теория и практика / Ф. Уоссермен; Перевод с англ. Ю. А. Зуева, В. А. Точенова; Под ред. А. И. Галушкина. М.: Мир, 1992. 236 с.

УДК 004.942; ГРНТИ 50.03.05

ИМИТАЦИОННАЯ МОДЕЛЬ СИСТЕМЫ АВТОРЕГУЛИРОВАНИЯ

В.С. Ноздрачева

*Рязанский государственный радиотехнический университет имени В.Ф. Уткина,
Российская Федерация, Рязань, nozdracheva1997@mail.ru*

Аннотация. В работе рассматривается имитационная модель системы авторегулирования.

Ключевые слова: имитационная модель, система авторегулирования, релейная астатическая система фазовой автоподстройки частоты (РАФАПЧ).

SIMULATION MODEL OF THE AUTO-REGULATION SYSTEM

V.S. Nozdracheva

*Ryazan State Radio Engineering University named after V. F. Utkin, Ryazan,
Russian Federation, nozdracheva1997@mail.ru*

Annotation. The paper considers a simulation model of the auto-regulation system.

Keywords: simulation model, autoregulation system, relay astatic system of phase-locked frequency (PLL).

Исследование систем авторегулирования, имеющих сложную структуру, либо поведение которых зависит от ряда случайных факторов, затруднительно выполнить аналитически. Часто для осуществления анализа таких систем прибегают к методам имитационного моделирования.

В работе рассматривается имитационная модель системы фазовой автоподстройки частоты, содержащей нелинейные элементы и элементы с дискретным управлением [1].

Алгоритм работы имитационной модели (рисунок 1) представлен рядом блоков, каждый из которых выполняет следующие функции:

- 1 – задание начальных параметров; определяются значения параметров модели;
- 2 – оператор цикла, определяющих время имитации;
- 3 – формулы, описывающие поведение фазы и частоты моделируемой системы;
- 4 – 10 – блок-схема, определяющая значения знака разности фаз в системе;
- 11 – 12 – блоки вывода результатов.

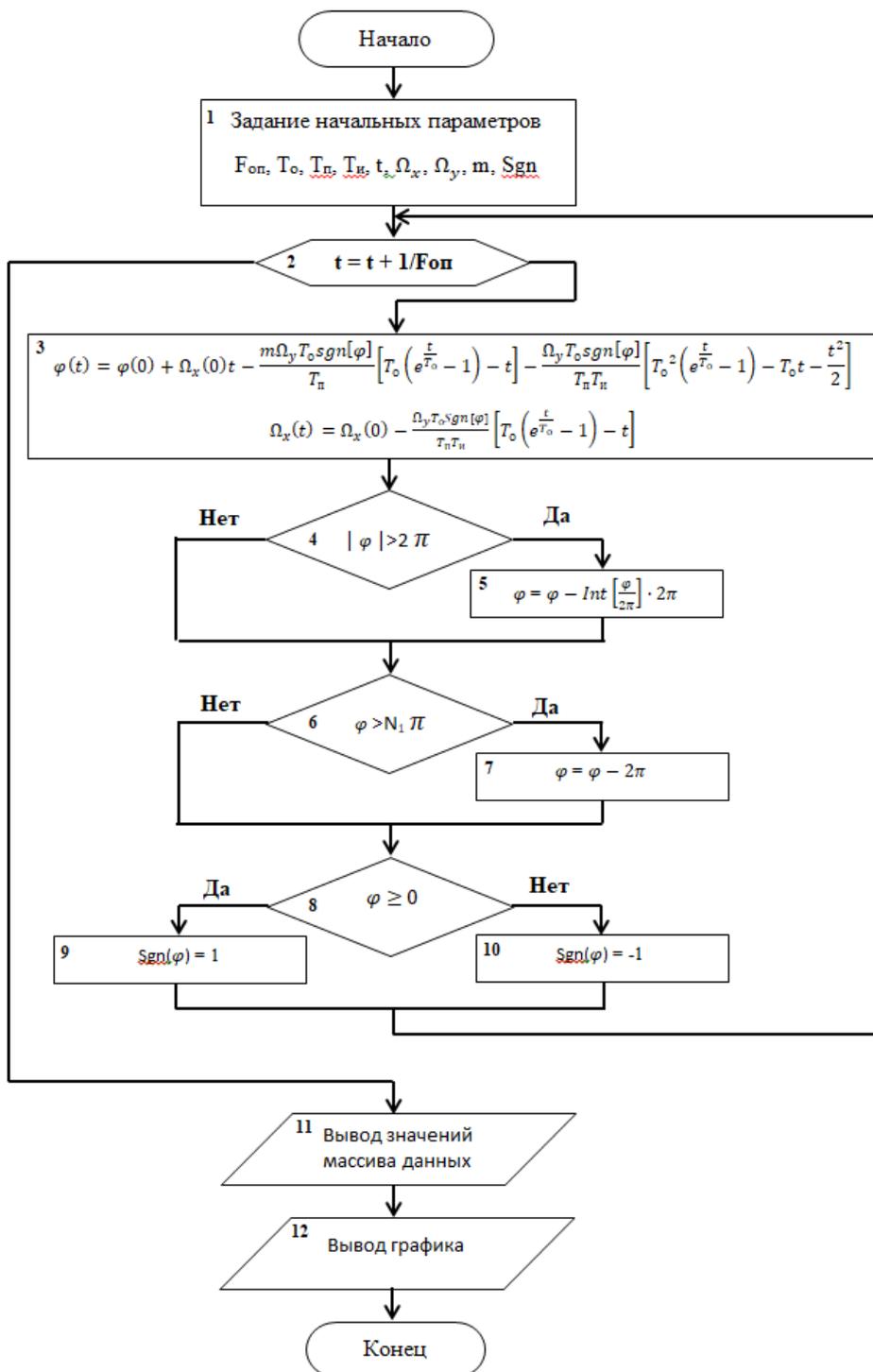


Рис. 4. Алгоритм функционирования имитационной модели

Разработанное программное средство, базирующееся на основе математической модели системы авторегулирования, позволяющее выполнять ряд операций, вызов которых реализуется через стартовое окно программного приложения. Выбор варианта решаемой задачи осуществляется вызовом определенного окна. Результат решения задачи выводится в виде массива числовых данных, а также в виде графической форме, отражающих результаты вычислений, осуществляемых в модели (рисунок 2).

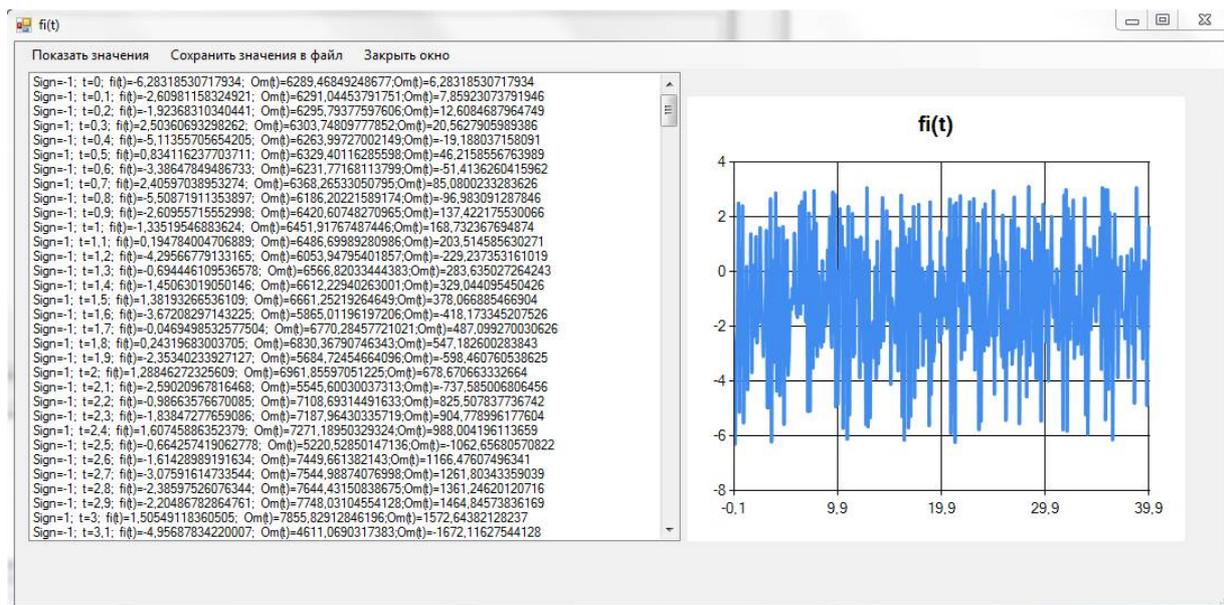


Рис. 2. Окно с результатами

Заключение

В работе рассмотрена имитационная модель системы авторегулирования, в основе которой лежит математическая модель, описывающая реальную систему РАФАПЧ при заданных параметрах.

Работа выполнена под научным руководством зав. кафедрой АСУ РГРТУ, к.т.н., доцента С.И. Холопова.

Библиографический список

1. А.С. 1415441 СССР, МКИ НОЗ L7/00. Устройство фазовой автоподстройки частоты / В.Ф. Одинокоев, С.И. Холопов, М.В. Петров (СССР). №4162161/24-09; Заявл. 15.12.86; Оpubл. 07.08.88 в Б.И. №29.
2. Бесекерский В.А., Попов Е.П. Теория системы автоматического управления / М.: Профессия, 2003. 752 с.
3. Воронов А.А. Теория автоматического управления. Часть 2. Теория нелинейных и специальных систем автоматического управления / М.: Высшая школа, 1986. 367 с.

СОДЕРЖАНИЕ

ИНФОРМАЦИЯ О III МЕЖДУНАРОДНОМ ФОРУМЕ «СОВРЕМЕННЫЕ ТЕХНОЛОГИИ В НАУКЕ И ОБРАЗОВАНИИ» СТНО-2020».....	3
МЕЖДУНАРОДНАЯ НАУЧНО-ТЕХНИЧЕСКАЯ КОНФЕРЕНЦИЯ «СОВРЕМЕННЫЕ ТЕХНОЛОГИИ В НАУКЕ И ОБРАЗОВАНИИ. ВЫЧИСЛИТЕЛЬНАЯ ТЕХНИКА И АВТОМАТИЗИРОВАННЫЕ СИСТЕМЫ».....	5
Секция «АЛГОРИТМИЧЕСКОЕ И ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ ВЫЧИСЛИТЕЛЬНЫХ СИСТЕМ И СЕТЕЙ».....	5
Пруцков А.В. Оптимизация зависимостей в программе с помощью правила Деметры.....	5
Каширин И.Ю. Нейронные сети, использующие модели знаний.....	9
Цуканова Н.И., Головкин Н.В., Шурыгина О.В. О решении практических задач с использованием глубоких нейронных сетей на базе облачных технологий.....	14
Ломаков А.В. Интеллектуальное программное обеспечение классификации данных с использованием алгоритма k-ближайших соседей.....	18
Соловьева А.С. Анализ методов многокритериальной оптимизации сети на основе графовых моделей.....	24
Васильев И.В. Обоснование актуальности разработки программного обеспечения классификации жанров.....	28
Мерзлякова М.Э. Интерфейс разработанной ИС для управления себестоимостью продукции на предприятии.....	31
Лацилин М.А. Определение тональности текста на основе рекуррентных искусственных нейронных сетей как один из методов, используемых при выявлении признаков мошенничества.....	37
Шитова К.Г. Проектирование интеллектуальной системы подбора кредитных продуктов.....	41
Иванов Л.А. Использование микросервисной архитектуры при реализации информационной системы для оптимизации работы патрулей.....	45
Николов П. Разработка программного обеспечения для исследования метрологических характеристик солемеров.....	48
Танцев А.С. Выявление мошеннических операций с кредитными картами с помощью нейросетевых технологий.....	53

Швечкова О.Г., Елисеев В.А. Разработка программного обеспечения для подбора комплектующих для компьютера на основе многокритериальной выборки.....	57
Соловов М.А. Разработка программного модуля на платформе 1С для расчета количества сотрудников, необходимого для исполнения рабочих процессов.....	60
Перехода И.Ю. Сравнительный анализ парсеров исходного кода в контексте возможностей обнаружения дефектов в нем.....	64
Горкин А.В. Метрики программ на языке JAVA и программные средства для их вычисления.....	67
Пылькин А.Н., Корнев П.А., Кокорев Е.И. Программная реализация алгоритма Левенберга-Марквардта для обучения искусственных нейронных сетей при аппроксимации полинома в среде Matlab.....	70
Фролов Д.А. Роль вычислительных систем в определении потребности заказа аптечной сети.....	77
Тараненко Е.А. Принципы построения автоматизированной информационной системы контроля и управления доступом на предприятии.....	81
Всемиров Н.В., Пылькин А.Н. Использование Docker для развертывания web приложения.....	84
Новинская Д.В. Применение графа для решения задач поиска наиболее популярных авторов книг в социальной сети.....	87
Секция «ЭВМ И СИСТЕМЫ».....	92
Костров Б.В., Гринченко Н.Н., Вьюгина А.А., Баранова С.Н. Исследование гистограмм распределения спектральных составляющих в спектре ландшафтных изображений.....	92
Бикташев Р.А., Жданов Д.М., Журкин Д.С. Модель планировщика задач в видефункциональных блоков IEC 61499.....	98
Бастрычкин А.С. Выбор размерности ортогонального базиса для передачи изображений.....	103
Саблина В.А., Савин А.В. Поиск антропометрических точек лица с помощью Openface и mediapipe.....	107
Титова М.А., Громов А.Ю. Генеративный дизайн с использованием интеллектуальных технологий.....	112
Благодаров Е.А., Тарасова В.Ю. Обзор и анализ применения полнотекстовых индексов В MSSQL.....	115
Кривоzubов В.О. Интеграция продуктов 1С с контрольно-кассовой техникой.....	120

Ефимов А.И. Алгоритмы обработки пространственно-координатных данных.....	125
Вьюгина А.А., Баранова С.Н. Реализация сложных связей с помощью современных СУБД.....	129
Ципорков Н.И. Анализ параллельных алгоритмов.....	132
Комлева Е.Р., Никифоров М.Б. Физически неклонлируемые функции микропроцессоров.....	135
Чан Туан Зунг Современные методы коррекции ошибок в памяти.....	140
Стеблюк А.Ю. особенности функционирования информационно-аналитических систем.....	145
Кабанова Е.С. Обзор и анализ методов оценки состояния оборудования.....	149
Баранова С.Н. Проектирование архитектур систем работы с большими данными.....	154
Карпова А.С. Решение строковых задач с помощью суффиксного автомата.....	159
Боровков И.В. Разработка алгоритма генерации ландшафтных изображений.....	162
Кузяков С.М. Циклические помехоустойчивые коды.....	168
Баранчиков А.И., Хизриева Н.И., Силкин Г.Д. Технические особенности автоматизации процессов обеспечения качества при непрерывном развертывании программного обеспечения.....	172
Секция «ИНФОРМАЦИОННЫЕ СИСТЕМЫ И ТЕХНОЛОГИИ».....	176
Селиванов Е.В. Свойства представления иерархических древовидных последовательностей.....	176
Немтинов В.А., Горелов А.А., Борисенко А.Б., Немтинова Ю.В., Трюфилкин С.В., Морозов В.В. Разработка виртуального тура по дому-музею семьи Асеевых в Тамбове.....	180
Кабулов А.А. Влияние современных технологий на качество образования.....	185
Архипов А.Е., Назарова А.О. Определение негативных факторов при взаимодействии с виртуальной реальностью в процессе обучения.....	190
Ищенко Д. А. Информационная поддержка принятия решений при проведении технологического обслуживания хлебопекарной печи.....	195
Немтинов К.В., Вязинкин А.Ю., Двухжилова И.В., Терехов С.М. Изучение исторических процессов с использованием технологии E-Learning.....	200

Соловьева И.А., Соловьев Д.С., Хлебников В.В. Разработка IDEF0-модели процесса проектирования архитектуры и выбора оборудования локальной вычислительной сети.....	203
Крылов А.В. Разработка пенного газопромывателя.....	207
Каев В.Ю., Дубинин В.Н. Моделирование раскрашенных сетей Петри с использованием языка SPARQL.....	212
Сальников А.П. Работа с фреймворком Slate для написания богатых текстовых редакторов.....	218
Огурцова Е.Ю., Фадеев Р.Н. Мобильная пожарная робототехника как направление внедрения информационных технологий в МЧС России.....	221
Шаронова Д.З., Миронов Г.С. Информационные системы и технологии в лесной промышленности.....	225
Аникеев Д.В. Особенности консолидирующей биллинговой системы расчетно-платежного комплекса «АБОНЕНТ+».....	230
Калашников В.А., Саватеев М.В., Мартышкин А.И., Трокоз Д.А. Предсказание сахарного диабета с помощью нейросети.....	233
Дмитриевский Б.С., Терехова А.А. Автоматизированная система управления сбытом готовой продукции.....	236
Костиков М.Г. Численный метод расчета данных потребления коммунального ресурса для проведения OLAP анализа.....	241
Куликова А.В., Маркин А.В. Автоматизированный расчет схемы обращения ТКО.....	245
Копцев Г.А., Аникеев С.В. Разработка технологии подготовки отчетной документации в системах с распределенным кэшированием данных.....	251
Андреев Д.А., Аникеев С.В. Разработка информационной модели ведения претензионно-исковой работы в сфере ЖКХ.....	257
Конькова А.В., Челебаев С.В. Разработка подсистемы функционального преобразования частоты в код на основе многослойной искусственной нейронной сети.....	261
Ноздрачева В.С. Имитационная модель системы авторегулирования.....	267

СОВРЕМЕННЫЕ ТЕХНОЛОГИИ В НАУКЕ И ОБРАЗОВАНИИ

Научное издание

В 10 томах

Том 4

Под общей редакцией О.В. Миловзорова.

Подписано в печать 15.06.21. Формат 60x84 1/8.

Бумага офсетная. Печать офсетная.

Гарнитура «Times New Roman».

Усл. печ. л.

Тираж 100 экз. Заказ №.

Рязанский государственный радиотехнический университет,
Редакционно-издательский центр РГРТУ,
390005, г. Рязань, ул. Гагарина, д. 59/1.
Отпечатано в типографии Book Jet,
390005, г. Рязань, ул. Пушкина, д. 18