

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-2022

**V МЕЖДУНАРОДНЫЙ НАУЧНО-
ТЕХНИЧЕСКИЙ ФОРУМ**

Сборник трудов

Том 4

Рязань
Book Jet
2022

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

Современные технологии в науке и образовании – СТНО-2022 [текст]: сб. тр. V международн. науч.-техн. форума: в 10 т. Т.4./ под общ. ред. О.В. Миловзорова. – Рязань: Рязан. гос. радиотехн. ун-т, 2022.

ISBN 978-5-7722-0301-9

Т.4: – 252 с.,: ил.

ISBN 978-5-7722-0301-9

Сборник включает труды участников V Международного научно-технического форума «Современные технологии в науке и образовании» СТНО-2022.

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

ISBN 978-5-7722-0301-9

© Рязанский государственный
радиотехнический университет, 2022

ИНФОРМАЦИЯ О V МЕЖДУНАРОДНОМ ФОРУМЕ «СОВРЕМЕННЫЕ ТЕХНОЛОГИИ В НАУКЕ И ОБРАЗОВАНИИ» СТНО-2022

V Международный научно-технический форум «Современные технологии в науке и образовании» СТНО-2022 состоялся 02.03.2022-04.03.2022 в г. Рязань в Рязанском государственном радиотехническом университете имени В.Ф. Уткина.

В рамках форума «Современные технологии в науке и образовании» СТНО-2022 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;
- Управление персоналом;
- Бухгалтерский учет, анализ и аудит;
- Экономическая безопасность;
- Внешнеэкономическая деятельность;

«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:

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

Богачева О.А., доцент кафедры "Государственное, муниципальное и корпоративное управление", к.полит.н., доц.;

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.;

Дмитриев В.Т., зав. кафедрой радиоуправления и связи, 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.;

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, д.т.н., доц.;

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

Ленков М.В., зав. кафедрой автоматизации информационных и технологических процессов, к.т.н., доц.

Литвинов В.Г., зав. кафедрой микро- и наноэлектроники, д.ф.-м.н., доц.;

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

Меркулов Юрий Аркадьевич, ст. преподаватель кафедры "Государственное, муниципальное и корпоративное управление";

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, зав. кафедрой автоматизированных систем управления, 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 004.67; ГРНТИ 50.41.25

**РАЗРАБОТКА ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ ДЛЯ ПРОКЛАДЫВАНИЯ
ПЕШИХ МАРШРУТОВ ПО ЗАДАНЫМ КРИТЕРИЯМ**

В.О. Зайцев

*Рязанский государственный радиотехнический университет имени В.Ф. Уткина,
Российская Федерация, Рязань, vlzaitsev2000@gmail.com*

Аннотация. В наше время при строительстве городов удобству пешеходов уделяется значительно меньше внимания, чем оптимизации транспортного движения. Негативно сказываются такие факторы, как выбросы выхлопных газов автомобилей, невысокое число зеленых насаждений, отсутствие маршрутов, охватывающих достопримечательности и значимые места. Для повышения информированности пешеходов о путях, лишенных перечисленных недостатков разрабатывается программное обеспечение для прокладывания пешеходных маршрутов по заданным критериям. Проектируемая система построена на трехзвенной клиент-серверной архитектуре, где в качестве клиентской части выступает одностраничное приложение, а взаимодействие клиента и сервера происходит на основе архитектурного стиля REST. В качестве средств разработки использовалось следующее программное обеспечение: источник информации – веб-картографический проект OpenStreetMap, основной язык программирования – Java в сочетании с фреймворком Spring Framework, хранилище данных – система управления базами данных PostgreSQL, построитель маршрутов – навигационный движок GraphHopper. Алгоритм работы системы заключается в загрузке информации в систему управления базами данных, ее обработке в таблицах базы данных, построении навигационного графа для проложения маршрутов.

Ключевые слова: навигация, маршрутизация, пешеходные маршруты, экология, OpenStreetMap, Java, GraphHopper.

**DEVELOPMENT OF SOFTWARE FOR LAYING HIKING ROUTES
ACCORDING TO SPECIFIED CRITERIA**

V. O. Zaitsev

*Utkin Ryazan State Radio Engineering University,
Russia, Ryazan, vlzaitsev2000@gmail.com*

Abstract. Nowadays, in the construction of cities, the convenience of pedestrians is given significantly less attention than the optimization of traffic. Such factors as car exhaust emissions, a low number of green spaces, the absence of routes covering attractions and significant places have a negative impact. To increase the awareness of pedestrians about the paths devoid of the listed disadvantages, we are developing software for laying hiking routes according to the specified criteria. Our system is built on a three-tier client-server architecture, where a single-page application acts as the client part, and the interaction of the client and the server is based on the REST architectural style. We have used the following software as development tools: the source of information is the OpenStreetMap web mapping project, the main programming language is Java in combination with the Spring Framework, the data warehouse is the PostgreSQL database management system, the route builder is the GraphHopper navigation engine. The algorithm of the system consists in loading information into a database management system, processing it in its tables, building a navigation graph for laying routes.

Keywords: navigation, routing, walking routes, ecology, OpenStreetMap, Java, GraphHopper.

Введение

Современные города зачастую спроектированы для оптимизации прежде всего транспортного движения, при этом удобству пешеходов уделяется намного меньше внимания. С одной стороны, выбросы выхлопных газов автомобилей, уменьшение числа зеленых насаждений, обилие дорожной пыли и другие экологические проблемы негативно сказываются на здоровье людей [1]. С другой стороны, в большинстве городов не хватает маршрутов, охватывающих достопримечательности и значимые места [2].

Для решения проблемы необходимо помочь пешеходам выстраивать свои маршруты с учетом вышеупомянутых аспектов. Целью разработки программного обеспечения является повышение уровня информированности населения о маршрутах, которые максимально снижают воздействие негативных факторов окружающей среды, а также являются более интересными для прогулок. Воплотить это можно путем реализации двух функций: (1) функции построения наиболее экологичных пешеходных маршрутов, что приведет к снижению влияния негативных факторов окружающей среды на организм человека, (2) функции создания интересных пешеходных маршрутов как для горожан, так и для туристов.

Архитектура

Разрабатываемая система основана на клиент-серверной архитектуре. Такая архитектура определяет взаимодействие программ (клиентской и серверной частей) в приложении. Сервер обслуживает запросы, инициируемые клиентами, по предоставлению интересующих их данных или выполнению различных расчетов. Система управления базами данных (СУБД) в таких системах может выделяться в отдельный уровень, в таком случае архитектура становится трехуровневой. В разрабатываемой системе используется трехуровневая клиент-серверная архитектура (рисунок 1). Программные компоненты обозначены прямоугольниками, СУБД обозначена цилиндром. Такой выбор обусловлен двумя причинами. Во-первых, такая архитектура позволяет сделать программное обеспечение крайне универсальным – для взаимодействия с системой пользователю необходим только браузер и доступ в сеть Интернет. Во-вторых, при таком подходе появляется возможность снизить нагрузку на клиентскую часть, перенеся все вычисления на сервер.

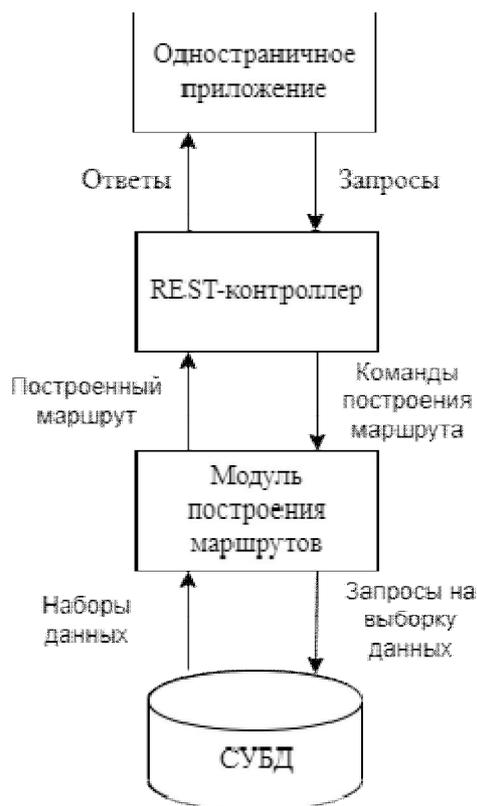


Рис. 1. Структура программного обеспечения

В качестве клиентской части выступает одностраничное приложение (*single page application*). Такой подход обладает следующими преимуществами. Во-первых, высокая скорость работы благодаря осуществлению загрузки данных посредством AJAX-запросов, что избавляет от необходимости перезагрузки или смены страницы. Во-вторых, упрощенная разработка, поскольку не требуется реализации динамической генерации страниц на стороне сервера.

Серверная часть представлена REST-контроллером и модулем построения маршрутов.

REST-контроллер обеспечивает взаимодействие клиентской части с модулем построения маршрутов. Связь реализована посредством протокола HTTP с использованием архитектурного стиля REST (*REpresentational State Transfer*). Для передачи параметров запросов (например, координат точки начала маршрута, координат пункта назначения, вида маршрута) клиентов и отправки ответов сервера (например, координат точек, из которых состоит построенный маршрут) выбран формат JSON.

Модуль построения маршрутов – компонент, выполняющий основную функцию системы – построение маршрутов по запросам пользователя. Модулю передаются параметры требуемого маршрута от REST-контроллера, после чего он выполняет построение пути с учетом заданных требований, обращаясь, в случае необходимости, за нужной информацией в СУБД.

СУБД отвечает за хранение и обработку необходимой для работы системы информации. К такой информации относится совокупность сведений о географической области, с которой происходит работа: данные о дорогах, интересующих географических объектах (парках, скверах, достопримечательностях и т.д.).

Средства разработки

Прежде всего для работы разрабатываемого программного обеспечения необходимо получать информацию о географических областях. Источником таких данных решено использовать некоммерческий веб-картографический проект OpenStreetMap [3], поскольку он является одним из наиболее полных источников. Информация, которая может быть получена из этого сервиса, распространяется на условиях свободной лицензии Open Database License [4], что делает этот источник данных более подходящим. Получение данных производится путем загрузки .pbf файла с веб-ресурса <https://www.geofabrik.de>.

Основным языком программирования был выбран Java в сочетании с фреймворком с открытым исходным кодом Spring Framework. Такое решение обусловлено рядом преимуществ Java: быстротой, встроенными функциями защиты, кроссплатформенностью [5, 6]. Эти преимущества обеспечат высокую скорость работы системы, ее надежность и простоту развертки. Фреймворк Spring Framework позволит облегчить и ускорить разработку, а также сделать проектируемую систему более гибкой и легко изменяемой [7].

Для автоматизации управления зависимостями (библиотеками) и автоматизации сборки программного обеспечения используется фреймворк Apache Maven.

В качестве хранилища данных решено использовать объектно-реляционную СУБД PostgreSQL. Для обеспечения возможности хранения географических объектов используется открытое расширение PostGIS, добавляющее поддержку географических объектов, а также функции для обработки подобных данных. Такое сочетание дает возможность модификации исходных данных любым образом, не предусмотренным веб-картографическим проектом OpenStreetMap.

Для решения задачи построения маршрута используется библиотека для маршрутизации с открытым исходным кодом GraphHopper, написанная на Java. Библиотека предоставляет широкий спектр возможностей: Directions API, Routing and Navigation, Route Optimization, Geocoding, Time-dependent Route Optimization. Из всего вышеперечисленного в контексте разрабатываемой системы интересен второй пункт. Поскольку описываемое программное обеспечение, как отмечалось выше, написано на Java, существует возможность легко встроить его в разрабатываемую систему. Более того, возможно частично переопределить логику работы библиотеки, добившись тем самым возможности построения маршрута по собственным правилам.

Для наполнения базы данных необходимой информацией из файла .pbf решено использовать открытое программное обеспечение Osm2pgsql. Оно обладает всей необходимой функциональностью, а также предоставляет возможность детальной настройки процесса загрузки данных (определение создаваемых таблиц, обработка всех объектов перед загрузкой и т.д.) в хранилище с помощью конфигурационного файла на скриптовом языке Lua.

Клиентская часть (одностраничное приложение) разрабатываемого программного обеспечения реализуется в виде HTML-страницы, содержащей карту необходимой географической зоны. За отображение карты отвечает Leaflet – написанная на JavaScript библиотека с открытым исходным кодом. Она позволяет встроить на веб-страницу интерактивную карту, а также добавлять на карту различные слои, что позволит выводить на экран пользователя различные маркеры, отметки, построенный маршрут.

Описание алгоритма работы системы

Алгоритм работы разрабатываемого программного обеспечения включает следующие этапы:

1. Загрузка данных из файла .pbf в СУБД.

2. Обработка информации в таблицах базы данных.
3. Построение навигационного графа библиотекой GraphHopper.
4. Построение маршрутов по запросам пользователей.

Рассмотрим каждый из этапов подробнее.

На первом этапе происходит чтение и обработка файла .pbf программой Osm2pgsql. Инструментами СУБД создаются таблицы дорог, мест, влияющих на экологичность маршрута и достопримечательностей. Таблица дорог содержит уникальный идентификатор, теги, геометрию и прочие поля, необходимые для построения навигационного графа библиотекой GraphHopper. Таблицы мест, влияющих на экологичность маршрута и достопримечательностей содержат следующие поля: уникальный идентификатор, тип объекта, геометрия, название, теги.

После создания таблиц и загрузки информации из файла можно приступить к обработке полученных данных. В ходе этого этапа происходит проверка факта пересечения между объектами дорог и вышеперечисленными местами с помощью функции ST_Intersects расширения Postgis. Также происходит определение расстояния между объектами дорог и объектами, влияющими на экологичность маршрута. В зависимости от этих проверок каждому объекту присваивается рейтинг экологичности. Аналогичным образом определяется рейтинг, зависящий от наличия достопримечательностей. Эти два значения впоследствии будут являться весами ребер навигационного графа для построения маршрутов.

В ходе третьего этапа Java-приложение с помощью библиотеки GraphHopper осуществляет чтение информации из таблицы дорог базы данных, производит построение навигационного графа. Полученный граф кэшируется в виде бинарных файлов на накопитель используемой ЭВМ, а потому для проложения маршрутов в дальнейшем не требуется его повторное построение. Это позволяет добиться высокой скорости работы серверной части разрабатываемого программного обеспечения.

Следует отметить, что вышеперечисленные этапы проводятся только в случае необходимости актуализации данных о географической области, с которой работает система. В противном случае этапы проводятся однократно при первом запуске приложения.

Четвертый этап представляет собой штатную работу программного обеспечения. Пользователь, используя веб-браузер, переходит на адрес одностраничного приложения описываемой системы. После выбора точки начала, точки конца и типа маршрута, он посылает запрос на сервер, где модуль построения маршрутов производит создание маршрута согласно заданным параметрам, после чего полученный маршрут отображается на клиентской части программного обеспечения.

На момент написания статьи реализовано одностраничное приложение, представляющее клиентскую часть описываемого программного обеспечения, механизм загрузки информации в базу данных и ее обработка, REST-контроллер серверной части, а также кэширование навигационного графа. В ближайшее время планируется реализация измененного варианта построения графа с учетом весов ребер, хранящихся в базе данных.

Заключение

Изложены следующие результаты:

1. Выявлены некоторые факторы, негативно влияющие на здоровье и моральное состояние населения, в частности, пешеходов: выхлопные газы автомобилей, обилие пыли вблизи автомобильных дорог, выбросы промышленных предприятий, располагающихся в черте города, малое количество зеленых насаждений (парки, скверы), а также отсутствие каких-либо интересных мест и достопримечательностей в привычных маршрутах.

2. Поставлена проблема неосведомленности населения о маршрутах, учитывающих и снижающих воздействие негативных факторов, описанных ранее.

3. Для того, чтобы справиться с проблемой решено разработать программное обеспечение для прокладки новых пешеходных маршрутов, лишенных описанных недостатков.

4. Для разрабатываемой системы выбрана трехуровневая клиент-серверная архитектура. Клиентская часть реализуется одностраничным браузерным приложением, а серверная часть – Java-приложением. Такой подход дает несколько преимуществ: (1) универсальность как клиента, так и сервера, (2) высокая скорость разработки, (3) высокая скорость работы, (4) надежность.

5. При разработке и обеспечении нормального функционирования системы используются следующие программные средства: в качестве первичного источника необходимой информации – бесплатная географическая карта мира OpenStreetMap, язык для реализации серверной части приложения – Java с использованием фреймворка Spring Framework для ускорения и упрощения разработки, постоянное хранилище информации – СУБД PostgreSQL с пространственным расширением PostGIS, добавляющим поддержку географических типов, библиотека для построения маршрутов – навигационный движок GraphHopper, алгоритм маршрутизации которого самостоятельно изменен для добавления возможности учета новых критериев выстраиваемого пути.

6. Алгоритм работы системы включает следующие шаги: преобразование информации о географической области из файла .pbf в табличные представления используемой СУБД, обработка полученных данных, заключающаяся в присвоении рейтингов экологичности и привлекательности каждой дороге, построение навигационного графа на основе информации, полученной на предыдущем шаге, прокладывание маршрутов по запросам пользователей.

Библиографический список

1. Туринец Е. В., Довганюк А. И. Экологические тропы г. Москвы: анализ современного состояния // Вестник ландшафтной архитектуры. – 2013. – №. 1. – С. 91-95.
2. Гогина Е. Г. Благоустройство общественных пешеходных пространств // Инженерный вестник Дона. – 2020. – №. 5. – С. 31-31.
3. Bennett J. OpenStreetMap. Packt Publishing, 2010.
4. Open Database License (ODbL) v1.0. URL: <https://opendatacommons.org/licenses/odbl/1-0/> (дата обращения: 18.01.2022).
5. Волгина И. А. Преимущества изучения объектно-ориентированного языка программирования Java в 2021 году // Современные условия интеграционных процессов в науке. – 2021. – С. 8.
6. Белов В. Н., Ульянов Д. М. Преимущества языка программирования Java в современном программировании // Математический вестник педвузов и университетов Волго-Вятского региона. – 2018. – №. 20. – С. 282-287.
7. Уоллс К. Spring в действии. – М.: ДМК Пресс, 2013. – 752 с.

УДК 007:681.512.2; ГРНТИ 28.23.37

ИНТЕЛЛЕКТУАЛЬНЫЙ АНАЛИЗ ДАННЫХ В РЕТРОСПЕКТИВНОЙ ДИАГНОСТИКЕ

И.Ю. Каширин

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

Аннотация. Анонсируется оригинальная концепция проектирования алгоритмов интеллектуального анализа данных с использованием классической модели представления знаний в онтологической форме. Для ретроспективного анализа данных в области медицинской диагностики применяется вычисление семантической близости концептов и признаков, использующее прикладную ICF-онтологию. Для вычисления семантической близости признаков используется алгебраическая система иерархических чисел. Проведенные эксперименты на основе инструментария Python v.3 показывают эффективность предложенного подхода.

Ключевые слова: интеллектуальный анализ данных, медицинская диагностика, ретроспективный анализ, ICF-онтология, иерархические числа, семантическая близость, кластеризация.

INTELLIGENT DATA ANALYSIS IN RETROSPECTIVE DIAGNOSIS

I.Yu. Kashirin

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

The summary. The original concept of designing data mining algorithms using the classical model of knowledge representation in ontological form is announced. For retrospective data analysis in the field of medical diagnostics, the calculation of the semantic similarity of concepts and features is used, using the applied ICF ontology. To calculate the semantic proximity of features, an algebraic system of hierarchical numbers is used. The conducted experiments based on the Python v.3 toolkit show the effectiveness of the proposed approach.

Keywords: data mining, medical diagnostics, retrospective analysis, ICF ontology, hierarchical numbers, semantic similarity, clustering.

Современное состояние методов интеллектуального анализа данных дает возможность использовать все более сложные и затратные по вычислительным ресурсам технологии, базирующиеся на интеллектуальных и статистических алгоритмах поиска закономерностей [1]. Одной из актуальных задач в этой сфере являются задачи диагностирования в частности медицинского, учитывающего не только текущие признаки наблюдаемого субъекта, но и аналитику динамики набора этих признаков. Эффективное решение таких проблем существует в области обучающихся статистических и интеллектуальных алгоритмов [2].

Сложность ретроспективного анализа признаков заключается в том, что появляются дополнительные задачи:

- определение глубины обучающей выборки,
- выбор способов оптимизации количества и качества признаков,
- структуризация априорных моделей знаний [3] для выявления диагностических родовидовых и причинно-следственных таксономий [4].

В качестве предметной области будем использовать медицинскую диагностику. Задача ставится следующим образом.

Пусть есть ряд измерительных приборов и автоматизированных способов, позволяющих фиксировать следующие классы признаков, относящихся к здоровью человека:

- климатические,
- физиологические,
- психосоматические.

Требуется:

- определить причину негативных отклонений здоровья человека,
- закономерности и взаимозависимости динамики физиологических признаков,
- при необходимости выбрать рекомендации по текущей терапии.

В качестве измерительных приборов обычно выступают наручные браслеты или интеллектуальные часы, способные производить холтеровское мониторирование (измерение пульса, давления, аритмии), снимать кардиограмму, определять факторы внешней среды (температура, влажность), жизненные фазы (сон, бодрствование). В программных приложениях часто могут использоваться более сложные и разнообразные приборы, такие как силовой эргометр, спирометр, калипер и др.

Перечисленные особенности характеризуют задачу ретроспективной диагностики как более сложную, нежели классические задачи, решаемые многослойными нейронными сетями или алгоритмами интеллектуального анализа данных [4]. Для решения такой задачи необходимо использовать базы знаний и интеллектуальные информационные хранилища.

Для этого предложим использовать ICF-онтологию[5] и теорию иерархических чисел[6]. Приведем фрагмент ICF-таксономии для предметной области медицинской диагностики «Влияние на организм» (рис.1). Особенностью такого отношения является смежное наследование [5]. Здесь оно выражается в том, что факторами влияния на организм человека могут быть пища, физическая и психологическая нагрузки, экология, которые могут быть полезными и вредными. И наоборот, полезное и вредное влияния могут проявляться как посредством пищи, так физической и психологической нагрузок, а также внешней (экологической).

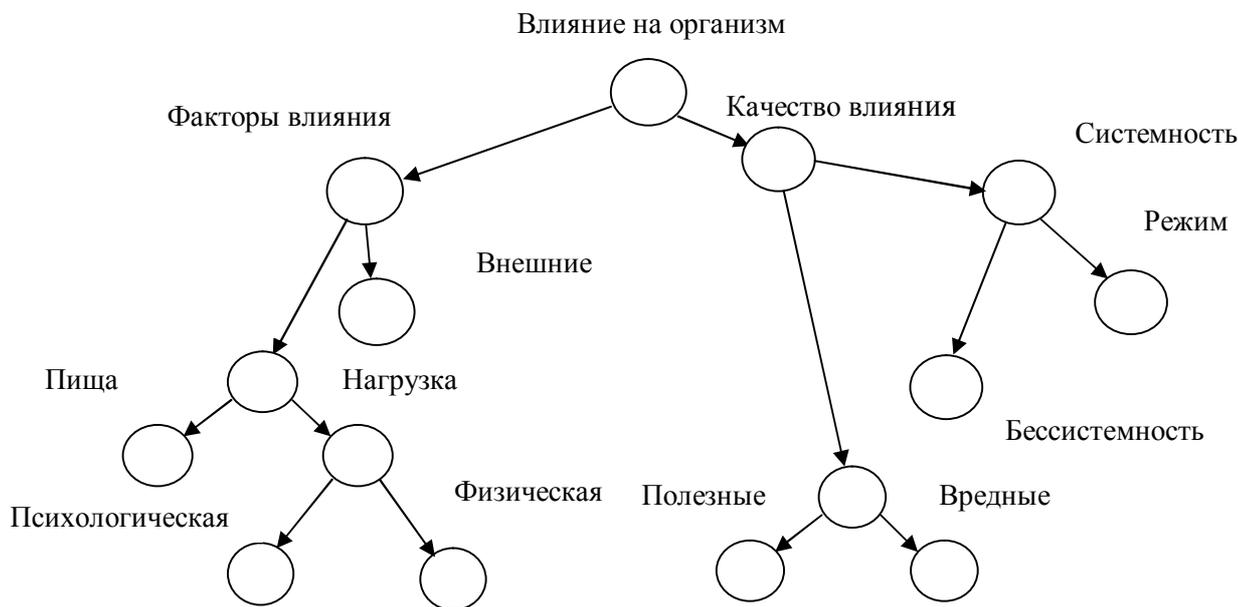


Рис. 1. ICF-отношение «Влияние на организм»

ICF-таксономия, заданная соответствующей иерархической схемой, отражает тот факт, что в любой вершине иерархии отражаются как составляющие все остальные вершины, расположенные как сверху по наследованию, так и снизу. Таким образом, можно «взяться за какую-либо вершину и вытянуть ее наверх иерархии», вся иерархия будет внутренней структурой концепта, соответствующего этой вершине. В то же время составляющие концепта могут быть более сильными и более слабыми. Например, вершины «Полезные» и «Вредные» явно ближе друг другу, чем «Полезные» и «Психологическая». Это свойство ICF-таксономии с оче-

видностью прослеживается на ее схеме. Математически точно степень смысловой близости концептов выражается при использовании теории иерархических чисел [6].

Кратко идея иерархических чисел выглядит так. Пусть N – множество целых положительных чисел с элементами $n_i \in N$, пусть также есть выделенный символ «.» . Множество $A = N \cup \langle \cdot \rangle \cup \wedge$ определяется как алфавит с целыми числами n , где $\langle \cdot \rangle$ – операция объединения множеств, а \wedge – пустой символ. Тогда грамматика:

$$\begin{aligned} \hat{h} &\rightarrow \wedge, \hat{h} \rightarrow h, \\ h &\rightarrow \langle n \rangle, h \rightarrow \langle n \rangle . \langle h \rangle \end{aligned}$$

описывает множество иерархических чисел \hat{H} с элементами \hat{h} .

Пример иерархического числа: 0.1.24.8 .

Алгебраическая система иерархических чисел определяется системой множеств:

$$\hat{H} = \langle \hat{H}, \{ -, \wedge \}, \{ >, <, = \} \rangle,$$

в которой $\{ -, \wedge \}$ – множество операций, а $\{ >, <, = \}$ – множество отношений на иерархических числах. Семантика операций дана в [6]. Операция $\langle \hat{h}_i - \hat{h}_j \rangle$ вычисляет минимальное количество вершин в некотором дереве, находящихся на пути между \hat{h}_i , \hat{h}_j , если такой путь существует. Эта разность задает расстояние между иерархическими числами. Если деревом является дерево онтологической модели знаний, то такое расстояние является семантической близостью двух концептов \hat{h}_i и \hat{h}_j , соответствующих двум разным вершинам этого дерева.

Отношения $\{ >, <, = \}$ определяют, находится ли одна вершина выше или ниже другой, или же является одной и той же вершиной.

В задаче диагностики нужно разделить признаки на полезные и вредные, удвоив таким образом их количество, а также на регулярные и бессистемные, что еще раз приведет к удвоению признаков. Такие удвоения безусловно оправданы, но между ними будет потеряна смысловая зависимость. Эта структура не дает возможности вычислить меры смысловой близости между концептами, и, следовательно, существенно затрудняет задачу поиска закономерностей и взаимозависимостей динамики признаков, сформулированную в начале настоящей статьи.

Для большей строгости изложения заметим, что объективно отражающая реальность структура признаков многократно сложнее структуры, представленной на предложенных рисунках. Например, логично было бы дополнить ее признаками «Интенсивность нагрузки», «Аэробика (как свежий воздух)» и «Апартаменты (как присутствие в доме, квартире)». Из сказанного несложно понять, в какую часть ICF-иерархии можно вставить каждый из этих признаков, и какие дополнительные вершины для этого понадобятся.

В целях прозрачности изложения материала из приведенной структуры для области здравоохранения выпущены такие важные вершины, как «Ретроспективные признаки» и «Текущая ситуация». Однако понятно, что два этих фактора весьма существенны в анализе данных. Текущие данные и их предыстория являются важными для каждого из концептов, представленных терминальными вершинами рисунков 1 и 2. Мало того, очень вероятно необходимость дальнейшего уточнения этих концептов такими денотатами, как: «Пуль», «Давление», «Интенсивность нагрузки», «Сон», «Бодрствование», «Белки», «Жиры», «Углеводы», «Жидкость», «Овощи», «Фрукты», «Стресс», «Положительные эмоции» и т.п. . Каждый из этих признаков может использоваться в качестве исходных данных для нейронных сетей или обучающихся интеллектуальных алгоритмов анализа. Эти данные могут быть включены в прикладную онтологическую модель данных [7], претендующих на ICF-основу. Однако, для каждого из этих данных текущее значение характеристик и их динамика очень важны.

Обладание каким-либо свойством для концептов каждой из вершин таксономии является главным признаком существования этого свойства в ICF-онтологии для выбранной предметной области. В этом случае онтологию нужно расширить в сторону большей абстракции, т.е. графически добавить более высокие вершины (рис.2).

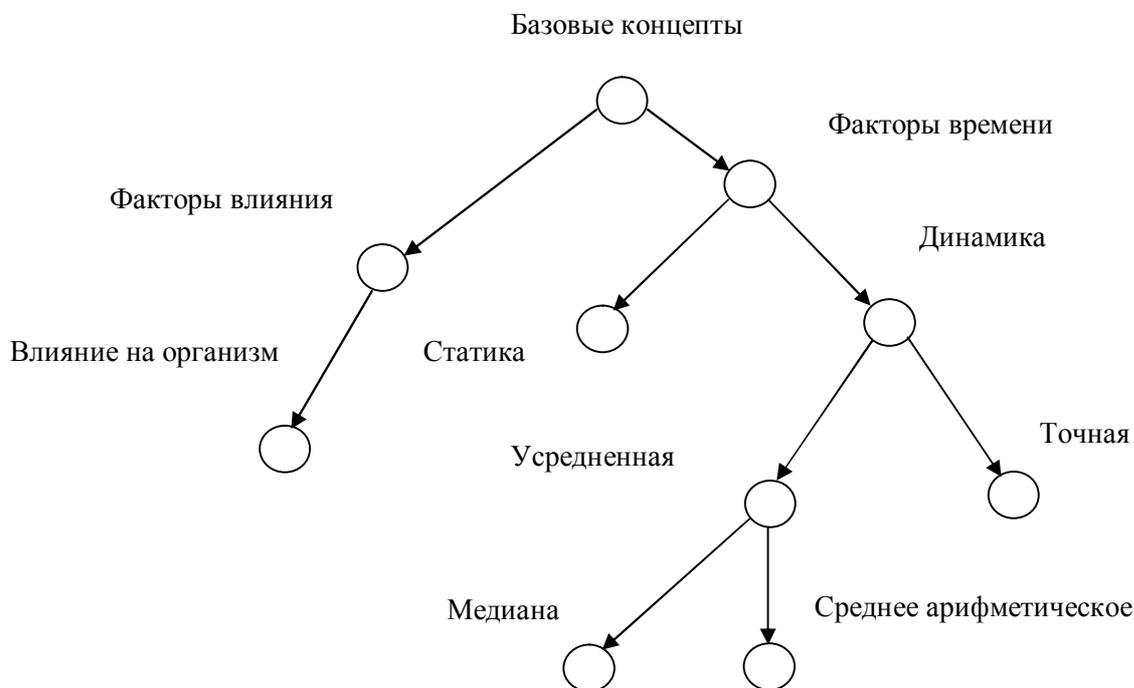


Рис. 2. Расширение таксономии факторами динамики и статики

В представленной ICF-таксономии допущено ряд упрощений, например, «Статика» и «Динамика» в общих онтологиях располагаются значительно выше по уровню абстрагирования, поскольку близки к базовым категориям «Пространства» и «Времени». Пример такой базовой онтологии можно найти в [5].

Иерархические числа, рассмотренные ранее, позволяют вычислить структурную семантическую близость концептов (характеристик). Семантическая близость вершин таксономии является величиной, которая при ее минимализме соответствует концептам, механизмы определения которых, по сути дела, одинаковы. Например, если концепт «Пища» разделить далее на «Белки», «Жиры» и «Углеводы», то механизм воздействия этих факторов на организм будет более схожим, чем, например, «Утренняя зарядка». Однако, если сравнивать «Пищу», «Физическую нагрузку» и «Внешнюю среду», то их семантическая близость не столь очевидна и может быть уточнена только ретроспективным анализом больших данных. Эти соображения определяют взаимозависимость алгоритмов интеллектуально анализа данных с моделями знаний соответствующей предметной области [8]. Точное знание значений семантической близости дают возможность упростить и сделать более эффективными алгоритмы анализа данных с обучением, т.к. позволяют априори определить признаки, подлежащие ретроспективному анализу и признаки, которые можно рассматривать в их текущих значениях.

Обратимся к задаче медицинской диагностики. Приведем фрагмент примерной таблицы признаков набора данных для обучения алгоритмов и анализа данных (табл.1).

Таблица 1. Выборка признаков исходного набора данных

№ п/п	Обозначение признака	Наименование признака
1	t, C	Температура воздуха
2	Humidity	Влажность в помещении
3	Pressure	Атмосферное давление
4	Pulse	Пульс утром
5	SYS pressure	Давление систолическое
6	DIA pressure	Давление диастолическое
7	Weight	Вес
8	Dream Total	Продолжительность сна
9	DreamContinuous	Качество сна (количество пробуждений)
10	Food Amount	Количество приемов пищи
11	Food healthy	Оценка разнообразия полезных продуктов
12	Food yesterday	Оценка качества питания накануне
13	Regime	Поддержание выбранного режима жизни
14	Walk Total	Длительность прогулок на свежем воздухе
15	Walk Amount	Количество прогулок на свежем воздухе

Для поставленной в начале настоящей статьи задачи необходимо в инструментарии Python исследовать все возможные характеристики на взаимовлияние, используя такой фрагмент программы:

```
import pandas.rpy.common as corr_health
import seaborn as seaborn_helth
%matplotlib inline
df = corr_health.load_data('Helth_Data')
corr = df.corr()
seaborn_helth.heatmap(corr,
                      xticklabels=corr.columns,
                      yticklabels=corr.columns)
```

Однако прямая корреляция признаков не дает возможности решить задачу поиска причины нежелательных отклонений показателей здоровья, поскольку наиболее значимые параметрами является предыстория динамики признаков. Для такого анализа необходимо задействовать аппарат моделей знаний на основе таксономий (рис.1).

Определение множества признаков для ретроспективного анализа вычисляется по формуле допустимой семантической близости следующим образом. Пусть b - таксономическая вершина концепта «Польза». $I(b)$ – иерархическое число, соответствующее b . U – универсум всех оцениваемых признаков, включенных в таксономию прикладной задачи о здоровье, а F – вычисляемое множество признаков, чье исследование нуждается в ретроспективном анализе. Обозначим через V порог допустимой семантической близости для выбора анализируемых признаков. Тогда критерием отбора таких признаков является следующая формула:

$$\forall x, x \in U, |I(x) - I(b)| < B \rightarrow x \in F,$$

где $|I(x) - I(b)|$ – значение семантической близости для концептов (признаков) x и b ;
 $\langle \rightarrow \rangle$ – операция логического следования (импликации);
 $\langle < \rangle$ – арифметическое отношение «меньше».

Например, испытательный стенд (ЭСМИАД v.10.02.2022), реализованный на языке Python v.3.7[4] в среде Anaconda3, дает возможность определить семантическую близость «Факторы влияния» и «Польза» как наиболее перспективную. В результате экспериментов становится понятным, что признаки 4-10 из таблицы 2 должны быть подвергнуты ретроспективному анализу. Практические эксперименты с небольшой группой испытуемых (14 человек) выявили эффективность предложенной концепции интеллектуального ретроспективного анализа и показали возможность кластеризации множества пациентов по индивидуальным особенностям их организмов. Так, например, если в одной группе оказались испытуемые, у которых при незначительном повышении веса (1-2%) уменьшалось кровяное давление, в то время как у второй группы испытуемых давление падало при таком же уменьшении веса. Кроме того, у первой группы ухудшалось самочувствие при регулярных прогулках перед сном, тогда как у второй – наоборот, основные показатели здоровья приходили в норму.

Библиографический список

1. Каширин И.Ю., Маликова Л.В., Маркова В.В. Основы формальных систем. / Под ред. И.Ю.Каширина – М.: Минобразования России, НИЦПрИС, 1999. 80с.
2. Каширин И.Ю. Нейронные сети, использующие модели знаний. Вестник Рязанского государственного радиотехнического университета. 2021. – № 75. – С.71-84.
3. 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.
4. Каширин И.Ю. Нейронные сети, использующие модели знаний. IV Международный научно-технический форум СТНО-2021. Сборник трудов. Том 4, Рязань, Book Jet, 2021, С.9-13.
5. Каширин Д.И. Формализм ICF-онтологий для представления знаний в глобальной сети нового поколения Semantic Вестник Тамбовского государственного технического университета. 2007. – Том 13. № 1. С. 91-103.
6. Каширин И. Ю. Иерархические числа для проектирования ICF-таксономий искусственного интеллекта. Вестник Рязанского государственного радиотехнического университета. 2020. – № 71. – С. 71-82.
7. Kashirin I.Yu., Filatov I.Yu. Formalized Description Of Intuitive Perception Of Spatial Situations. 2019 8th Mediterranean Conference on Embedded Computing (MECO), Budva, Montenegro, 2019, pp. 1-4.
8. Каширин И.Ю., Каширина О.И. Обзор концепций формального исследования инструментальных программных средств. Вестник Рязанского государственного радиотехнического университета. 2015. – № 2. – С. 74-83.

УДК 007:681.512.2; ГРНТИ 28.23.37

ИЕРАРХИЧЕСКИЕ ЧИСЛА В ПРОЕКТИРОВАНИИ ОБЩИХ ОНТОЛОГИЙ

И.Ю. Каширин

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

Аннотация. Рассматривается концепция проектирования онтологических моделей знаний, структурированных ICF-отношением. Отношение дает возможность использования полиморфизма, обеспечивающего одновременное рассмотрение концептов под разными углами зрения для решения множества существенно-разнообразных задач. ICF-структуризация достигается применением оригинальной технологии проектирования таксономий. Основой критериев структуризации при этом является вычисление семантической близости признаков с помощью теории иерархических чисел. Проведенные эксперименты на основе инструментария Python v.3 показывают эффективность предложенного подхода.

Ключевые слова: иерархические числа, модели знаний, семантическая близость, общая онтология, родовидовая таксономия, правила преобразования.

HIERARCHICAL NUMBERS IN THE DESIGN OF GENERAL ONTOLOGIES

I.Yu. Kashirin

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

The summary. The concept of designing ontological knowledge models structured by the ICF relation is considered. The relation makes it possible to use polymorphism, which provides simultaneous consideration of concepts from different angles of view for solving a set of essentially diverse problems. ICF-structuring is achieved by using the original taxonomy design technology. The basis of the structuring criteria in this case is the calculation of the semantic similarity of features using the theory of hierarchical numbers. The conducted experiments based on the Python v.3 toolkit show the effectiveness of the proposed approach.

Keywords: hierarchical numbers, knowledge models, semantic similarity, general ontology, generic taxonomy, transformation rules.

Общая онтология [1] используется для описания абстрактных знаний о понятиях, свойствах, отношениях и процессах, которые существуют в любой предметной области.

ICF-онтологией [1] является любая иерархическая модель знаний, построенная с использованием исключительно ICF-отношения.

Перечислим и опишем принципы проектирования общей ICF-онтологии.

1. Дихотомическая классификация концептов.
2. Строгая родовидовая таксономия (Is-A).
3. Исключение множественного наследования.
4. Строгая видо-видовая таксономия (Form).
5. Противоположность дочерних вершин в дихотомической триаде.
6. Уникальность концептов (редукция в смежное наследование).
7. Минимальность пути смежного наследования как семантической близости концептов.

Правила преобразования онтологий будем представлять следующей схемой.

{<Контрафактное утверждение>}

P_n: _____ [<Условие применимости>],

{<Регулярное утверждение>}

где **P_n** - индексное обозначение правила;

<Контрафактное утверждение> - утверждение, не соответствующее реальным фактам или плохо отражающее их;

<Регулярное утверждение> - правильное (исправленное) утверждение;

<Условие применимости> - необходимое условие возможности применения правила.

Дихотомическая классификация концептов

Дихотомическая классификация предполагает использование общей онтологии в форме бинарного дерева концептов. Для этого при существовании у одного концепта-предка родовидовой иерархии более чем двух потомков, необходимо найти промежуточные концепты, которые бы сделали дерево онтологии бинарным:

P1: {Is-A (X, x) & Is-A (X, y) & Is-A (X, z)}

————— [∃t Is-A (X, t) & Is-A (t, x) Is-A (t, y)],

{Is-A (X, t) & Is-A (X, z) & Is-A (t, x) Is-A (t, y)}.

Здесь и далее приняты обозначения: Is-A (X, x) – родовидовое отношение, для которого значение семантической близости в иерархических числах [2] равно 1, т.е. I(X,x)=1, & – операция логического умножения, ∃ – квантор существования.

Пример Пусть необходимо вписать в классификацию три концепта: “начальная ситуация”, “элемент процесса”, “конечная ситуация”.

Для каждого из этих понятий можно найти одно понятие, соответствующее отцовской вершине родовидовой иерархии – “смена состояний процесса”, что графически представлено на рисунке 1а.

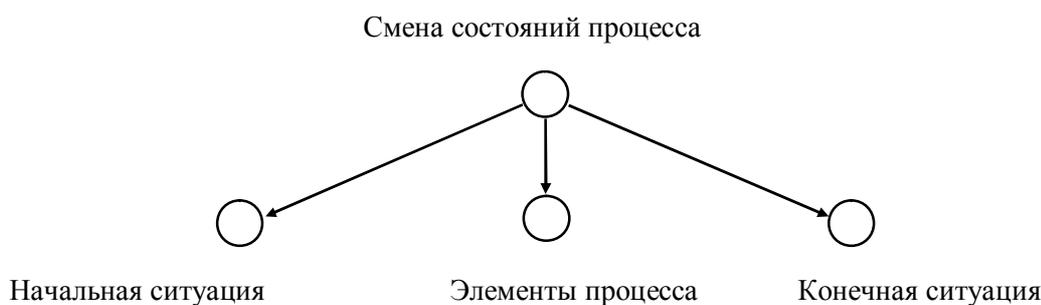


Рис. 1а. Фрагмент небинарного дерева наследования

Чтобы получить бинарное дерево, необходимо ввести промежуточный концепт, например, “границы процесса”, или проще, “границы” (рис.1б).

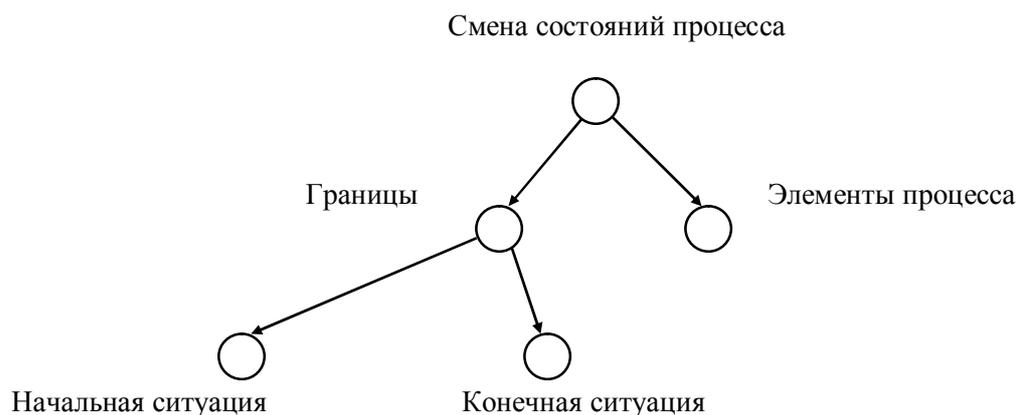


Рис. 1б. Фрагмент дихотомического дерева

Строгая родовидовая таксономия (Is-A)

Таксономия предполагает строгую последовательность родовидового наследования. Случаи инверсии наследования распознаются на основе анализа логики направления:

$$\begin{array}{l}
 \mathbf{P2:} \{Is-A(t, X) \& Is-A(X, y) \& Is-A(X, z)\} \\
 \hline
 [Is-A(X, t) \& Is-A(t, y) \& Is-A(t, z)], \\
 \{Is-A(X, t) \& Is-A(t, y) \& Is-A(t, z)\} .
 \end{array}$$

Пример Понятие “Элемент процесса” включает элементы какого-либо процесса, а следовательно, и смену состояний в качестве части процесса, поэтому в предыдущем примере обнаруживается инверсное наследование.

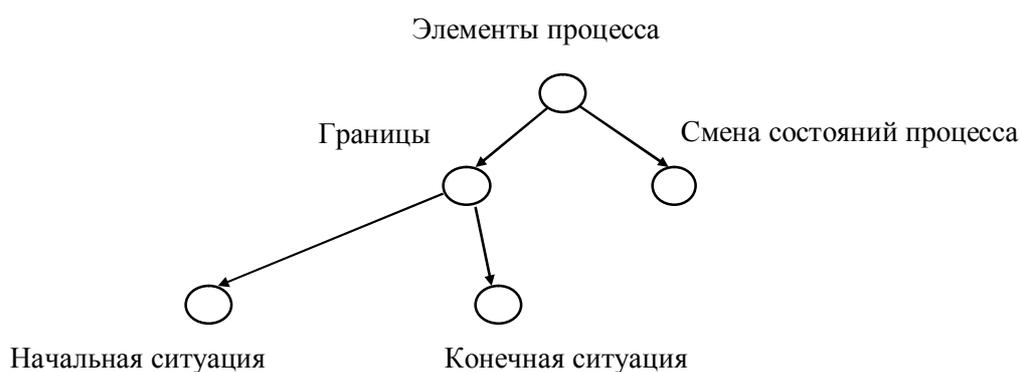


Рис. 2. Фрагмент онтологии после исправления инверсного наследования

После исправления инверсии фрагмент классификации будет выглядеть так (рис.2).

Исключение множественного наследования

В этом случае необходимо пересмотреть всю иерархическую классификацию так, чтобы найти в ней место новому концепту с единственной отцовской вершиной:

$$\mathbf{P3:} \{Is-A(X, x) \& Is-A(x, y) \& Is-A(x, z) \& Is-A(y, t) \& Is-A(z, t)\}$$

$$\frac{}{[\exists h \text{ Is-A}(X, h) \ \& \ \text{Is-A}(h, x)], \\ \{\text{Is-A}(X, t) \ \& \ \text{Is-A}(X, h) \ \& \ \text{Is-A}(h, x) \ \& \ \text{Is-A}(x, y) \ \& \ \text{Is-A}(x, z)\}}.$$

Пример Пусть в предыдущий фрагмент онтологии требуется включить новое понятие “мгновенное событие”. Его можно рассматривать как некоторый случай процесса, в котором начало и конец процесса совпадают (рис.3а). При сохранении предыдущих требований здесь появляется множественное наследование. Ликвидируем его (рис.3б). “Мгновенное событие” совпадает с “элементом события”, т.к. является тривиальным случаем.

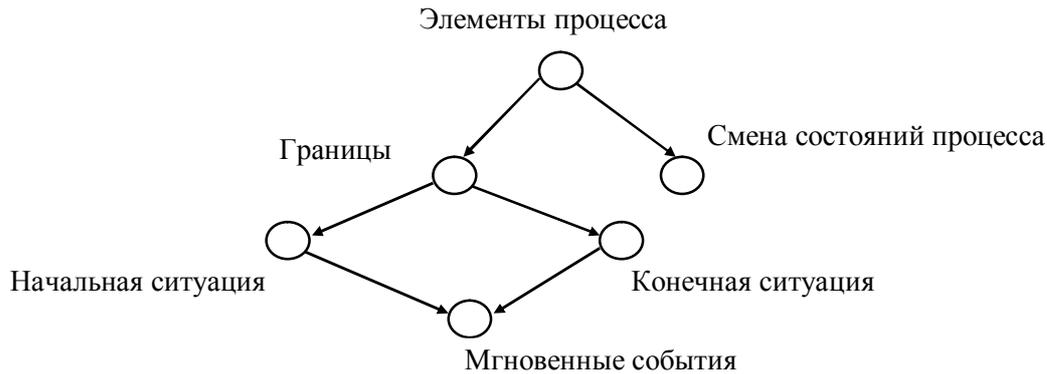


Рис. 3а. Фрагмент онтологии с множественным наследованием

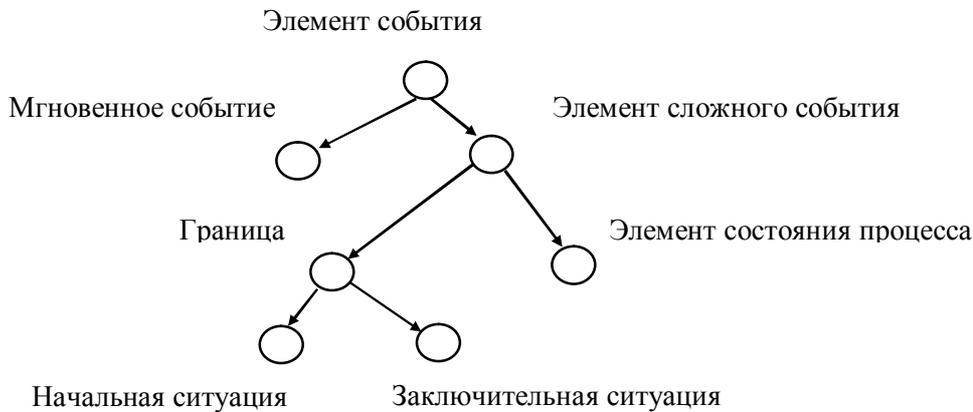


Рис. 3б. Фрагмент онтологии без множественного наследования

Строгая видо-видовая таксономия (Form)

Существование отношения $\text{Form}(X, y)$ говорит о том, что концепт X может проявляться в форме концепта y , т.е. быть частью свойств, присущих X . Это отношение совместно с Is-A позволяет утверждать о возможности смежного наследования, т.е. взаимного наследования свойств двумя дочерними вершинами через общую отцовскую. То же самое касается концептов более верхних уровней онтологии:

$$\mathbf{P4:} \quad \{\text{Is-A}(X, y) \ \& \ \text{Is-A}(X, z)\} \\ \frac{}{[\text{Form}(X, y) \ \& \ \text{Form}(X, z)], \\ \{\text{Is-A}(X, y) \ \& \ \text{Is-A}(X, z) \ \& \ \text{Form}(X, y) \ \& \ \text{Form}(X, z)\}}.$$

Правильное понимание отношения Form позволяет не только структурировать онтологическое описание, но и давать правильные наименования концептам онтологий.

Пример Для предыдущего рисунка учет видо-видовых отношений даст только коррекцию имен концептов. “Элемент события” может проявляться в форме “Мгновенного события”, “Элемента-границы”, “Элемента состояния процесса” и т. д., поэтому множественное число в наименовании этих концептов в рассматриваемом случае неуместно.

Скорректированный фрагмент онтологии представлен на рисунке 4.

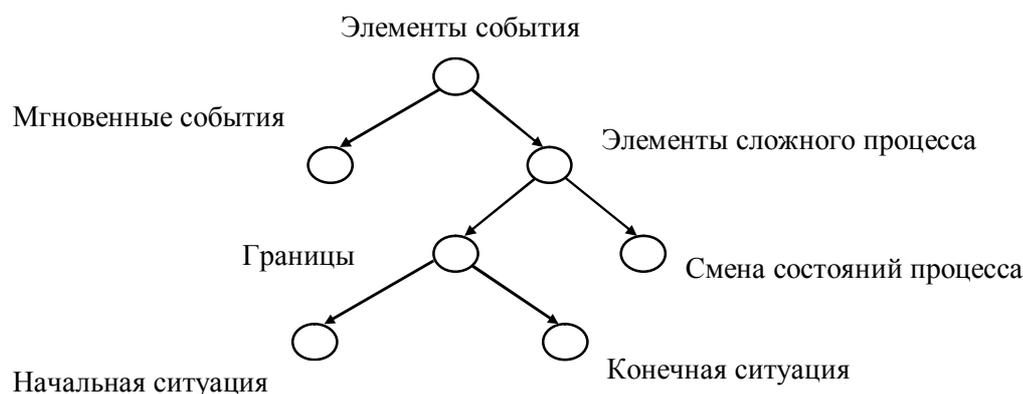


Рис. 4. Фрагмент онтологии с учетом видо-видовой таксономии

Противоположность дочерних вершин в дихотомической триаде

Противоположность задается отношением Cont. Это означает, что два понятия непосредственно подчиненные по родовидовой и видо-видовой иерархии одному отцовскому понятию должны иметь явную смысловую противоположность:

$$\begin{array}{l}
 \mathbf{P5:} \{ \text{Is-A} (X, y) \ \& \ \text{Is-A} (X, z) \} \\
 \hline
 \text{[Cont} (y, z)\text{]}, \\
 \{ \text{Is-A} (X, y) \ \& \ \text{Is-A} (X, z) \ \& \ \text{Cont} (y, z)\}.
 \end{array}$$

В качестве примера можно привести фрагменты на рисунках 1а и 1б. В этом случае дихотомическое разбиение было выполнено правильно, с учетом противоположности граничных событий “Начальная ситуация” и “Конечная ситуация”. Противоположность сохраняется, но ослабляется на более высоких уровнях иерархии. Например, на рисунке 4 “Начальная ситуация” как граничное событие, некоторым образом действительно противоположна “Элементу состояния процесса”, но более точная противоположность “Заключительной ситуации” говорит о том, что это понятие занимает правильное место во фрагменте онтологии.

Уникальность концептов. Редукция в смежное наследование

Этот принцип организации общей ICF-онтологии является одним из базовых. Он гарантирует безызбыточность онтологических структур. Если существуют два или более онтологических фрагмента, находящихся по своей семантике в полной аналогии, это означает, что существует единое уникальное описание концептов соответствующей структуры, свойства которого должны унаследовать по ICF-отношению все аналогичные структуры:

$$\begin{aligned}
 \text{P6: } & \{ \text{Is-A}(X, x) \& \text{Is-A}(X, y) \& \text{Is-A}(x, z) \& \\
 & \& \text{Is-A}(x, t) \& \text{Is-A}(y, z) \& \text{Is-A}(y, t) \} \\
 & \text{-----} [\exists h \exists k \text{Is-A}(X, h) \& \text{Is-A}(X, k)], \\
 & \{ \text{Is-A}(X, h) \& \text{Is-A}(h, z) \& \text{Is-A}(h, t) \& \text{Is-A}(X, k) \& \text{Is-A}(k, x) \}
 \end{aligned}$$

Пример Структура, заданная рисунком 4, является фрагментом онтологии, пригодным для его использования в тех областях знаний, где необходимо рассматривать только системы каких-либо событий. В то же время, существуют другие понятия, которые можно рассматривать как сложные и элементарные, имеющие начало и конец. Для этого могут подойти любые сложные объекты, имеющие линейную структуру, например, очередь в магазине или железнодорожный состав. Учитывая это, можно составить общую уникальную иерархию. Представленный на рисунке 16 фрагмент онтологии позволяет рассматривать с единым понятием линейной внутренней структуры и элементы процессов, и элементы статических объектов.

Минимальность пути смежного наследования

Принцип 6 (редукция аналогичных друг другу онтологических структур в смежное наследование) предполагает необходимость отыскания места в ICF-иерархии для включения вновь разработанного фрагмента онтологии. Достаточно найти необходимое место для “привязки” лишь корневого концепта, т.к. сложность здесь заключается в соблюдении всех предыдущих принципов организации онтологии, в том числе принципа дихотомии. При правильном построении нового фрагмента, его внутренняя структура уже отвечает всем требованиям этих принципов:

$$\begin{aligned}
 \text{P7: } & \left| \bigcup_{i=1}^m \bigcup_{j=n-1}^1 \text{Is} - A(x_i^j, x_i^{j+1}) \right| = \tau \\
 & \text{-----} [\exists \mu, \mu < \tau], \\
 & \left| \bigcup_{i=1}^m \bigcup_{j=k-1}^1 \text{Is} - A(x_i^j, x_i^{j+1}) \right| = \mu
 \end{aligned}$$

где m – число терминальных вершин;

x_i^1 - корневая вершина, в которую сходятся все пути от терминальных вершин, а x_i^n - терминальные вершины. Верхняя и нижняя части правила - мощности множества всех путей от терминальных вершин к корневой как элементов бинарного отношения Is-A, соответственно до реорганизации и после нее. Суть правила в возможности реорганизации онтологии, уменьшающей мощность этого множества (рис.5).

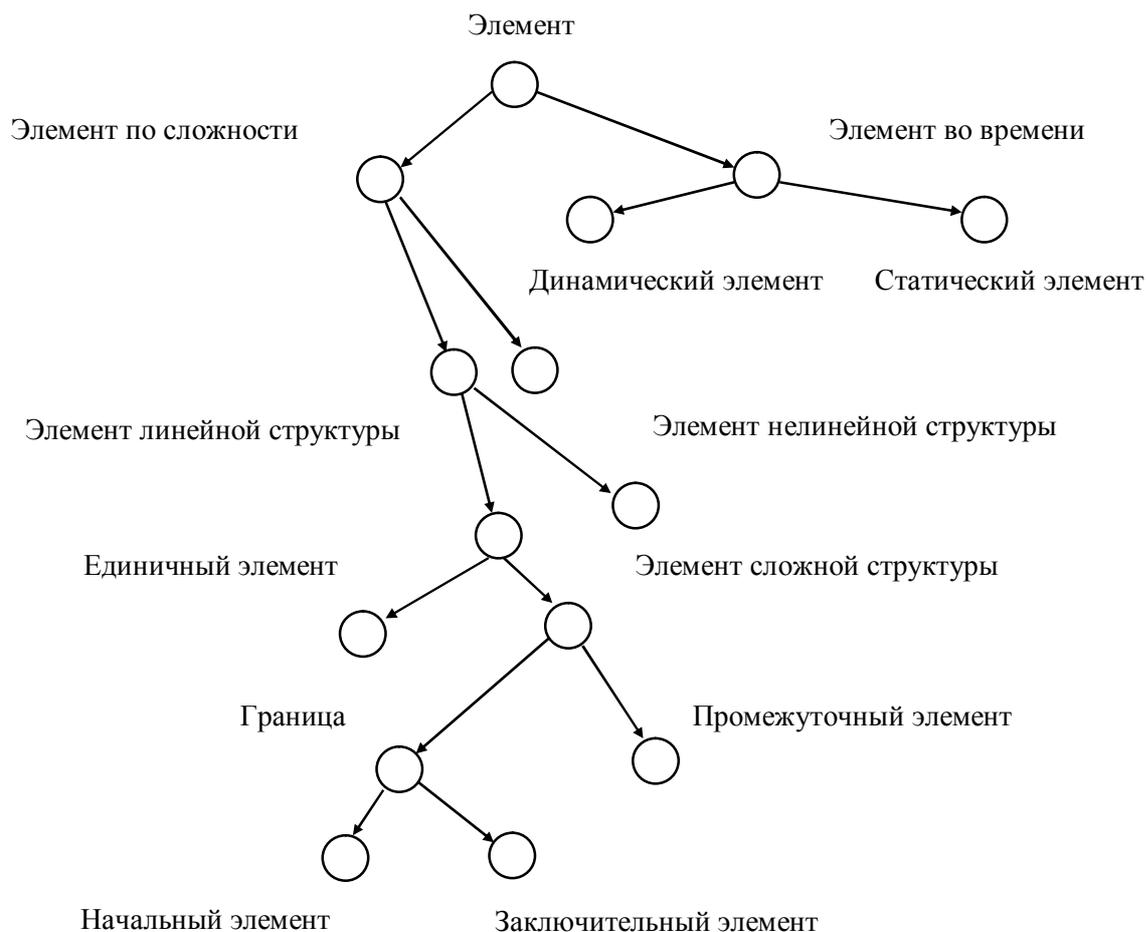


Рис. 5. Фрагмент онтологии с уникальными концептами

Предложенные правила структуризации моделей знаний на основе иерархических чисел лежат в основе технологии проектирования ICF-онтологии, замечательные свойства лаконичности и полиморфизма теоретически которой обоснованы в [2] и практически подтверждены в [3].

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. Каширин И.Ю. Нейронные сети, использующие модели знаний. IV Международный научно-технический форум СТНО-2021. Сборник трудов. Том 4, Рязань, Book Jet, 2021, С.9-13.

УДК 004.432.2; ГРНТИ 50.05.09

ИСПОЛЬЗОВАНИЕ JAVASCRIPT ИНТЕРФЕЙСА ПРИКЛАДНОГО ПРОГРАММИРОВАНИЯ ДЛЯ УПРАВЛЕНИЯ NOSQL БАЗОЙ ДАННЫХ «POUCHDB»

В.А. Антошкин, В.И. Щербакова

Рязанский государственный университет имени С.А. Есенина,
Российская Федерация, Рязань, v.antoshkin@rsu.edu.ru, v.shcherbakova2705@stud.rsu.edu.ru

Аннотация. В статье обсуждаются основы работы с NoSQL базой данных PouchDB с использованием JavaScript программного интерфейса API, которые включают базовые операции: подключения, чтения, редактирования, сохранения и удаления данных.

Ключевые слова: СУБД, базы данных, NoSQL, PouchDB, JavaScript.

USING JAVASCRIPT APPLICATION INTERFACE TO MANAGE NOSQL DATABASE «POUCHDB»

V.A. Antoshkin, V.I. Shcherbakova

Ryazan State University named after S.A. Yesenin,
Russia, Ryazan, v.antoshkin@rsu.edu.ru, v.shcherbakova2705@stud.rsu.edu.ru

The summary. This article describes the basics of working with a NoSQL database PouchDB using the JavaScript API, which includes basic operations: connect, read, edit, save and delete data.

Keywords: DBMS, databases, NoSQL, PouchDB, JavaScript.

PouchDB – это *NoSQL(Notonly SQL)* база данных с открытым исходным кодом, написанная на языке *JavaScript*, созданная для помощи разработчикам при создании веб приложений, работающих без постоянного доступа к сети.

PouchDB является базой данных, которая работает в браузере, позволяя хранить данные локально и работать с приложением автономно. Такая возможность позволяет создавать веб-приложения и работать с ними как в онлайн, так и в офлайн режимах с возможностью обращения к данным с помощью специального API интерфейса, написанного на языке JavaScript.

Эта база данных была выпущена в 2019 году в рамках *NoSQL* подхода, разработчики которой *Apache CouchDB*. Она поставляется по разрешительной открытой лицензии *ApacheLicense 2.0*.

Загрузить *PouchDB* можно с официального сайта (рис. 1).

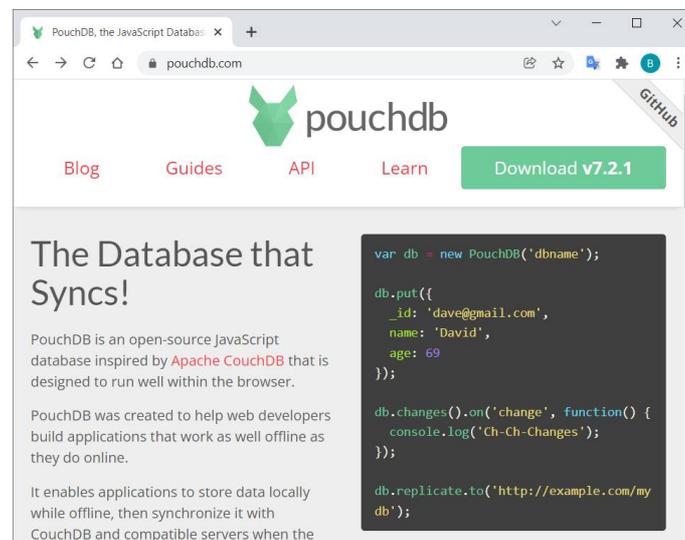
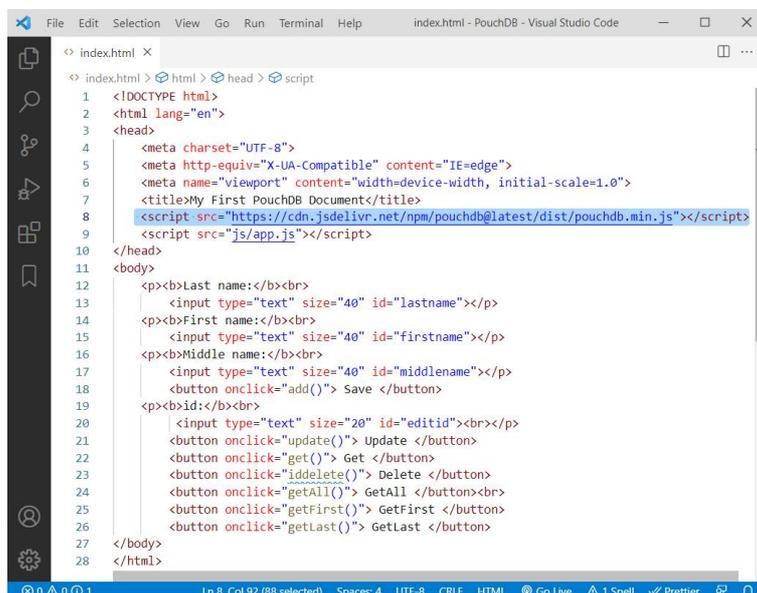


Рис. 1. Страница загрузки с сайта PouchDB.com

Для работы с базой данных *PouchDB* нужно подключить библиотеку «*pouchdb.min.js*» в заголовок любого HTML-файла, указав ссылку на нее (рис. 2).



```
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4   <meta charset="UTF-8">
5   <meta http-equiv="X-UA-Compatible" content="IE=edge">
6   <meta name="viewport" content="width=device-width, initial-scale=1.0">
7   <title>My First PouchDB Document</title>
8   <script src="https://cdn.jsdelivr.net/npm/pouchdb@latest/dist/pouchdb.min.js"></script>
9   <script src="js/app.js"></script>
10 </head>
11 <body>
12   <p><b>Last name:</b><br>
13     <input type="text" size="40" id="lastname"></p>
14   <p><b>First name:</b><br>
15     <input type="text" size="40" id="firstname"></p>
16   <p><b>Middle name:</b><br>
17     <input type="text" size="40" id="middlename"></p>
18   <button onclick="add()"> Save </button>
19   <p><b>id:</b><br>
20     <input type="text" size="20" id="editid"><br></p>
21   <button onclick="update()"> Update </button>
22   <button onclick="get()"> Get </button>
23   <button onclick="delete()"> Delete </button>
24   <button onclick="getAll()"> GetAll </button><br>
25   <button onclick="getFirst()"> GetFirst </button>
26   <button onclick="getLast()"> GetLast </button>
27 </body>
28 </html>
```

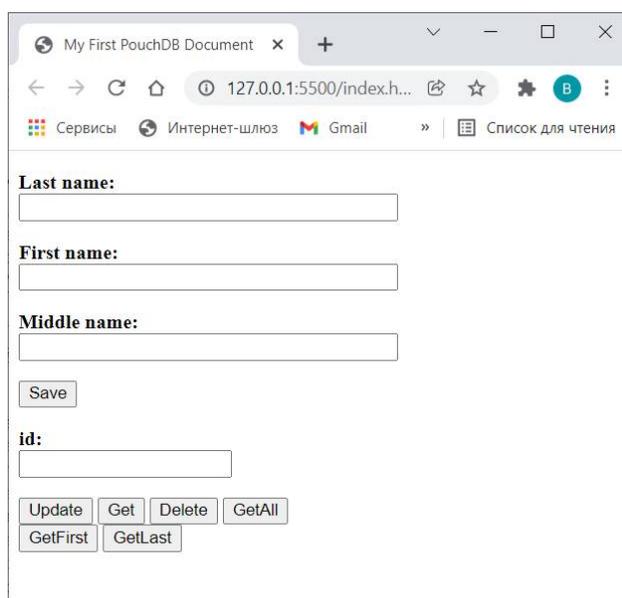
Рис. 2. Файл index.html

Доступ к данным осуществляется в асинхронном режиме.

Для работы с хранилищем объектов используются четыре базовых CRUD-операции:

1. Создание (*Create*).
2. Чтение (*Read*).
3. Обновление (*Update*).
4. Удаление (*Delete*).

Для демонстрации работы всех этих операций был разработан несложный интерфейс веб-страницы (рис. 3), который позволяет задавать фамилию, имя и отчество пользователя и сохранять эти значения в локальной базе данных.



My First PouchDB Document

127.0.0.1:5500/index.h...

Сервисы Интернет-шлюз Gmail Список для чтения

Last name:

First name:

Middle name:

Save

id:

Update Get Delete GetAll
GetFirst GetLast

Рис. 3. Интерфейс веб-страницы для работы с данными

Изначально в хранилище нет объектов, только присутствуют пустые поля Ключ и Значение (рис. 4).

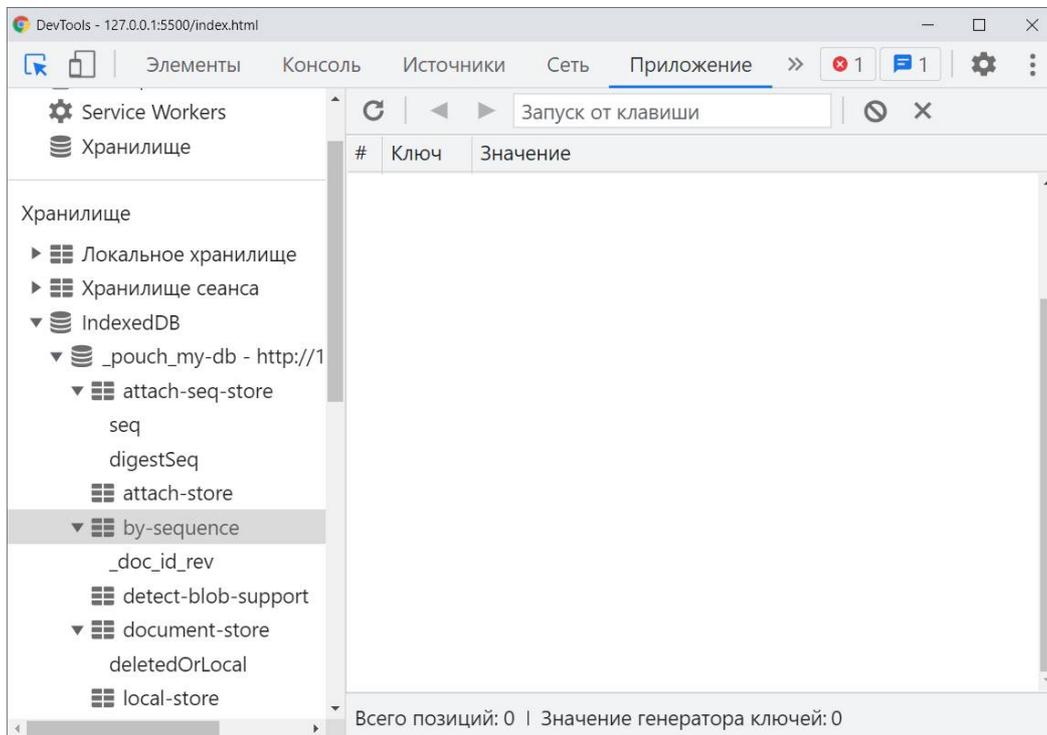


Рис. 4. Пустое хранилище PouchDB

Для сохранения нового объекта в хранилище можно выполнить с помощью специального метода «put» (рис. 5). Поле `_id` является обязательным для создания объекта в хранилище и должно быть уникальным. Данное поле потребуется для получения файла.

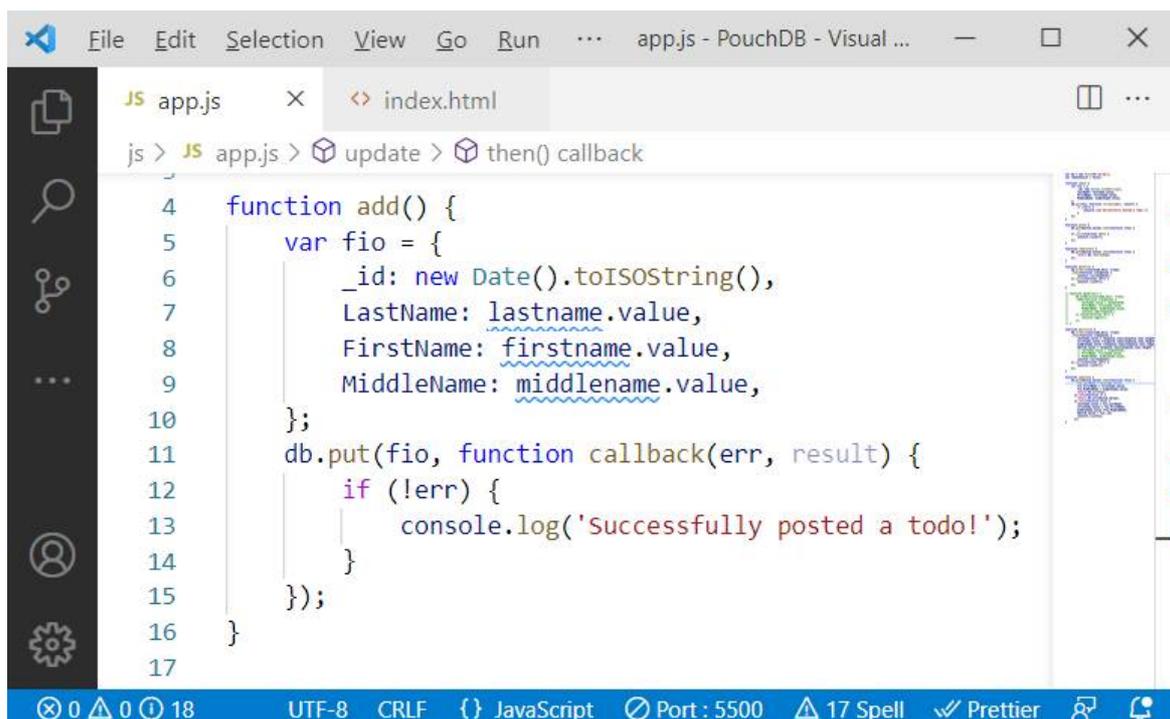


Рис. 5. Сохранение данных в хранилище объектов «my-db»

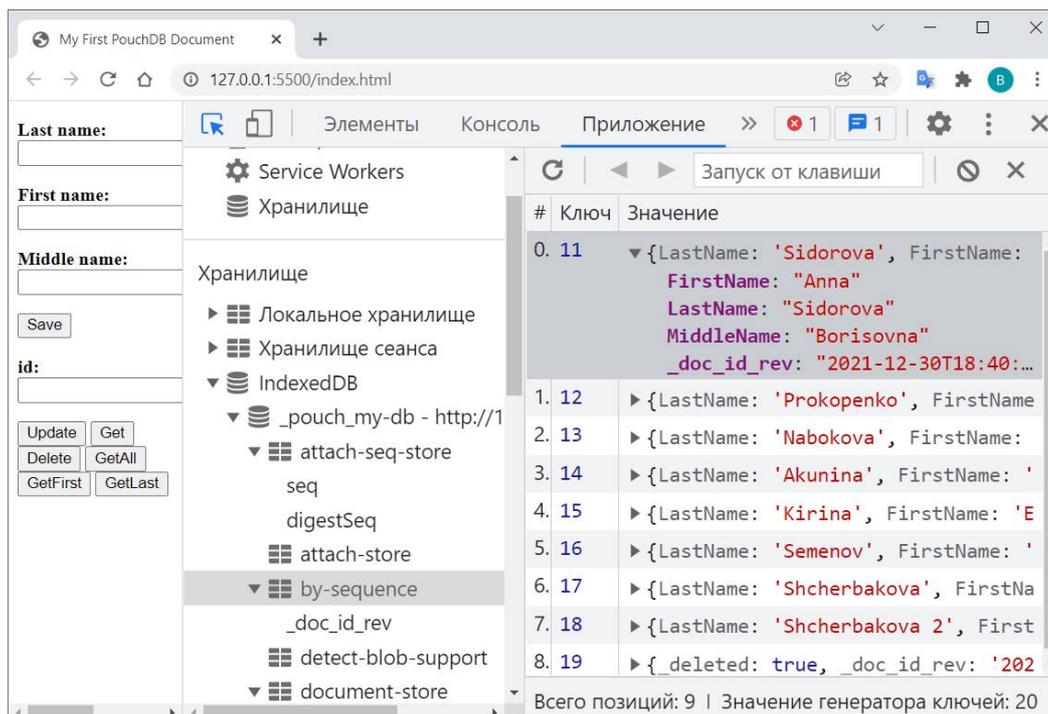


Рис. 6. Добавленные объекты в хранилище «my-db»

Получить список всех объектов, находящихся в хранилище, можно с помощью специального метода «*allDocs*» (рис. 7).

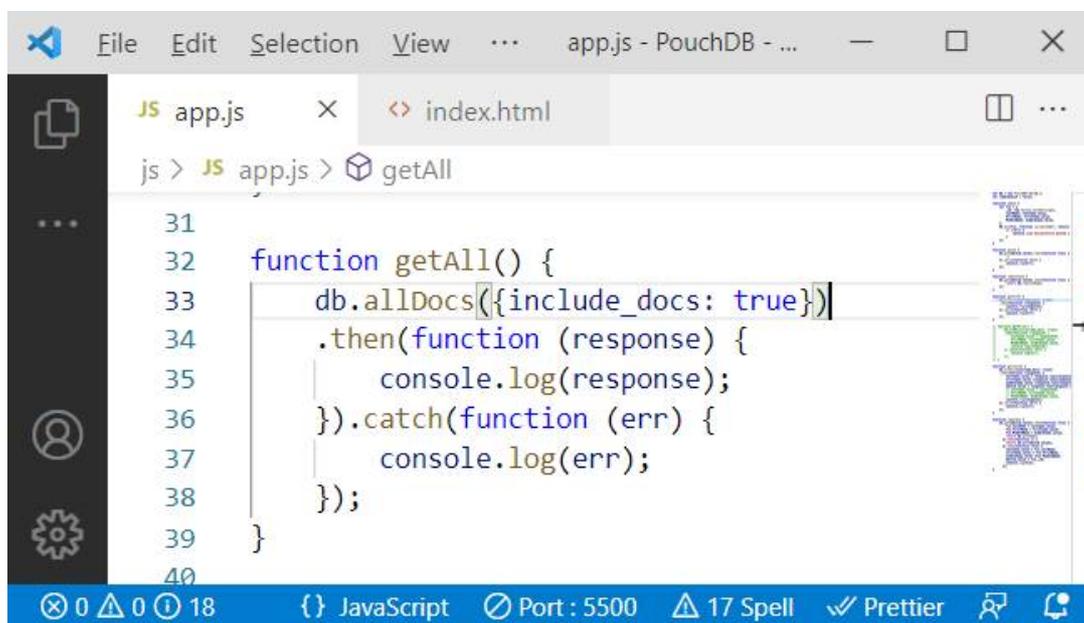
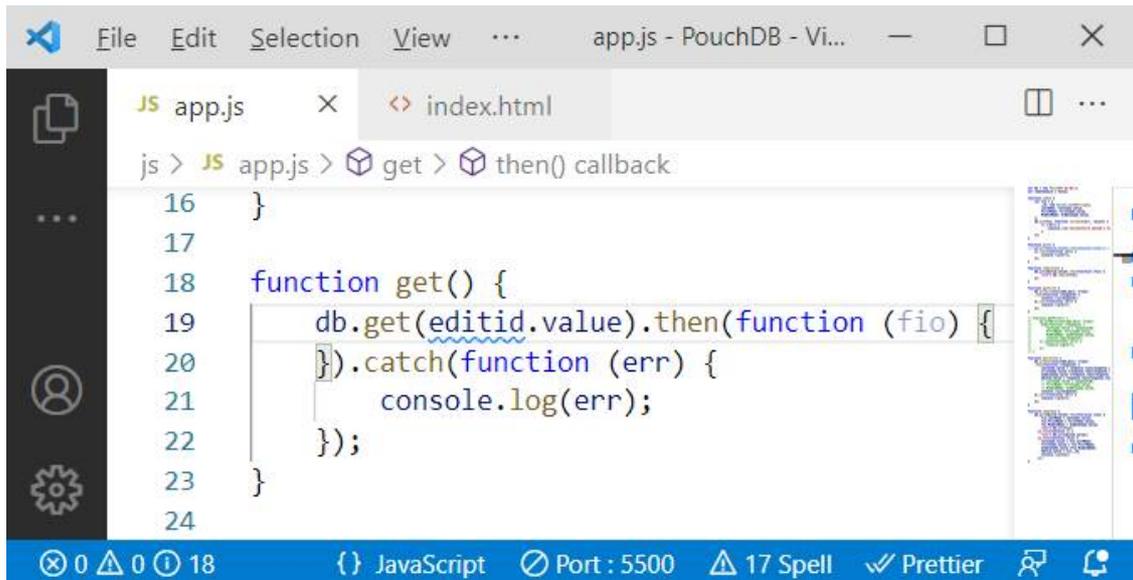


Рис. 7. Чтение всех объектов из хранилища «my-db»

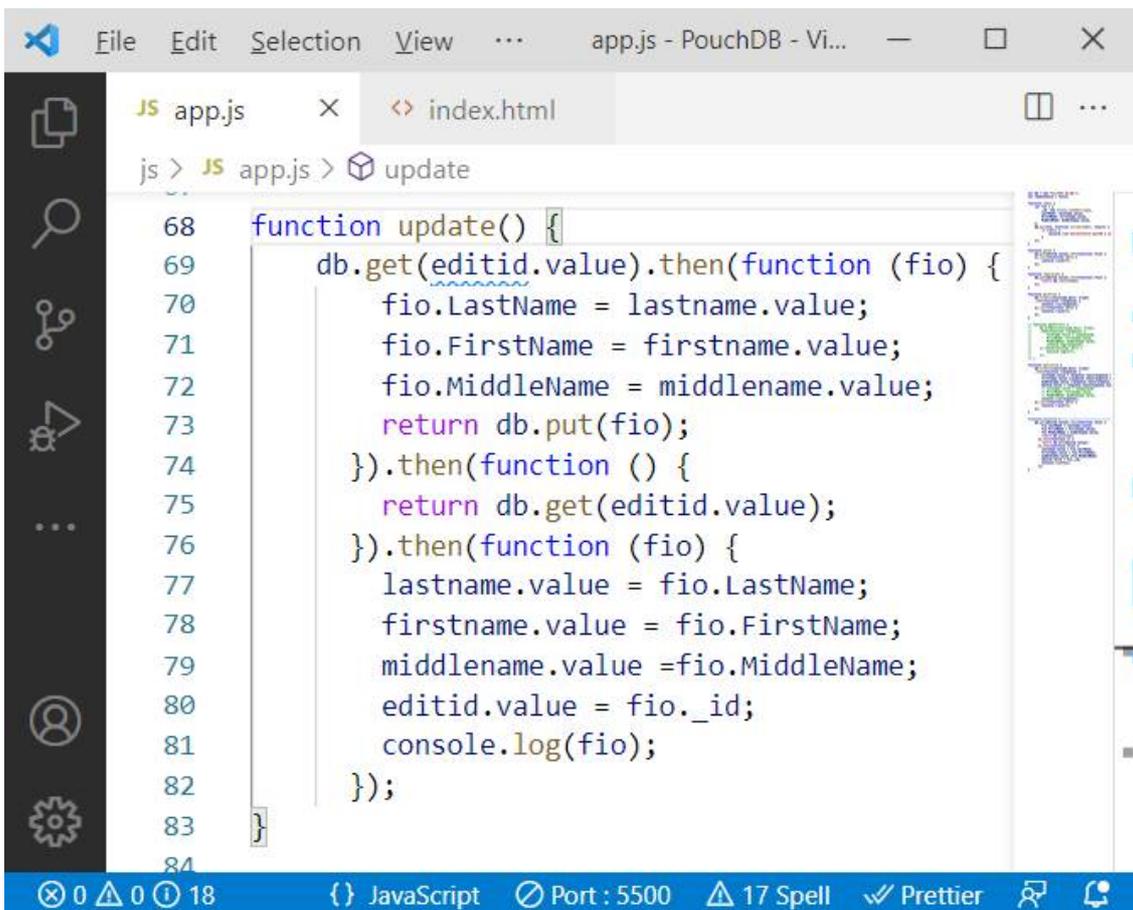
Для получения определенного объекта из хранилища вместо метода «*allDocs*» необходимо использовать метод с именем «*get*». Он в качестве параметра принимает первичный ключ объекта и возвращает все его данные, если такой объект в хранилище существует (рис. 8).



```
File Edit Selection View ... app.js - PouchDB - Vi...
JS app.js x <> index.html
js > JS app.js > get > then() callback
16 }
17
18 function get() {
19   db.get(editid.value).then(function (fio) {
20     }).catch(function (err) {
21       console.log(err);
22     });
23 }
24
0 0 18 JavaScript Port: 5500 17 Spell Prettier
```

Рис. 8. Получение данных определенного объекта

Для обновления данных определенного объекта необходимо использовать специальный метод «put» (рис. 9).



```
File Edit Selection View ... app.js - PouchDB - Vi...
JS app.js x <> index.html
js > JS app.js > update
68 function update() {
69   db.get(editid.value).then(function (fio) {
70     fio.LastName = lastname.value;
71     fio.FirstName = firstname.value;
72     fio.MiddleName = middlename.value;
73     return db.put(fio);
74   }).then(function () {
75     return db.get(editid.value);
76   }).then(function (fio) {
77     lastname.value = fio.LastName;
78     firstname.value = fio.FirstName;
79     middlename.value = fio.MiddleName;
80     editid.value = fio._id;
81     console.log(fio);
82   });
83 }
84
0 0 18 JavaScript Port: 5500 17 Spell Prettier
```

Рис. 9. Обновления данных определенного объекта

Этому методу необходимо передавать в списке параметров данные обновляемого объекта с обязательным указанием уникального идентификатора. При этом в списке будут храниться два объекта, но с разными ревизиями (рис. 10).

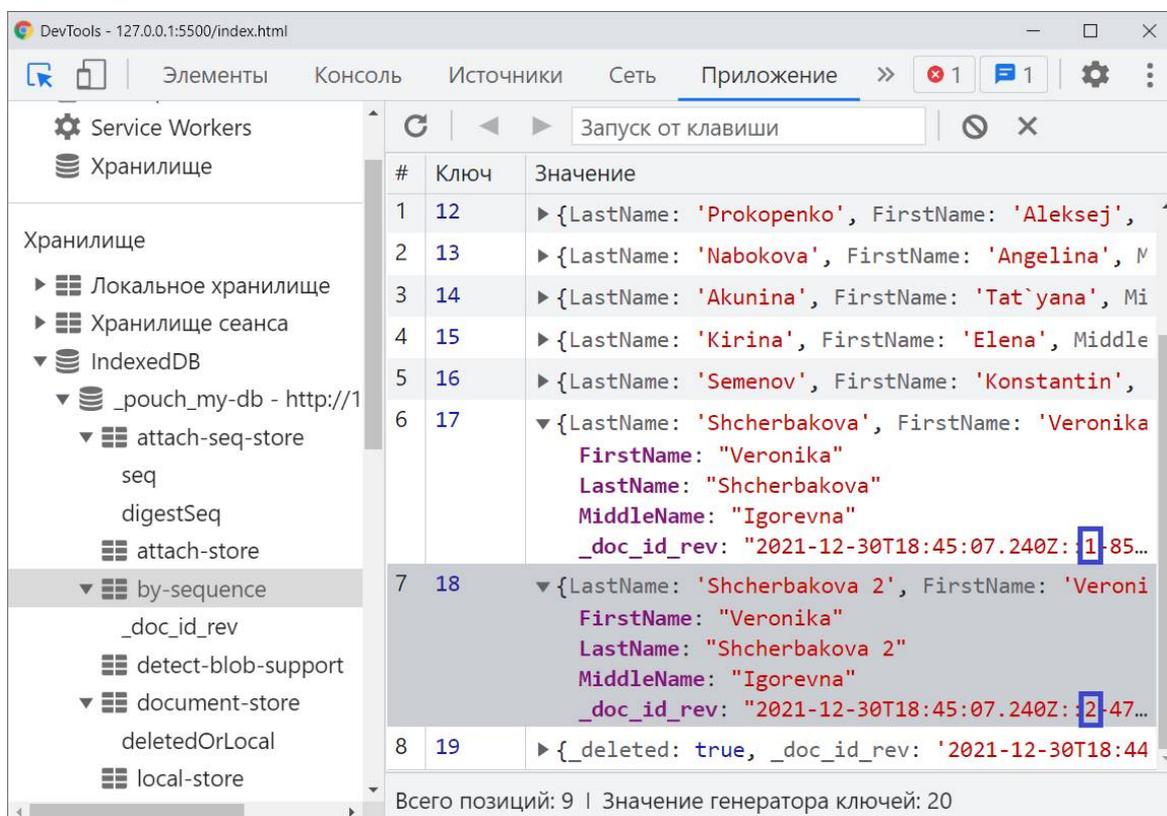


Рис. 10. Обновления данных определенного объекта

Таким образом возможно отследить изменения по ревизии объекта.

Для выполнения удаления определенного объекта необходимо использовать специальный метод *«remove»*, которому в качестве единственного параметра передается уникальный идентификатор удаляемого объекта (рис. 11).

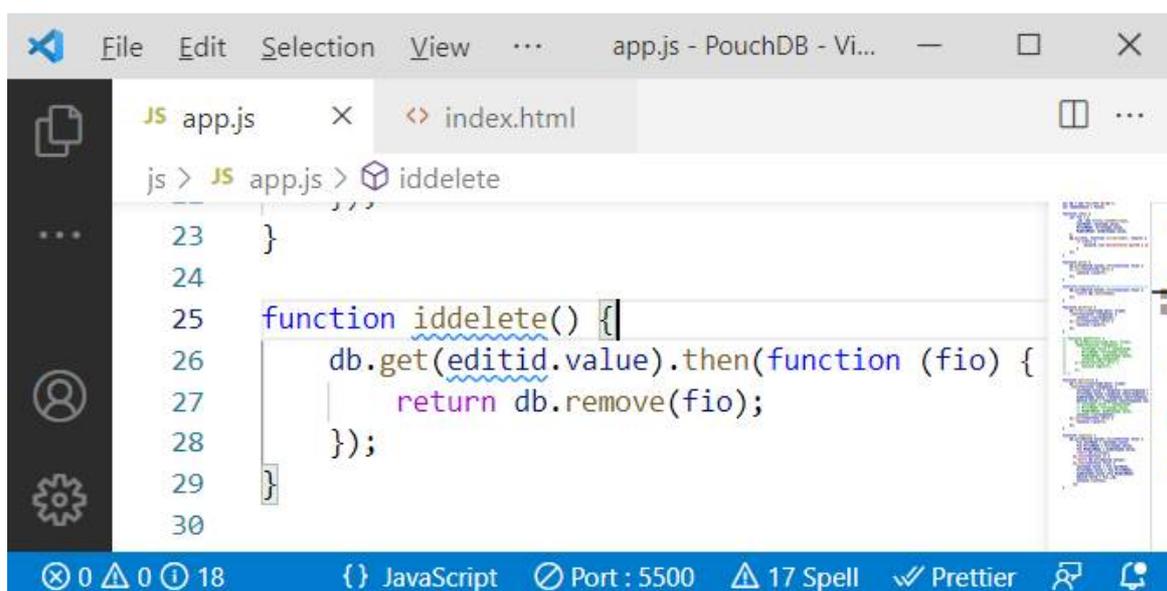


Рис. 11. Удаления определенного объекта из хранилища

После удаления объект остается в списке хранилища, при этом имеет пометку «**_deleted: true**» (рис. 12).

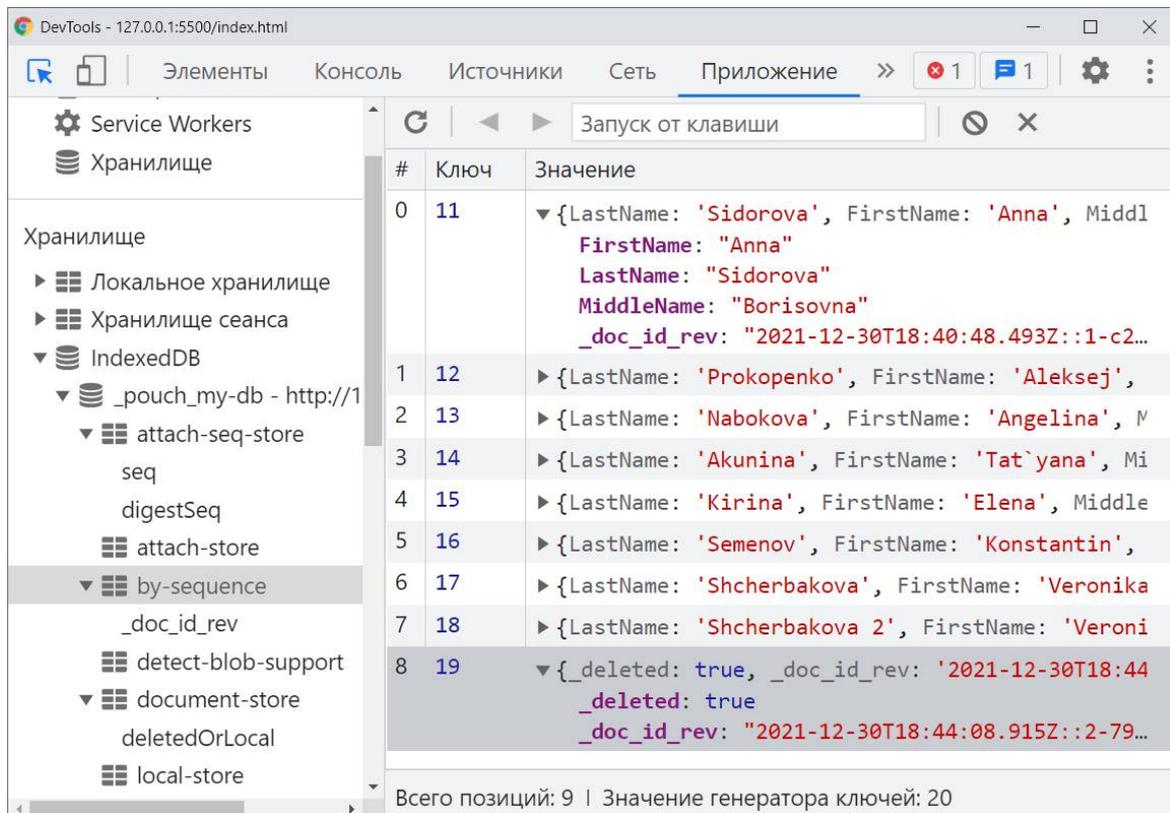


Рис. 12. Удаленный объект в списке хранилища

Таким образом, NoSQL базу данных PouchDB можно использовать для эффективного хранения данных в локальном хранилище браузера с возможностью их дальнейшего экспорта в другие базы, например, CouchDB.

Библиографический список

1. PouchDB [Электронный ресурс]: сайт. URL: <https://ru.bmstu.wiki/PouchDB> (дата обращения: 22.12.2021)
2. API Reference [Электронный ресурс] : сайт. URL: <https://pouchdb.com/api.html#overview> (дата обращения: 22.12.2021)
3. PouchDB, the JavaScript Database [Электронный ресурс] : сайт. URL: <https://pouchdb.com/> (дата обращения: 22.12.2021)
4. ApacheCouchDB [Электронный ресурс] // Википедия. Свободная энциклопедия. URL: https://en.wikipedia.org/wiki/Apache_CouchDB (дата обращения: 22.12.2021).

УДК 004.432.2; ГРНТИ 50.05.09

АНАЛИЗ КЛАССОВ JDK. ПОДГОТОВКА ИСХОДНЫХ ДАННЫХ

А.В. Горкин

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

Аннотация. Анализ классов из состава JavaClassLibraries (один из компонентов Java Development Kit) с помощью вычисления метрик их программного кода позволит соотнести теоретические рекомендации по грамотному написанию текстов программ с показателями, полученными в ходе анализа ядра языка программирования. Предлагается метод и инструмент для вычисления показателей программного кода, а также выделяется набор метрик, по которому будет произведен анализ.

Ключевые слова: JDK, OpenJDK, Java, ООП, Метрики программного кода.

JDK CLASSES ANALYSIS. INITIAL DATA PREPARATION

A.V. Gorkin

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

The summary. Analysis of classes from the Java Class Libraries (one of the components of the Java Development Kit) by calculating the metrics of their program code will allow you to correlate theoretical recommendations for the competent writing of program texts with the indicators obtained by analyzing the core of the programming language. A method and a tool for calculating program code indicators are proposed, as well as a set of metrics is proposed, according to which the analysis will be performed.

Keywords: JDK, OpenJDK, Java, OOP, Software code metrics.

JavaDevelopmentKit (JDK) – комплект разработчика приложений на языке Java. Набор содержит в себе ряд библиотек, образующих ядро языка программирования. С течением времени выпускаются более новые версии языка, в которых добавляется новая функциональность, а также производится рефакторинг старого кода. Обновление языка происходит периодически, на момент написания статьи (январь, 2022) последней доступной версией являлась JDK17, которая была представлена в 2021 году. Ожидаемая дата выхода JDK18 – март 2022 года [1].

В промышленной разработке важно писать код, который будет не только отвечать поставленным бизнес требованиям, но в то же время будет понятен другим разработчикам, а также легко изменяться и масштабироваться. Одним из методов проверки соответствия программного кода этим требованиям является его анализ посредством вычисления набора метрик. Существует ряд рекомендаций по написанию «чистого» программного кода [2]. Но эта информация носит более теоретический характер. Целью исследования является анализ исходных текстов классов JDK. Подобный анализ позволит понять в каком стиле написан исходный код языка Java, и как фактические значения метрик программного кода ядра языка соотносятся с теоретическими рекомендациями. Анализ будет производиться в разрезе различных версий Java, чтобы была возможность проанализировать тенденции изменения стилистики в написании исходных текстов программ. В результате анализа планируется получить численные значения метрик программного кода ядра языка Java различных версий.

Существует ряд сборок JDK, часть из которых носят коммерческий характер. Так, версии OracleJDK, начиная с 16 апреля 2019 года (Java 8 SE Update 211), являются платными. Анализ таких сборок затруднителен и не будет производиться в рамках исследования. Но существуют также сборки JDK с открытым исходным кодом, такие как OpenJDK. Основным их недостатком является более краткосрочная поддержка. Этот критерий не столь важен применительно к анализу исходных текстов программ. Таким образом, в рамках исследования будем рассматривать классы из состава OpenJDK.

Ранее было представлено программное обеспечение, позволяющее вычислить метрики программного кода [3]. Основной принцип работы утилиты заключается в следующем. На вход программе передается папка с исходными текстами программы (для языка Java – файл с расширением .java), после чего в каждом из файлов происходит выделение конструкций языка (начиная с верхнего уровня – класса, заканчивая низкоуровневыми конструкциями, такие как оператор if). Далее, на основе построенной древовидной структуры конструкций языка происходит вычисление различных метрик программного кода, которые представляются пользователю. Присутствует также функциональность по экспорту метрик в табличный формат (.xls – Microsoft Excel). JDK в своем составе содержит несколько тысяч классов, поэтому анализ совокупности всех сущностей затруднителен с использованием существующего программного средства. В программное обеспечение будет добавлена возможность формировать сводный отчет по всем классам в виде табличного документа по заданному набору выходных метрик.

Для того, чтобы загрузить набор OpenJDK на ваш компьютер:

- 1) можно воспользоваться загрузкой с веб сайта;
- 2) пользователям Linux/MacOs удобно использовать для этой задачи утилиту командной строки Homebrew [4].

Набор JDK состоит из ряда компонентов (рисунок 1), все из которых были загружены на предыдущем шаге. В рамках исследования нас интересует компонент Java Class Libraries (JCL, библиотеки классов Java). После загрузки (установки) набора OpenJDK, стоит извлечь непосредственно исходные тексты ядра языка, они находятся в архиве src.zip. Пример расположения src.zip для AdoptOpenJDK-8 в операционной системе MacOS: /Library/Java/JavaVirtualMachines/adoptopenjdk-8.jdk/Contents/Home/src.zip. Далее следует извлечь директорию src из архива src.zip. Мы получили набор исходных данных для анализа. Так, например, исходный текст класса java.util.ArrayList находится по следующему пути: src/java/util/ArrayList.java.

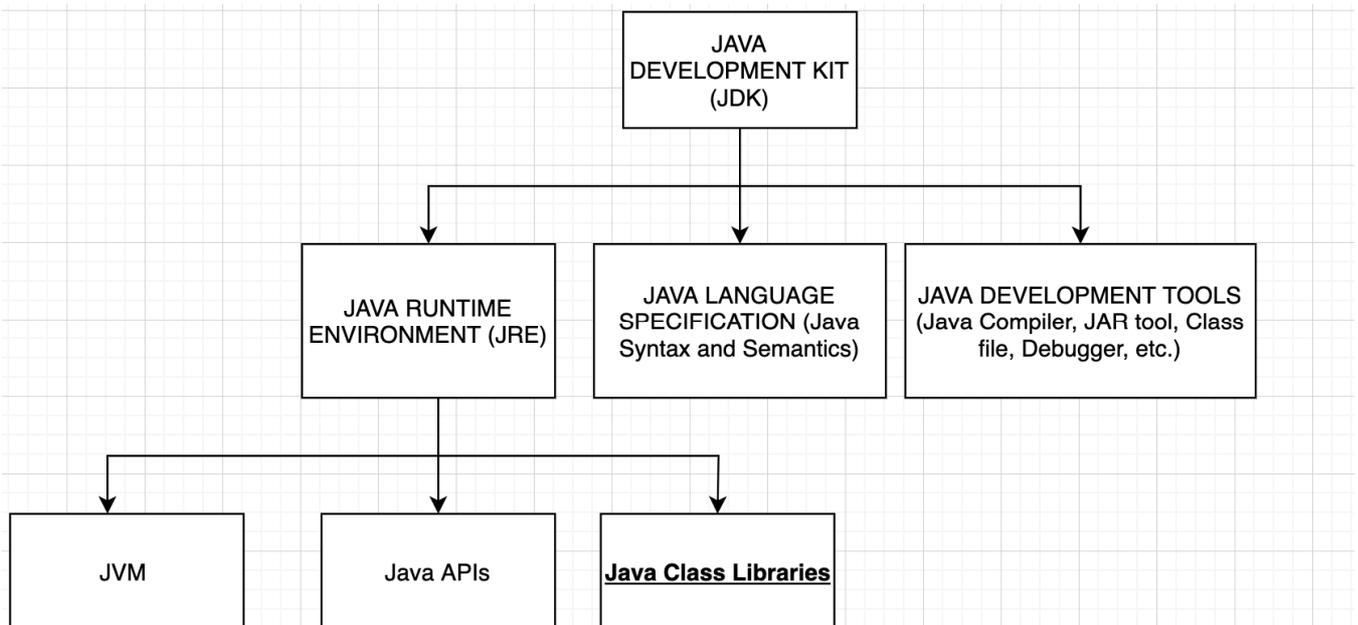


Рис. 1. Компоненты JDK

Исходные тексты классов JDK будем анализировать по следующим метрикам:

Цикломатическая сложность кода. Характеристика, от которой напрямую зависит сложность поддержки кода. В первую очередь значение метрики зависит от количества вложенных операторов ветвления и циклов. Чем индекс ниже, тем лучше, и тем легче в будущем

будет менять структуру кода. При анализе текстов программ следует учитывать рекомендуемые значения цикломатической сложности кода [5] (Таблица 1).

Таблица 1. Рекомендуемые значения цикломатической сложности кода

Сложность кода	Описание
1–10	Хорошо написанный код, легко тестируем, дешев в поддержке
10–20	Код средней сложности, средняя сложность тестирования, средняя стоимость поддержки
20–40	Очень сложный код, тяжело тестировать, дорог в обслуживании
>40	Практически невозможно полностью протестировать программный код

Размер блоков кода. Под блоком кода подразумевается любой метод или конструктор. Существует рекомендация проектировать методы таким образом, чтобы они состояли из ≤ 15 строк кода [6]. Но как правило, из-за специфики предметной области не удается следовать такому жесткому ограничению. Поэтому вводится более лояльная шкала оценивания кода (Таблица 2), состоящая из 4 требований. Например, если в вашей программе есть 2 метода с более 60 строк кода (61 и 70 строк кода), то в программе должно содержаться как минимум $((61+70)/0,069) = 1899$ строк кода.

Таблица 2. Требования к коду по критерию объема методов/конструкторов

Количество строк в методе	Отношение к общему числу строк в программе
>60	Не более 6,9%
>30	Не более 22,3%
>15	Не более 43,7%
≤ 15	Как минимум 56,3%

АВСметрика – метрика, которая позволяет измерять как объем, так и сложность программного кода. В ее основе лежит подсчет 3 составляющих: присвоений (A – assignments), ветвлений (B – branches) и условий (C – conditions). Численное значение вычисляется следующим образом: $|ABC| = \sqrt{(A*A) + (B*B) + (C*C)}$. Применяя АВСметрику к отдельному методу, можно определить его сложность. Рекомендуемые значения сложности метода по метрике АВС[7] представлены в таблице 3.

Таблица 3. Рекомендуемые значения метрики АВС для отдельного метода

ABC	Описание
0 – 10	Отлично
10 – 20	Очень хорошо
21 – 40	Хорошо, но возможно уже стоит задуматься о рефакторинге кода
40 – 60	Требуется определить, оправдана ли используемая сложность
61 – 100	Опасные области кода (требуется рефакторинг)
100 +	Очень опасные области кода (требуется рефакторинг)

Выводы:

1. Анализ исходных классов языка Java предмет метрик программного кода позволит понять, насколько соотносятся теоретические рекомендации написания кода программ с реальным их применением в языке программирования.
2. Исходные классы языка Java (JCL – JavaClassLibraries) являются частью JDK (находятся в архиве src.zip) и открыты для анализа в opensourceверсиях JDK.

3. Из-за специфики анализа (общий объем JDK составляет несколько тысяч классов) разработанное ранее средство по вычислению метрик требуется доработать.
4. Определены метрики, по которым будут анализироваться классы JDK и выделены области их рекомендуемых значений. В расчетах будет использоваться метрика цикломатической сложности кода, ABC метрика, а также метрика размеров блоков кода.

Библиографический список

1. JDK 18. – Текст: электронный ресурс // OpenJdk — URL: <https://openjdk.java.net/projects/jdk/18/> (дата обращения: 18.01.2022).
2. Мартин Р. Чистый код: создание, анализ и рефакторинг. Библиотека программиста. –СПб.: Питер, 2013. – 464 с.
3. Горкин А.В. Программное обеспечение сбора статистики и анализа структур программ на языке Java // Новые информационные технологии в научных исследованиях: материалы XXV Всероссийской научно-технической конференции студентов, молодых ученых и специалистов; Рязань: ИП Коняхин А. В. (BookJet), 2020 – С. 168-171.
4. Homebrew– Текст: электронный –URL: <https://brew.sh/> (дата обращения 22.01.2022)
5. Desikan S. Software Testing: Principles and Practice, 2006. – 486с.
6. Joost V., Rigal S., Leek R., Eck P., Wijnholds G. Building Maintainable Software, Java Edition, 2016 – 168с.
7. GMetrics – Текст: электронный ресурс // GMetricsdocumentation – URL: <https://dx42.github.io/gmetrics/metrics/AbcMetric.html> (дата обращения: 07.02.2022).

УДК 004.032.26; ГРНТИ 28.23.37

ИСПОЛЬЗОВАНИЕ БИБЛИОТЕКИ KERAS ДЛЯ ОБУЧЕНИЯ НЕЙРОСЕТЕЙ ВЫПОЛНЕНИЮ ПРОСТЕЙШИХ ЛОГИЧЕСКИХ ОПЕРАЦИЙ

В.А. Антошкин, А.И. Иванов

*Рязанский государственный университет имени С.А. Есенина,
Российская Федерация, Рязань, v.antoshkin@rsu.edu.ru, sasha9182-2015@yandex.ru*

Аннотация. В статье рассмотрен принцип построения нейронной сети с помощью библиотеки машинного обучения Keras для языка программирования Python. Исследовано применение однослойной нейронной сети для реализации простейших логических операций.

Ключевые слова: нейронная сеть, обучение нейросети, Keras, логические операции, Python.

USING THE KERAS LIBRARY FOR TRAINING NEURAL NETWORKS TO PERFORM THE SIMPLEST LOGICAL OPERATIONS

V.A. Antoshkin, A.I. Ivanov

*Ryazan State University named for S.A. Yesenin,
Russia, Ryazan, v.antoshkin@rsu.edu.ru, sasha9182-2015@yandex.ru*

The summary. The article discusses the principle of building a neural network using the Keras machine learning library for the Python programming language. The use of a single-layer neural network for the implementation of the simplest logical operations has been studied.

Keywords: neural network, training a neural network, Keras, logical operations, Python.

Согласно диаграмме популярности библиотек машинного обучения (рис. 1), *TensorFlow* является самым популярным фреймворком для искусственного интеллекта.

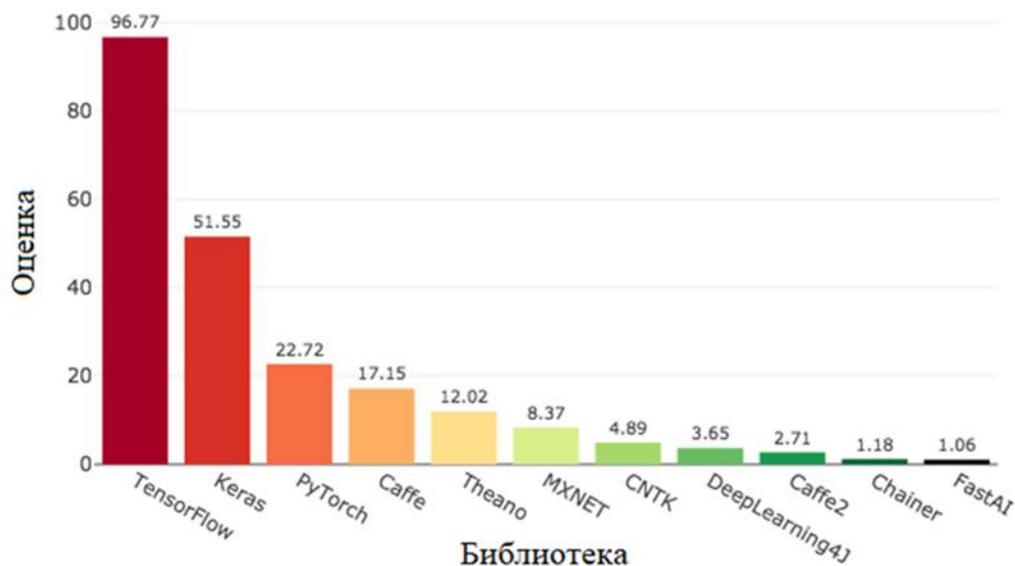


Рис. 1. Диаграмма популярности библиотек машинного обучения

Библиотека *TensorFlow*, разработанная компанией *Google*, позволяет строить нейросети различной архитектуры. Однако данный фреймворк является относительно низкоуровневым, поэтому можно воспользоваться библиотекой *Keras*, которая тоже является одной из самых популярных в машинном обучении. Её применение поможет упростить и ускорить проектирование нейронной сети. К тому же *Keras* представляет собой оболочку для *TensorFlow*, поэтому она не требует отдельной установки и тоже является быстрой и эффективной библиотекой.

Рассмотрим применение нейросети для реализации конъюнкции – логической операции «И». Для построения нейронной сети необходимо создать файл на языке программирования Python. Затем в файле необходимо подключить библиотеку машинного обучения *Keras* (рис. 2), входящую в состав фреймворка *TensorFlow*. Для удобства отдельно импортируется класс *Dense*, входящий в библиотеку *Keras* и описывающий слой нейросети. Также нужно подключить библиотеки *NumPy* и *Matplotlib*. Библиотека *NumPy* используется для создания массивов входных и выходных данных, а *Matplotlib* применяется для дальнейшей обработки результатов, полученных при обучении модели нейросети [1, 2].

```
import numpy as np
import matplotlib.pyplot as plt
from tensorflow import keras
from tensorflow.keras.layers import Dense
```

Рис. 2. Подключение необходимых библиотек

Входные и выходные данные состоят из 0 и 1 (0 – «ложь», 1 – «истина»). Входной массив представляет собой четыре различных сочетания нулей и единиц, а выходной – ре-

зультат применения логической операции к каждому сочетанию. Массивы создаются с помощью библиотеки *NumPy* (рис. 3).

```
# Входные и выходные данные
input_data = np.array([[0, 0], [1, 0], [0, 1], [1, 1]])
output_data = np.array([[0], [0], [0], [1]])
```

Рис. 3. Массивы входных и выходных данных

Нейросеть строится на основе последовательной модели, которая описывается с помощью класса *Sequential* (рис. 4). Архитектура такой модели состоит из последовательности нейронных слоёв. Такая модель удобна для построения нейросети: каждый новый слой добавляется в модель, являющуюся экземпляром класса *Sequential*, с помощью метода *add()*. Класс *Dense*, используемый в качестве аргумента метода *add()*, позволяет описать нейронный слой. Применяемая в статье нейросеть состоит из одного слоя (рис. 5), поскольку входной слой не учитывается приобщем подсчёте слоёв. На входе нейросети расположены 2 нейрона, что соответствует количеству операндов. Выходной слой состоит из одного нейрона (результат применения логической операции к операндам). В качестве функции активации выбирается сигмоида [1, 2].

```
# Создание модели нейросети
model = keras.Sequential()
model.add(Dense(units=1, input_shape=(2,), activation="sigmoid"))
```

Рис. 4. Создание модели нейросети

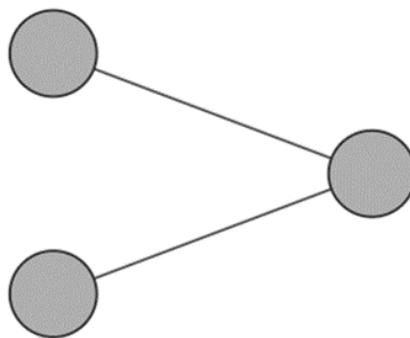


Рис. 5. Однослойная нейронная сеть

После построения нейросети её необходимо скомпилировать, вызвав одноимённый метод (рис. 6). В качестве аргументов метода *compile()* необходимо задать функцию потерь нейронной сети (в данном случае минимум среднеквадратичной ошибки) и оптимизатор (здесь выбрана оптимизация по *Adam*, а шаг сходимости алгоритма обучения равен 0,1). В процессе компиляции нейросети веса автоматически инициализируются начальными значениями [1, 3].

```
# Компиляция модели
model.compile(loss="mean_squared_error", optimizer=keras.optimizers.Adam(0.1))
```

Рис. 6. Компиляция модели нейросети

После компиляции модели нейросети нужно произвести её обучение (рис. 7). Для этого применяется метод *fit()*, при вызове которого указываются следующие аргументы: входной и выходной массивы; максимальное количество эпох обучения; параметр, запрещающий выводить в консоль информацию о процессе обучения нейросети; а также функция обратного вызова, которая работает с объектом, позволяющим заранее завершить обучение нейросети. В качестве такого объекта выступает класс, основанный на методе ранней остановки, который применяется для борьбы с переобучением нейросетей. В данном случае обучение останавливается, когда потери практически перестают уменьшаться. Результаты обучения сохраняются в объект *log* [1, 2, 3].

```
# Обучение модели
callback = keras.callbacks.EarlyStopping(monitor="loss", min_delta=5e-06, patience=2)
log = model.fit(input_data, output_data, epochs=800, verbose=0, callbacks=callback)
```

Рис. 7. Обучение нейросети

Затем в консоли указывается номер эпохи, на которой было остановлено обучение нейросети (рис. 8).

```
# Номер эпохи
stop = callback.stopped_epoch
print(f"Обучение остановлено на {stop}-й эпохе")
```

Рис. 8. Вывод номера эпохи, на которой было остановлено обучение

История, полученная в процессе обучения, выводится в виде графика с помощью библиотеки *Matplotlib* (рис. 9). График обучения представляет собой зависимость функции потерь нейронной сети от числа эпох обучения. На графике видно, что с увеличением числа эпох потери уменьшаются (рис. 13) [2].

```
# График обучения (зависимость потерь от числа эпох)
plt.plot(log.history["loss"])
plt.grid(True)
plt.show()
```

Рис. 9. Вывод графика обучения

Метод `get_weights()` позволяет получить матрицу весов (рис. 10) [3].

```
# Получение матрицы весов
print("Матрица весов:")
print(model.get_weights())
```

Рис. 10. Вывод матрицы весов в консоль

Для проверки данной нейросети применяется массив входных данных. Он используется в качестве аргумента для метода `predict()` (рис. 11) [1, 2].

```
# Проверка модели с помощью входных данных
results = model.predict(input_data)
```

Рис. 11. Проверка модели нейросети

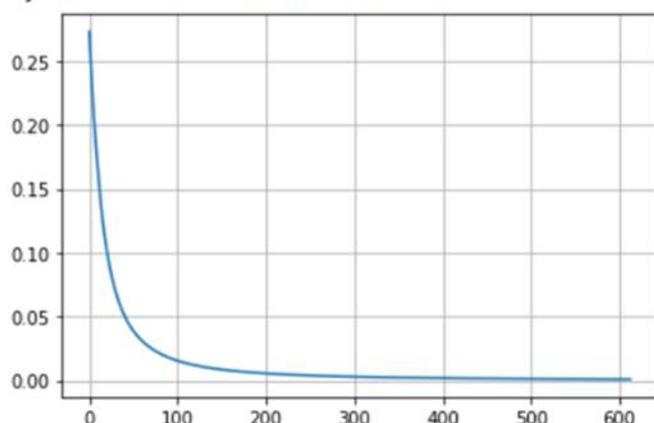
Затем можно вывести результаты проверки в консоль (рис. 12).

```
# Вывод результатов проверки в консоль
print("-" * 80, "\nПроверка модели:")
for i, data in enumerate(input_data):
    print(f"{data[0]} & {data[1]} = {results[i][0]}")
```

Рис. 12. Вывод результатов проверки модели в консоль

Все результаты, полученные при работе с нейросетью, представлены на рисунке 13.

Обучение остановлено на 612-й эпохе



Матрица весов:

```
[array([[6.6457105],
        [6.5671487]], dtype=float32), array([-9.952187], dtype=float32)]
```

Проверка модели:

```
0 & 0 = 4.76233872177545e-05
1 & 0 = 0.03534969687461853
0 & 1 = 0.032766371965408325
1 & 1 = 0.9630547761917114
```

Рис. 13. Результаты работы нейросети

Для проверки других простейших логических операций (дизъюнкция и отрицание) нейросеть строится аналогично, но необходима некоторая коррекция кода программы. Операция дизъюнкции требует изменения массива выходных данных (рис. 14), а в коде, который выводит информацию о проверке нейросети (рис. 12), символ «&» заменяется на символ «|».



Массив выходных данных

```
output_data = np.array([[0], [1], [1], [1]])
```

Рис. 14. Массив выходных данных для операции «ИЛИ»

Чтобы проверить операцию отрицания, нужно изменить входной и выходной массивы, количество входных нейронов поменять на 1 (рис. 15); также необходимо отредактировать код для вывода результатов проверки работы нейросети (рис. 16).



Входные и выходные данные

```
input_data = np.array([[0], [1]])
output_data = np.array([[1], [0]])
```

Создание модели нейросети

```
model = keras.Sequential()
model.add(Dense(units=1, input_shape=(1,), activation="sigmoid"))
```

Рис. 15. Изменение данных и модели нейросети для операции «НЕ»

```

▶ # Вывод результатов проверки в консоль
print("-" * 80, "\nПроверка модели:")
for i, data in enumerate(input_data):
    print(f"NOT {data[0]} = {results[i][0]}")

```

Рис. 16. Проверка операции отрицания

Результаты реализации всех рассмотренных логических операций с помощью нейросети сведены в табл. 1.

Таблица 1. Реализация логических операций с помощью нейросети

X_1	X_2	$\text{NOT}X_1$	$X_1 \text{ AND } X_2$	$X_1 \text{ OR } X_2$
0	0	0,971	$4,762 \cdot 10^{-5}$	0,038
0	1	0,971	0,035	0,978
1	0	0,028	0,033	0,974
1	1	0,028	0,963	1,000

Полученные результаты позволяют заметить, что однослойная нейросеть достаточно точно реализует простейшие логические операции, однако она не подходит для реализации исключаящего «ИЛИ». Для выполнения данной операции необходимо добавить хотя бы один скрытый слой с двумя нейронами [4, 5].

Библиографический список

1. Джулли, А. Библиотека Keras – инструмент глубокого обучения. Реализация нейронных сетей с помощью библиотек Theano и TensorFlow / пер. с англ. Слинкин А.А. [Текст] / А. Джулли, П. Суджит. – М.: ДМК Пресс, 2018. – 294 с.: ил.
2. Шолле, Ф. Глубокое обучение на Python [Текст] / Ф. Шолле. – СПб.: Питер, 2018. – 400 с.: ил.
3. Гафаров, Ф.М. Искусственные нейронные сети и приложения [Текст]: учебное пособие / Ф.М. Гафаров, А.Ф. Галимянов. – Казань: Изд-во Казан. ун-та, 2018. – 121 с.
4. Сергеев, А.П. Введение в нейросетевое моделирование [Текст]: учебное пособие / А.П. Сергеев, Д.А. Тарасов; под общ. ред. А.П. Сергеева – Екатеринбург: Изд-во Урал. ун-та, 2017. – 128 с.
5. Вакуленко С.А. Практический курс по нейронным сетям [Текст]: учебное пособие / А.П. Сергеев, Д. А. Тарасов; под общ. ред. А. П. Сергеева – СПб.: Университет ИТМО, 2018. – 71 с.

УДК 004.42; ГРНТИ 50.05.13

МЕТРИЧЕСКОЕ ПРОСТРАНСТВО ПРОГРАММНОГО КОДА

В.В. Белов, И.Ю. Перехода

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

Аннотация. В работе предлагается группировка метрик программного кода предназначенных для обеспечения качества разрабатываемого программного продукта за счёт анализа и прогнозирования с использованием средств интеллектуального анализа данных.

Ключевые слова: качество, метрики кода, метрическое пространство, интеллектуальный анализ данных.

METRIC CODE SPACE

V.V. Belov, I.Yu. Perekhoda

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

The summary. The paper proposes a grouping of program code metrics in order to ensure the quality of the developed software product by analyzing and predicting using data mining tools.

Keywords: quality, code metrics, metric space, data mining.

Введение

Вопросы оценки качества программного кода возникли практически одновременно с появлением первого опыта написания программ. Однако до сих пор строгое определение качества кода не сформулировано, хотя понятие качества программного обеспечения имеет достаточно тщательное профессиональное обсуждение и достаточно интенсивную практику стандартизации.

19 декабря 1991 года был выпущен стандарт ISO/ IEC 9126:1991, определивший модель качества, классифицирующую качество программного обеспечения по шести категориям, соответствующим разным наборам структурных характеристик, которые детализируются подхарактеристиками.

15 июня 2001 года начался выпуск четырёх взаимосвязанных стандартов:

- ISO/IEC 9126-1:2001. Part 1: Quality model;
 - ISO/IECTR 9126-2:2003. Part 2: External metrics;
 - ISO/IECTR 9126-3:2003. Part 3: Internal metrics;
 - ISO/IECTR 9126-4:2004. Part 4: Quality in use metrics,
- расширяющих ISO/ IEC 9126:1991.

1 марта 2011 года на смену ISO/IEC 9126 выпущен стандарт ISO/IEC 25010:2011 (Системная и программная инженерия. Требования и оценка качества систем и программного обеспечения. Модели качества систем и программного обеспечения). Перечень основных характеристик качества программного обеспечения был дополнен характеристиками «безопасность» и «совместимость». На основе моделей ISO/IEC 25010:2011 консорциум по качеству программного обеспечения (CISQ) определил пять главных целесообразных структурных характеристик, необходимых для того, чтобы программное обеспечение способствовало созданию ценности для бизнеса: 1) Надежность; 2) Эффективность; 3) Безопасность; 4) Ремонтпригодность; 5) (адекватный) Размер [1]. Установлена взаимосвязь (см. рисунок) между указанными структурными характеристиками программного обеспечения и измеримыми атрибутами:

1. Стандарты архитектуры приложений

- Соответствие многоуровневому дизайну (пользовательский интерфейс против дизайна приложения против инфраструктуры / данных)
- Производительность доступа к данным

- Коэффициенты сцепления
 - Коэффициенты повторного использования компонентов (или шаблонов)
2. Методы кодирования
 - Обработка ошибок/исключений (все уровни пользовательского интерфейса / логики/данных)
 - Соблюдение принципов объектно-ориентированного и структурированного программирования (если применимо)
 - Безопасный контроль (доступ к системным функциям, контроль доступа к программам)
 3. Сложность
 - Транзакция
 - Алгоритмы
 - Методы программирования (например, использование полиморфизма, динамическое создание экземпляров)
 - Грязное программирование (мёртвый код, пустой код...)
 4. Документация
 - Удобочитаемость и структурированность кода
 - Соотношение документации на уровне архитектуры, программы и кода
 - Организация файлов исходного кода
 5. Переносимость:

Уровни зависимости аппаратных средств, ОС, программных компонентов и БД
 6. Технические и функциональные объёмы
 - Местоположение по технологии, артефактов, файлов
 - Функциональные моменты – соответствие спецификации IFPUG (организация, объединяющая пользователей метрических программ для анализа функциональных точек по всему миру).

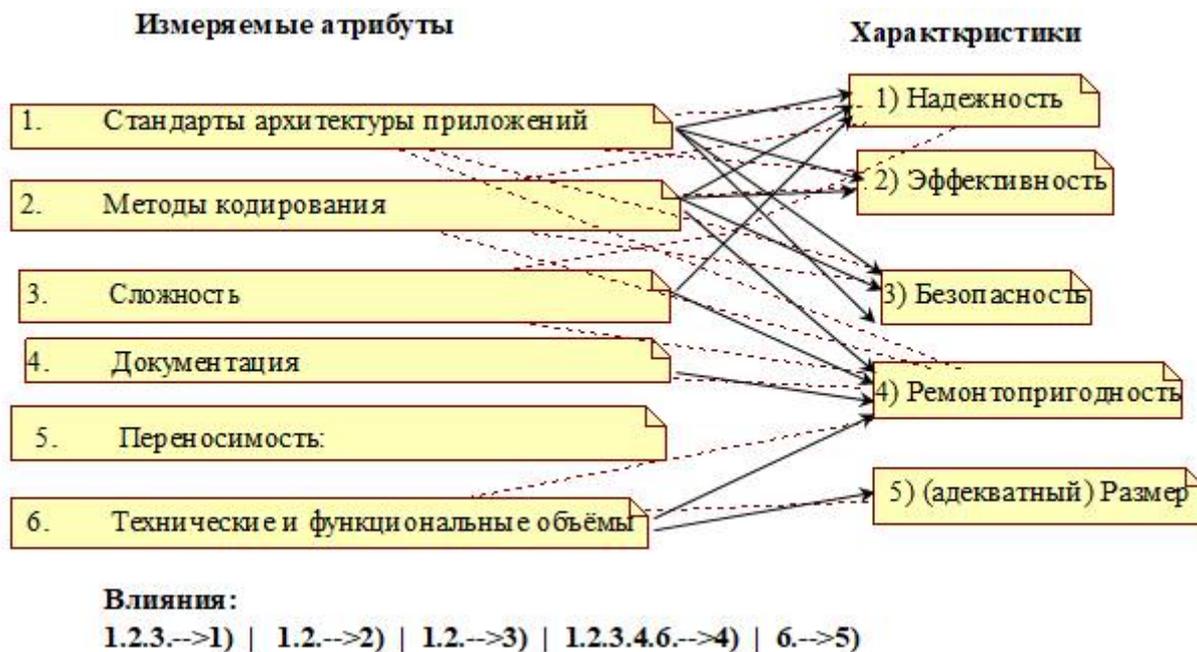


Рис. 1. Взаимосвязь между желаемыми характеристиками программного обеспечения (справа) и измеримыми атрибутами (слева) по CISQ

В 2021 году OMG представила стандарт Международной организации по стандартам,

которая приняла его как ISO/IEC 5055:2021 (Информационная технология. Измерение показателей программного обеспечения. Количественная оценка качества программного обеспечения. Меры качества для автоматического исходного кодирования). Этот стандарт предлагает измерять внутреннюю структуру программного продукта по четырём критически важным для бизнеса факторам: 1) Безопасность, 2) Надёжность, 3) Производительность (PerformanceEfficiency), 4) Ремонтпригодность. (Заметим, что, по сути дела, из перечня CISQ исключена пятая характеристика и характеристики стали называться более корректно – факторами).

В крупных интегрированных средах разработки программного обеспечения (IDE – IntegratedDevelopmentEnvironment) существуют механизмы, обеспечивающие вычисление метрик программного кода, нацеленных на обеспечение высокого качества разрабатываемого программного продукта и минимизацию дефектов в нём. Однако с помощью этих механизмов не удастся в полной мере получить объективную оценку разработанного программного продукта. Выбранные метрики, как правило, являются примитивными измерениями исходного кода и дизайна системы, а возможности для дальнейшей эксплуатации и практическая значимость зачастую вовсе отсутствуют.

Для преодоления указанной проблемы необходимо использовать комплекс метрик и создавать более сложные показатели, открывающие возможность какоценки, так и прогнозирования качества программного обеспечения, включая его важнейшую составляющую – программный код. Целью настоящей работы является поиск компоновки метрик, предназначенных для реализации интеллектуального анализа качества программного кода и выявление дефектов в нём.

Анализ стандартов и публикаций позволяет сделать вывод: качество программного продукта определяется в первую очередь тем, насколько полно и эффективно он решает задачи, поставленные на этапе формирования требований, а во вторую – грамотно выдержанным архитектурным стилем, чётким разделением кода на функциональные блоки и строгой структуризацией[4]. Несмотря на второе место в перечислении очередей, определяющим фактором качественного программного продукта является именно качественный программный код, поскольку именно он решающим образом определяет способность программного продукта решать поставленные задачи. В неформальном определении качественным программным кодом называют код, позволяющий легко поддерживать, расширять и корректировать уже существующую алгоритмическую структуру в нём.

Метрическое пространство программного кода

Совокупность численных показателей, в том числе и вычисляемых, обеспечивающих оценку и контроль качества программного кода, образует метрическое пространство, поскольку для каждой пары показателей вполне определяемо расстояние, называемое метрикой [2]. Обработку контролируемых показателей может осуществляться с использованием средств интеллектуального анализа данных. [3]

Для формирования объективного представления о качестве программного кода можно использовать связанный набор метрик (комплекс), которые будут наиболее полно отражать качественные признаки программного кода. К основным группам, подходящим для этих целей, можно отнести:

- сложность программного кода;
- связность программного кода;
- структурированность программного кода.

После измерения и вычисления метрик каждой группы, необходимо объединить результаты этих показателей и получить единый результат по каждому комплексу метрик, образующих метрическое пространство.

К первому комплексу метрик относится сложность программного кода. Для измерения сложности могут быть использованы метрики Холстеда [2], отражающие трудоёмкость кодирования (1) и понимания (2) разрабатываемого программного обеспечения. Помимо этого, для оценки сложности программного кода хорошо подойдут метрики сложности Мак-Кейба (3), а также базовые метрики, к которым относятся количество строк исходного кода (LOC, linesofcode), количество классов в пространстве имен (CIN, classesinnamespace), количество методов в классе (MOC, methodsofclass).

$$D = 1/L^* \text{ (мера Холстеда – трудоёмкость кодирования),} \quad (1)$$

где L^* -уровень качества программирования, основанный лишь на параметрах реальной программы без учёта теоретических параметров.

$$E_c = V/L^* \text{ (мера Холстеда – трудоёмкость понимания),} \quad (2)$$

где V – объем программы, L^* -уровень качества программирования, основанный лишь на параметрах реальной программы без учёта теоретических параметров.

$$\lambda(G) = m - n + 2 \text{ (цикломатическое число Мак-Кейба),} \quad (3)$$

где m – количество ребер графа, n – количество вершин графа.

Следующий комплекс метрик - связность программного кода, который отражает зависимости компонентов внутри кода.[4] Основными метрики в данном комплексе являются метрики внешних (4) и внутренних (5) зависимостей (efferentcoupling и afferentcoupling) и глубина наследования (6).

$$C_e = \sum C_{in}(C_{out}) \text{ (внешние зависимости),} \quad (4)$$

$$C_a = \sum C_{out}(C_{in}) \text{ (внутренние зависимости),} \quad (5)$$

где C_{in} и C_{out} – внутренние пакеты классов и внешние соответственно.

$$DIT = \sum base(C) \text{ (глубина наследования),} \quad (6)$$

где C – текущий узел дерева классов, а сама формула выражает максимальную длину от узла до корня дерева.

Группа метрик, характеризующих структурированность программного кода. Данная группа отражает степень разбиения программного кода на функциональные блоки и уровень инкапсуляции. Для обеспечения этого контроля используются, как правило, метрики Хансена (7) и Пивоварского (8), которые в свою очередь являются модификацией меры Мак-Кейба.

$$S = \{\lambda(G), N\} \text{ (метрика Хансена),} \quad (7)$$

где $\lambda(G)$ –цикломатическое число Мак-Кейба, N – число операторов в программном коде.

$$N(G) = n^*(G) + \sum P_i \text{ (метрика Пивоварского),} \quad (8)$$

где $n^*(G)$ – модифицированная цикломатическая сложность, P_i – глубина вложенности -й предикатной вершины.

Таким образом, мы получили перечень метрических характеристик по каждому из предложенных комплексов, образующих метрическое пространство программного кода, пригодное для обеспечения качества программного продукта и прогнозирования дефектов в нем с использованием инструментов интеллектуального анализа данных.

Заключение

Проблема обеспечения и контроля качества программного кода на данный момент лежит целиком и полностью на возможности корректного использования и вычисления метрического пространства программного кода. При решении этой проблемы требуется грамотно сгруппировать метрики среди их многообразия. Можно предположить, что выбранные комплексы обеспечат стабильный контроль качества при использовании средств интеллектуального анализа данных и возможности прогнозирования различных свойств программного кода, в том числе качественных характеристик и возможных дефектов в нем.

Библиографический список

1. Software Sizing Standards | CISQ - Consortium for Information & Software Quality (https://translated.turbopages.org/proxy_u/en-ru.ru.1fca8d9a-6234ec19-49d87b47-74722d776562/https/www.it-cisq.org/s tandards/software-sizing-standards/)
2. Кайгородцев, Г.И. Введение в курс метрической теории метрологии программ / Г.И. Кайгородцев. – Новосибирск: НГТУ, 2009 – 187 с.
3. Перехода И.Ю. Обзор методов DATAMINING для анализа параметров программного кода // Материалы VI научно-технической конференции магистрантов Рязанского государственного радиотехнического университета. – Рязань: РГРТУ, 2020 – С. 321 – 322
4. Alpert S. R., Brown K., Woolf B. The Design Patterns Smalltalk Companion. Addison Wesley, 1998.
5. Briand L.C., Daly J., Wust J.A. Unified Framework for Cohesion Measurement in Object-Oriented Systems. Empirical Software Engineering // International Journal. 1998. Vol 3, № 2.

УДК 004.021; ГРНТИ 50.05.03

СПОСОБЫ УСКОРЕНИЯ ВЫЧИСЛЕНИЙ НА ПРИМЕРЕ ПРЕОБРАЗОВАНИЯ УОЛША

Н.И. Ципорков

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

Аннотация. The paper considers ways to speed up processing using the Walsh transform algorithm as an example.

Ключевые слова: parallel algorithm, Walsh transform, matrix multiplication, speedup, efficiency, processing.

METHODS FOR ACCELERATED CALCULATIONS ON THE EXAMPLE OF WALSH TRANSFORMATION

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

Изображение является источником важной информации, так как визуальное представление несет в себе много данных об объектах. Качество изображений постоянно растет, в следствие чего методы обработки требует все больше вычислительных мощностей.

Из-за роста объемов информации, которые несут в себе современные изображения, их обработка становится время затратой задачей. Поэтому для того, чтобы уменьшить время работы алгоритмов, работающих с изображениями, необходимо все чаще прибегать к применению технологий параллельного программирования. Ставится задача анализа существ-

вующих алгоритмов, выявления их узких мест, а также частей задачи, которые можно выполнить параллельно, в следствие чего в разы сократить время выполнения [1].

Каждый год развивается все больше сфер, а также появляются новые, которые используют вычислительные системы. Объем данных и, как следствие, объем вычислений также растет.

Решение задач последовательными алгоритмами занимает все большее время, что очень сильно сказывается на различных процессах в нашей жизни, которые должны осуществлять вычисления быстро, чтобы выдавать результат с максимальной скоростью.

Для того, чтобы преодолеть временной порог, которые устанавливают нам последовательные алгоритмы, необходимо прибегать к различным методам ускорения вычислений:

- Оптимизация алгоритмов. При анализе алгоритмов, иногда становится возможным добиться улучшения, применив различные стратегии оптимизации.
- Распараллеливание на CPU. В настоящий момент большая часть компьютеров оснащено многоядерными процессорами, что дает возможность реализовать параллельный алгоритм, который даст прирост в скорости работы.
- Распараллеливание на GPU. В большинстве компьютеров стоят видеокарты, которые внутри себя имеют множество вычислительных блоков, при помощи которых можно распараллеливать большинство несложных задач, чтобы добиться прироста производительности.

Задача написания параллельного алгоритма достаточно сложна, так как содержит множество нюансов, таких как общая память, координация потоков и т.д. Отсюда можно сделать вывод, что для каждой задачи необходимо анализировать и находить оптимальный способ решения данной задачи, исходя из условий, в которых будет применяться данный алгоритм [2].

Преобразование Уолша

Функции Уолша и основанное на этих функциях преобразование обладает рядом свойств, благодаря которым их применение в системах обработки сигналов часто оказывается более предпочтительным по сравнению с другими. Эти функции получили широкое распространение при обработке речевых сигналов, при обработке изображений в биологии и медицине, в цифровой голографии и многих других областях [14].

Функции Уолша определяются через функции Радемахера. Последние, в свою очередь, образуются из синусоидальных функций с помощью формулы:

$$r_0(\theta) \equiv 1, \quad r_i(\theta) = \text{sign}[\sin(2^i\pi\theta)], \quad i = 1, 2, \dots \quad (1)$$

Здесь $\theta = t/T$, где T – период функций, и $0 \leq \theta < 1$. Символом sign обозначается сигнум-функция:

$$\text{sign} = \begin{cases} 1, & \text{при } x > 0, \\ -1, & \text{при } x < 0. \end{cases}$$

Из (1) следует, что функции Радемахера являются дискретными и принимают только два значения: +1 и -1.

Система функций Радемахера является ортогональной на интервале $[0,1)$, но неполной, так как на том же интервале существуют другие функции, ортогональные им. Система функций Уолша является расширением системы функций Радемахера до полной системы [14].

Рассмотрим в качестве базовой систему функций Уолша, в которой функции упорядочены по числу пересечений ими нулевого уровня. Функции Уолша определяются с помощью функций Радемахера следующим образом:

$$wal_0(\theta) = 1; wal_i(\theta) = \prod_{j=1}^n [r_j(\theta)]^{i_j^\Gamma}, \quad (2)$$

где i_j^Γ – значение j-го разряда в записи номера функций i в коде Грея.

На примере задачи разработки алгоритма, который будет осуществлять преобразование Уолша, рассмотрим способы ускорения времени работы, используя следующие технологии:

- Java Thread API
- CUDA

Главной задачей при ускорении алгоритма для осуществления преобразования Уолша является разработка более быстроедейственных алгоритмов умножения матриц, так как эта самый трудоемкий этап. Также кроме использования технологий распараллеливания, попробуем оптимизировать классический подход к умножению матриц, при помощи использования кэша первого уровня и уменьшения количества циклов вычисления посредством объединения операций транспонирования и умножения в один вычислительный цикл.

С увеличением размера матриц, которые необходимо перемножить, время увеличивается со скоростью n^3 для классического алгоритма. Для анализа ускорения других алгоритмов приведем временные характеристики алгоритмов в таблице 1.

Таблица 1. Временные характеристики алгоритмов матричного умножения

Алгоритм	Время (с)	Ускорение
Изображение 256*256		
Классический	11.807	1
Оптимизированный	1.387	8.5
Java Thread API	1.189	60.23
CUDA	0.196	3 935
Изображение 512*512		
Классический	174.346	1
Оптимизированный	7.268	23.99
Java Thread API	5.159	33.8
CUDA	0.739	235.92
Изображение 1024*1024		
Классический	2044.06	1
Оптимизированный	52.918	38.66
Java Thread API	40.723	50.19
CUDA	3.038	672.83

Так как классический алгоритм показывает достаточно большое время, то необходимо привести также график (рисунок 1), включающий все алгоритмы кроме классического, чтобы можно было сделать вывод о времени выполнения этих алгоритмов относительно друг друга.

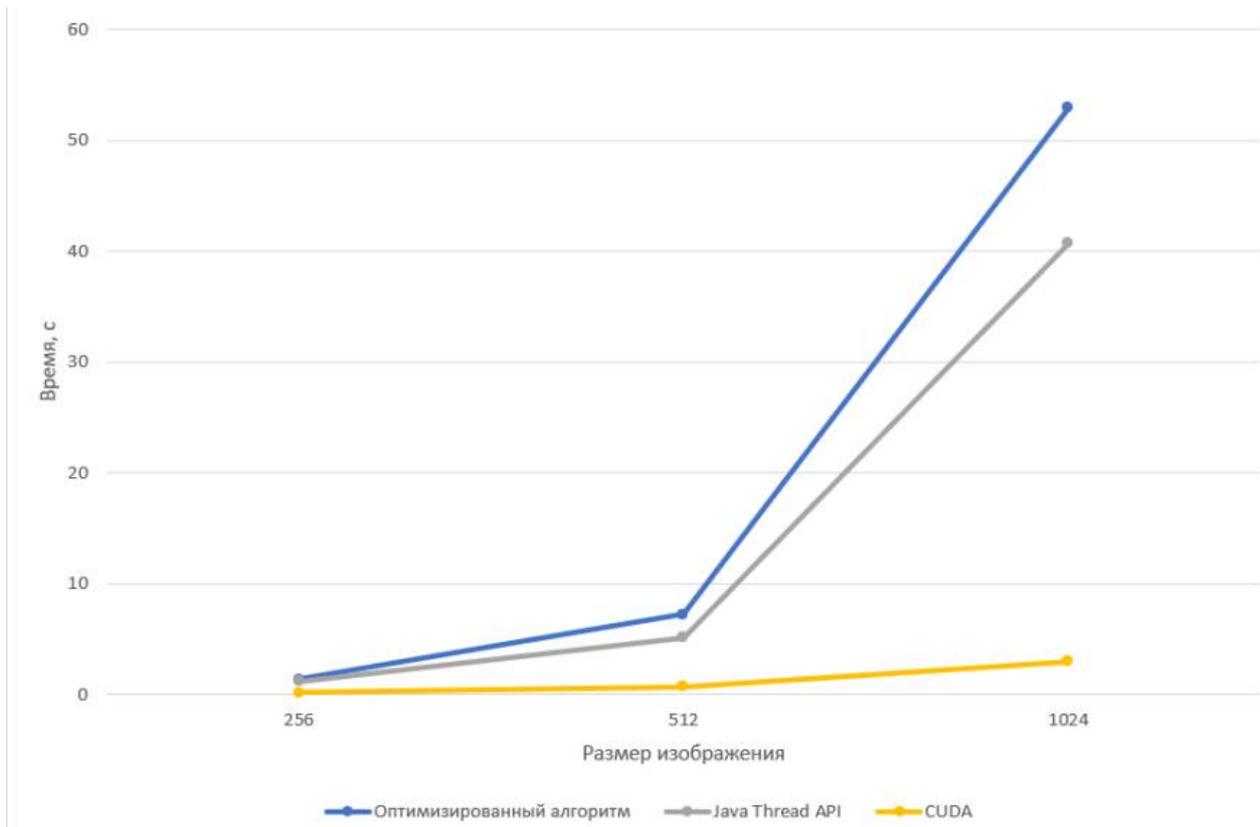


Рис. 1. График времени выполнения алгоритмов матричного умножения

Исходя из полученных данных, самым эффективным является алгоритм, разработанный под технологию CUDA. Из-за технического ограничения, тесты для алгоритма под технологию Java Thread API проводились на процессоре с четырьмя ядрами. Этот алгоритм, в теории мог показать лучшее время, так как внутри себя он вычисляет сколько доступно ядер, чтобы выделить под них такое же число потоков. Для технически “слабых” вычислительных систем самым удачным будет использовать оптимизированный алгоритм, так как он задействует кэш память и выполняет меньшее число вычислительных циклов.

Библиографический список

1. Айдинян А. Р., Аппаратные средства вычислительной техники [Текст]: учебник – М., Берлин: Директ-Медиа, 2016 г. – 125 с
2. Биллиг В. А. Параллельные вычисления и многопоточное программирование [Текст]: 2-е изд. исправленное. – М.: НОУ «ИНТУИТ», 2016. – 311 с.
3. Спектральное разложение сигналов по функциям Лагера и Уолша [Электронный ресурс] <https://portal.tpu.ru/SHARED/v/VOS/study/disc1/Tab/tema04.pdf>.

УДК 005.519.8; ГРНТИ 28.23.37

ИЕРАРХИЧЕСКАЯ КЛАССИФИКАЦИЯ ТЕКСТОВ С ПОМОЩЬЮ НЕЙРОННЫХ СЕТЕЙ

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

Рязанский государственный радиотехнический университет имени В.Ф. Уткина,

Российская Федерация, Рязань, e-mail: ninakorobova77@gmail.com

Аннотация. Вработерассматриваются вопросы иерархической двухуровневой классификации текстов с помощью нейронных сетей.

Ключевые слова: нейронные сети, иерархическая классификация текста, предобработка текстовых данных, рекуррентные нейронные сети, сверточные нейронные сети, язык Python, библиотека Keras .

HIERARCHICAL CLASSIFICATION OF TEXTS USING NEURAL NETWORKS

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

Ryazan State Radio Engineering University named after V.F. Utkin,

Russian Federation, Ryazan, e-mail: ninakorobova77@gmail.com

Abstract. The paper deals with the issues of hierarchical two-level classification of texts using neural networks.

Keywords: neural networks, hierarchical text classification, pre-processing of text data, recurrent neural networks, convolutional neural networks, Python language, Keras library

Тема настоящей работы связана с исследованиями , проводимыми в ведомственной лаборатории РГРТУ по автоматизированному контролю, анализу и оценке эффективности коллективно-договорных актов в сфере образования.

В работе [1] рассмотрена процедура оценки качества коллективного договора в ВУЗе. Для ее реализации коллективный договор разбивается на фрагменты. Каждый фрагмент индексируется с помощью двух индексов. Первый индекс отвечает за принадлежность фрагмента к определенному разделу, например: *Заработная плата* или *Рабочее время*. Всего 11 разделов. Второй индекс отвечает за вопрос, рассматриваемый во фрагменте, например, *Гарантии, Обращения, Жалобы* и т.д. Для каждого раздела свой список вопросов. С учетом обоих индексов фрагменты могут принадлежать 80 разным классам. Процесс разметки фрагментов трудоемкий, требует наличия опытных экспертов, поэтому встает вопрос его автоматизации.

Настоящая работа посвящена исследованию применимости нейронных сетей [2,3, 5] к решению задачи автоматизированной классификации (индексации) фрагментов коллективного договора. Чтобы решить задачу классификации с помощью нейронной сети нужно иметь в наличии размеченную выборку большого размера фрагментов различных договоров. Такая выборка была получена из базы данных коллективных договоров Вузов Росии ведомственной лаборатории РГРТУ по автоматизированному контролю, анализу и оценке эффективности коллективно-договорных актов в сфере образования. Для реализации нейронных сетей были выбраны следующие инструментальные средства: язык Python, библиотеки TensorFlow, Keras, виртуальная среда GoogleColab[2]. Этот выбор обоснован тем, что в настоящее время большинство новых результатов в области нейронных сетей получены и опубликованы на основе перечисленных инструментальных средств.

Существуют следующие виды классификации: плоская, иерархическая и сетевая. При плоской классификации алгоритм (одна нейронная сеть) должен получить правила (процедуру) отнесения фрагмента к одному из 80 классов без учета подчиненности вопроса теме раздела. Классов много и потеряна дополнительная информация о связи раздела с вопросом, поэтому при эксперименте эта методика показала плохой результат: точность менее 60% .

Выбор иерархической классификации применительно к поставленной задаче показывает, что уровней всего два: по разделам и внутри раздела по вопросам, что облегчает задачу. Для реализации двухуровневой классификации была предложена система, состоящая из двух взаимосвязанных классификаторов. Один классификатор (первой степени) должен решать классификацию по разделам, другой в зависимости от полученного первым классификатором результата (раздела) должен выполнить классификацию (второй степени) по вопросам внутри этого раздела.

Т.к. классификация по вопросам зависит от раздела, то для каждого раздела нужен свой классификатор. Следовательно, необходимо иметь столько классификаторов (нейронных сетей) второй степени, сколько разделов.

Классификатор первой степени обучается на всей выборке, классификаторы второй степени обучаются на той части выборки, которая относится только к заданному разделу. И в этом случае число обучающих примеров сокращается, что снижает точность классификации на второй степени.

Если все классификаторы обучены, их модели сохранены в долговременной памяти, то работа системы нейронных сетей происходит следующим образом. В оперативную память загружается классификатор первой степени, на него подается текст фрагмента, на выходе нейронной сети получаем индекс раздела фрагмента. Далее по индексу раздела в память загружается классификатор второй степени. На него подается тот же фрагмент текста и на выходе второго классификатора получаем индекс вопроса.

Каждый классификатор (первой или второй степени) выполнен как ансамбль двух нейронных сетей[4]. В работе рассмотрены две конфигурации нейронной сети: рекуррентная нейронная сеть[2], основанная на ячейках LSTM,

```
model_lstm = Sequential()
model_lstm.add(Embedding(num_words, 32, input_length=max_titel_len))
model_lstm.add(LSTM(25))
model_lstm.add(Dense(n_class_razd, activation='softmax'))
```

и одномерная свёрточная нейронная сеть [2].

```
model = Sequential()
model.add(Embedding(num_words, 32, input_length=max_titel_len))
model.add(Dropout(0.2))
model.add(Conv1D(filters=64,
                 kernel_size=3,
                 activation='relu',
                 bias_initializer='he_normal',
                 padding='same'))
model.add(MaxPooling1D(pool_size=3))
model.add(Conv1D(filters=128,
                 kernel_size=3,
                 activation='relu',
                 bias_initializer='he_normal',
                 padding='same'))
model.add(GlobalAveragePooling1D())
model.add(Dropout(0.2))
model.add(Dense(n_class_razd, activation='softmax'))
```

При их работе в ансамбле возможны разные случаи: обе нейронные сети дают одинаковый результат и он правильный, две нейронные сети дают разные результаты, но один из

них правильный. Решение о выборе правильного результата в этом случае решается по наибольшей величине вероятности полученного ответа выхода активационной функции «soft-max» в последнем слое.

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.

Вся система иерархической двухуровневой классификации состоит из трех модулей: модуль предобработки текстовых фрагментов, модуль обучения классификаторов первой и второй ступеней и модуль эксплуатации системы, который отвечает за синхронную работу классификаторов первой и второй ступеней, а также за оценку качества работы всей системы.

Общая выборка составила 46894 фрагмента, каждый фрагмент – это обучающий пример, т.к. для каждого фрагмента заданы оба индекса: раздел и вопрос. В качестве обучающей выборки при разработке классификатора первой ступени использовались 37514 примеров, для тестовой 9379. Все примеры получены по запросу из базы данных коллективных договоров и преобразованы в формат Excel. Этот формат позволяет в удобной форме вводить данные в среду GoogleColabi использовать их для обучения.

Ниже приводится пример классификации текста фрагмента разработанной системой. При анализе приведенного ниже текста система предсказала результат *ОТ_СУМ*, что означает раздел «Охрана труда», вопрос «Средства на охрану труда». Предсказанное системой значение совпало с истинным значением.

Текст фрагмента

Финансирование мероприятий по улучшению условий и охраны труда осуществляется в размере не менее 0,2% суммы затрат на производство продукции (работ, услуг) .

Предсказанное Истинное значение

ОТ_СУМ ОТ_СУМ

Оценка качества классификаторов каждой ступени проводилась на основе тестовой выборки. Результаты приведены в следующей таблице.

Таблица 1. Оценка качества нейронных сетей

	Классификатор первой ступени	Классификатор второй ступени (среднее значение по классификаторам всех разделов)
Точность на обучающей выборке	0.8753	0.8954
Точность на проверочной выборке	0.8645	0.8534
Точность на тестовой выборке	0.8574	0.8453

Результаты достаточно хорошие и, главное, есть пути для дальнейшего повышения точности классификаторов. Это 1) предобработка: сейчас есть много методов, позволяющих выделять наиболее важные слова для данной предметной области; 2) возможно увеличение обучающей выборки и размера словаря; 3) появились новые методы нейронной классификации текстов. Все эти вопросы требуют дальнейшего исследования.

Библиографический список

1. Александров В.В., Макаров Н.П., Шустов А.С. Автоматизированный анализ и оценка статей коллективных договоров // Вестник Рязанского государственного радиотехнического университета. – 2013. – № 3 (45). – С. 71-75.
2. Цуканова Н.И., Головкин Н.В., Шурьгина О.В. О решении практических задач с использованием глубоких нейронных сетей на базе облачных технологий. //IV Международный научно-технический форум СТНО-2021. Сборник трудов. Том 4. Стр. 14-18
3. Бенгфорт Бенджамин, Билбро Ребекка, Охеда Тони. Прикладной анализ текстовых данных на Python. Машинное обучение и создание приложений обработки естественного языка.— Спб.: Питер, 2019. — 368с.
4. Брюхнова В.О., Цуканова Н.И. Ансамбли нейронных сетей при прогнозировании объемов продаж в торговой сети. //Вестник Рязанского государственного радиотехнического университета. 2018. № 66. Часть 1. С.90-98
5. Цуканова Н.И., Шитова К.Г. Поиск допустимых значений параметров заявки на кредитование с помощью нейронной сети. //Вестник Рязанского государственного радиотехнического университета. 2021. № 78. С.89-101

УДК 004.3; ГРНТИ 50.41

ИСПОЛЬЗОВАНИЕ ТЕХНОЛОГИЙ DOCKER SWARM И ORBITER ДЛЯ РЕАЛИЗАЦИИ АВТОМАСШТАБИРОВАНИЯ СЕРВЕРА

А.О. Бодров

*Московский государственный университет имени М.В.Ломоносова,
Российская Федерация, antonybodrov@mail.ru*

Аннотация. В работе рассматриваются возможности масштабирования сервера в автоматическом режиме с помощью виртуализации сервера на основе технологии контейнеризации docker при использовании балансировщика нагрузки.

Ключевые слова: контейнеризация, масштабирование, автомасштабирование, docker, балансировка нагрузки.

USING DOCKER SWARM AND ORBITER TECHNOLOGIES TO IMPLEMENT SERVER AUTOSCALING

A.O. Bodrov

*Moscow State University named after M.V.Lomonosov,
Russian Federation, antonybodrov@mail.ru*

The summary. The paper discusses the possibilities of scaling the server in automatic mode using server virtualization with docker containerization technology and a load balancer.

Keywords: containerization, scaling, autoscaling, docker, load balancing.

При создании серверной платформы необходимо понимать, что ограничение на число клиентов платформы включает её гибкость в жёсткие рамки. Поэтому для функционирования платформы с различным числом клиентов следует обеспечить её масштабируемость, то есть способность увеличивать производительность системы пропорционально дополнительным ресурсам. Таким образом возможно избавиться от ограничения на число клиентов. Однако выделение ресурсов сверх меры приведёт к тому, что значительная часть выделенных мощностей будет простаивать. В связи с этим помимо простого масштабирования для серверных платформ удачным решением будет масштабирование в автоматическом режиме, когда система будет сама реализовывать свою масштабируемость в зависимости от увеличения или уменьшения числа клиентов.

Архитектура масштабируемого сервера

Для лёгкого копирования сервера удобно использование технологию контейнеров. Контейнеризация – это метод легковесной виртуализации, при котором вместе упаковываются и изолируются от домашней операционной системы (ОС) приложения, бинарные файлы и библиотеки. В отличие от виртуальных машин контейнеры обеспечивают виртуализацию на уровне операционной системы, а не аппаратного обеспечения (Таблица 1).

Таблица 1. Сравнительный анализ виртуальных машин и контейнеров.

Показатели	Виртуальные машины	Контейнеры
Уровень виртуализации	Аппаратный	Операционная система
Операционная система	Отдельная у каждой виртуальной машины	Общая
Время загрузки	Высокое	Низкое
Использование ресурсов	Высокое	Низкое
Предварительные построенные образы	Трудно найти и управлять	Доступны на домашнем сервере
Пользовательские предварительно настроенные образы	Трудно настроить	Легко настроить
Размер занимаемого на жёстком диске места	Больше, потому что виртуализацию происходит с операционной системой полностью	Меньше
Миграция	Легко перемещается с одного сервера на другой	Уничтожается и пересоздаётся вместо перемещения
Время создания	Выше (в среднем минуты)	Ниже (в среднем секунды)
Производительность	Накладные расходы при передаче инструкций из гостевых ОС в хостовую ОС	Производительность, близкая к ОС хоста

Благодаря виртуализации на уровне ОС, то есть изоляции только приложения, бинарных файлов и библиотек без ОС, возможно быстрое развёртывание и простое масштабирование. Также уменьшается объём занимаемого места на жёстком диске по сравнению с виртуализацией с помощью виртуальных машин, как показано на рисунке 1. За счёт легковесности решения при большом количестве копий сервера мы получим существенную экономию занимаемого виртуальными серверами места [1]. Таким образом, сервер предварительно будет упакован в контейнере. Абсолютным лидером в сфере ПО контейнерных технологий является компания Docker, имеющая подробную документацию, удобные утилиты для управления контейнерами и большое сообщество пользователей. Использовать в качестве решения резонно именно docker-контейнеры [2].

Важно правильно распределить нагрузку между копиями сервера. Для этого в каждом контейнере с сервером будет находиться брокер. Балансировщик нагрузки будет устанавливать соединение с брокерами, от которых получит ответ о загруженности каждой из копий сервера [3]. Если узел загружен, то балансировщик передаст запрос следующему брокеру (рисунок 2).

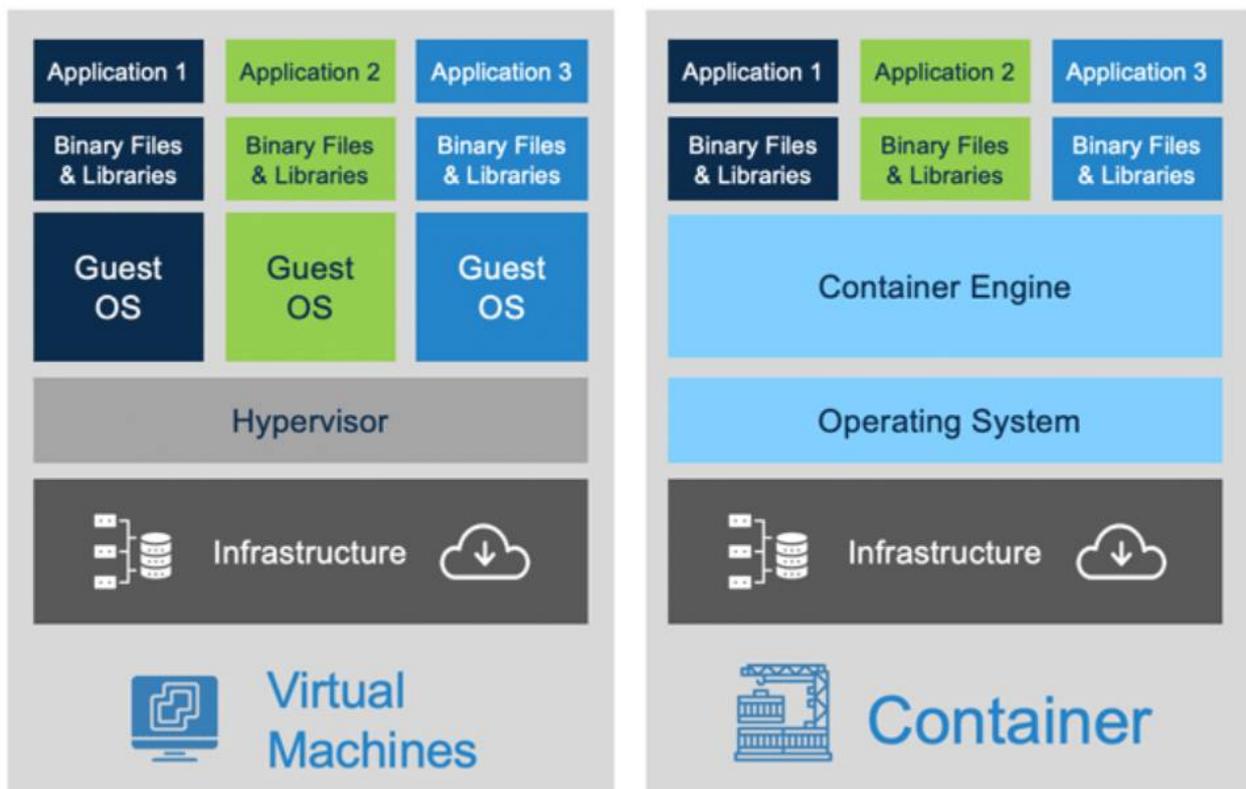


Рис. 1. Сравнение технологий виртуализации виртуальных машин и контейнеров

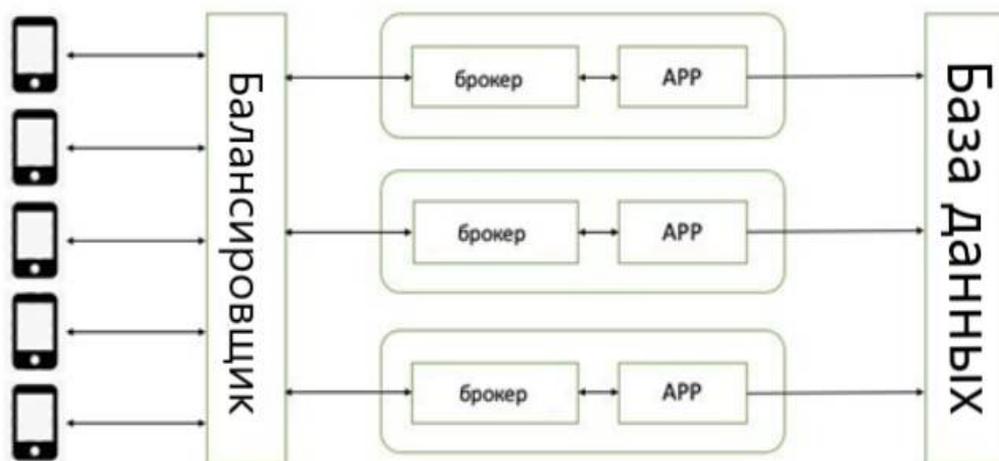


Рис. 2. Архитектура взаимодействия между контейнерами и балансировщиком

Реализация автомасштабирования

Как показано на рисунке 2, заранее создается установленное число контейнеров, из-за чего при маленькой нагрузке часть из них может простаивать, а при большой необходимо добавлять дополнительные копии сервера вручную. На помощь приходит встроенное в docker решение по оркестрации контейнеров dockerswarm. Эта утилита позволяет управлять не только виртуальными серверами, но и физическими. Dockerswarm предполагает настройку таким образом, что часть хостов (виртуальных или физических) является менеджерами и

управляет остальными хостами (нодами). Хост может быть одновременно и менеджером, и нодой, как показано на рисунке 3 [4].

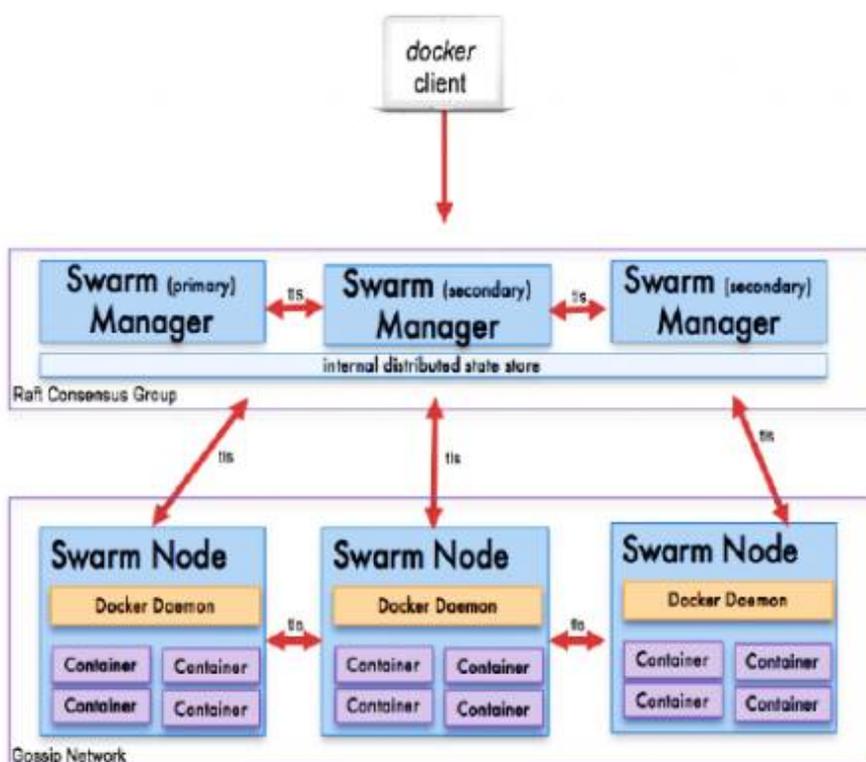


Рис. 3. Схема работы dockerswarm

Таким образом возможна реализация двойной балансировки нагрузки – локальной (среди контейнеров на одном сервере) и глобальной (между разными серверами).

В dockerswarm не встроено автомасштабирование, однако его легко можно добавить с помощью такого программного дополнения для dockerswarm, как Orbiter.

Команда orbiterdaemon включает режим автодетекции. Если сервис помечен меткой orbiter=true, то он начнёт автомасштабирование. Контейнеры для автомасштабирования удобно разметить в стандартном файле docker-compose.yml с информацией о всех разворачиваемых контейнерах итоговой системы. В таком случае будет также удобно запустить сам orbiter в контейнере.

Библиографический список

1. A.M.Potdar, D.G.Narayan, S.Kengond, and M.M.Mulla, "Performance Evaluation of Docker Container and Virtual Machine," *Procedia Comput. Sci.*, vol. 171, no. 2019, pp. 1419–1428, 2020, doi: 10.1016/j.procs.2020.04.152.
2. Docker Documentation. [Электронный ресурс]. - режим доступа: <https://docs.docker.com/>
3. MarischaElveny, Ari Winata, BaihaqiSiregar and RahmadSyah "A tutorial: Load balancers in a container technology system using docker swarms on a single board computer cluster", *EEO*, vol. 19, issue 4, no. 2020, pp.744-751, doi: 10.17051/ilkonline.2020.04.178
4. IT Solution Architects, "Containers 102: Continuing the Journey from OS Virtualization to Workload Virtualization," *medium.com*, 2017. [Online]. Available: <https://medium.com/@ITSolutions/containers-102-continuing-the-journey-from-os-virtualization-to-workload-virtualization-54fe5576969d>. [Accessed: 17-Nov-2021].

УДК 378.14; ГРНТИ 14.35.09

ОЦЕНИВАНИЕ РЕЗУЛЬТАТОВ ОБУЧЕНИЯ ПО ИНДИВИДУАЛЬНЫМ УЧЕБНЫМ ТРАЕКТОРИЯМ

В.В. Белов*, И.Н. Филоненко**

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

** Государственное образовательное учреждение высшего образования Московской области
«Государственный социально-гуманитарный университет».
Российская Федерация, Коломна, filonenkoin@mail.ru

Аннотация. Представлена графовая модель конечного автомата процесса аттестации студента по результатам обучения. Аттестация проводится в виде решения определённого комплекса заданий по изучаемой теме.

Ключевые слова: конечный автомат, модель, оценка студента, набор задач.

ASSESSMENT OF LEARNING OUTCOMES ACCORDING TO INDIVIDUAL LEARNING TRAJECTORIES

V.V. Belov*, I.N. Filonenko,**

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

** State Educational Institution of Higher Education of Moscow Region
«State University of Humanities and Social Studies»,
Russia, Kolomna, filonenkoin@mail.ru

Annotation. The issue related to the evaluation of learning outcomes for individual learning trajectories is discussed. A state machine graph model of the process of appraisal of a student based on learning outcomes is presented. Certification is carried out in the form of solving a certain set of tasks on the studied topic.

Keywords: state machine, model, appraisal of a student, set of tasks.

Введение

К настоящему времени сформировалось достаточно распространённое мнение [1 – 4] о необходимости применения методов раздельного обучения учащихся с разной степенью готовностью к обучению. Различия в степенях готовности к обучению могут иметь разные причины, главные из которых – неизбежные различия индивидуальных интеллектов и способностей учащихся. При этом раздельное обучение может быть как групповым [5], так и персонафицированным. Предельно персонафицированным является обучение по индивидуальным учебным траекториям [7].

В настоящей статье обсуждается вопрос, связанный с оцениванием результатов обучения по индивидуальным учебным траекториям. Представлена графовая автоматная модель процесса аттестации учащегося по результатам обучения. Аттестация проводится в форме решения некоторого набора задач по изученной теме. Важнейшей особенностью реализуемой организации учебного процесса является применение принципа планирования основных фаз учебного процесса *самим учащимся*: изучение теории и выполнение задач осуществляется в периоды времени, когда оказываются выполненными необходимые для обучения условия как пространственно-временные, так и организационно-когнитивные. Пространственно-временные условия выражают наличие времени и удобного места для обучения. Организационно-когнитивные условия выражают тот факт, что учащийся имеет необходимое техническое обеспечение (компьютеры, доступ к Интернету и прочее), а также убеждение, желание и намерение осваивать учебный материал.

Предоставление учащемуся возможности самостоятельно выбирать место и время изучения теории и решения задач, включая возможность останова учебного процесса и его возобновления по собственному усмотрению, обусловлено стремлением обеспечить наилучшее соответствие условий обучения учебному процессу.

Основные информационные элементы модели аттестации

В процессе аттестации используются: модель знаний, подлежащих освоению; модель учащегося; оперативные переменные. Мета-модель знаний, подлежащих освоению, имеет образ (View) оглавления изучаемого учебника, совмещённого с задачником. Соответствующая модель представляет собой граф с элементами содержимого учебника в узлах, дополненный механизмом навигации по содержимому. Содержимое хранится в базе данных «Модель знаний». Элементы навигации реализуются самостоятельным сервером. Данные, ассоциированные с учащимся, хранятся в базе данных «Модель учащегося» с доступом посредством сервера. Важен следующий факт: для разных изучаемых тем учебный процесс, в том числе и аттестации учащегося по результатам обучения, осуществляется в соответствии с одними и теми же автоматными моделями, но разными механизмами реализации состояний автоматов. Оперативные переменные предназначены для хранения информации непосредственно в потоке управления, реализующем модель. Основной оперативной переменной является объект status класса Status.

Поведение автоматов модели процесса аттестации

В моменты останова учебного процесса осуществляется запоминание текущего статуса обучаемого – набора сведений, определяющих местоположение учащегося в пространстве модели изучаемого учебного материала, включающих, в частности, NumTheme – номер текущей изучаемой темы и Counter – номер решаемой задачи в изучаемой теме. При возобновлении учебного процесса осуществляется восстановление (актуализация) запомненного статуса, и учебный процесс продолжается с того места, на котором он был остановлен.

Указанный принцип организации учебного процесса иллюстрируется предлагаемой графовой автоматной моделью процесса контроля результата изучения конкретной темы, представленной на рисунке 1 в виде последовательного (не ортогонального) составного состояния «Контроль темы». Контроль осуществляется в виде процесса решения NTask задач по изучаемой теме NumTheme.

Состояние «Настройка контроля» вложенного автомата используется для актуализации статуса учащегося и для формирования задания для начала или продолжения проверки знаний, т. е. для решения задач изучаемой темы. Актуализация статуса означает извлечение некоторых атрибутов динамической модели обучаемого из базы данных «Модель учащегося» в оперативную память – в атрибуты объекта status класса-сущности Status. Формирование задания сводится к загрузке в оперативную память необходимого фрагмента модели знаний, подлежащих освоению, а именно набора задач темы NumTheme с номерами от Origo до NTask. При первом вхождении в состояние «Контроль темы» значение Origo равно единице, поскольку задачи всех тем нумеруются, начиная с единицы. Уникальный номер задачи является составным NumTheme.Counter, т. е. состоит из номера темы и номера задачи внутри темы.

В состоянии «Инициализация счётчика задач» осуществляется присваивание атрибуту Counter объекта status начального значения, извлечённого из базы данных «Модель учащегося» в атрибут Origo. В процессе работы вложенного автомата значение атрибута Origo помещается в базу данных «Модель учащегося» в составе данных статуса обучаемого во время выполнения деятельности «Фиксация текущего статуса».

Ссылочное состояние «Шаг контроля» ссылается на составное последовательное состояние «Решение задачи № Counter». Диаграмма работы автомата, вложенного в это состояние, представлена на рисунке 2.

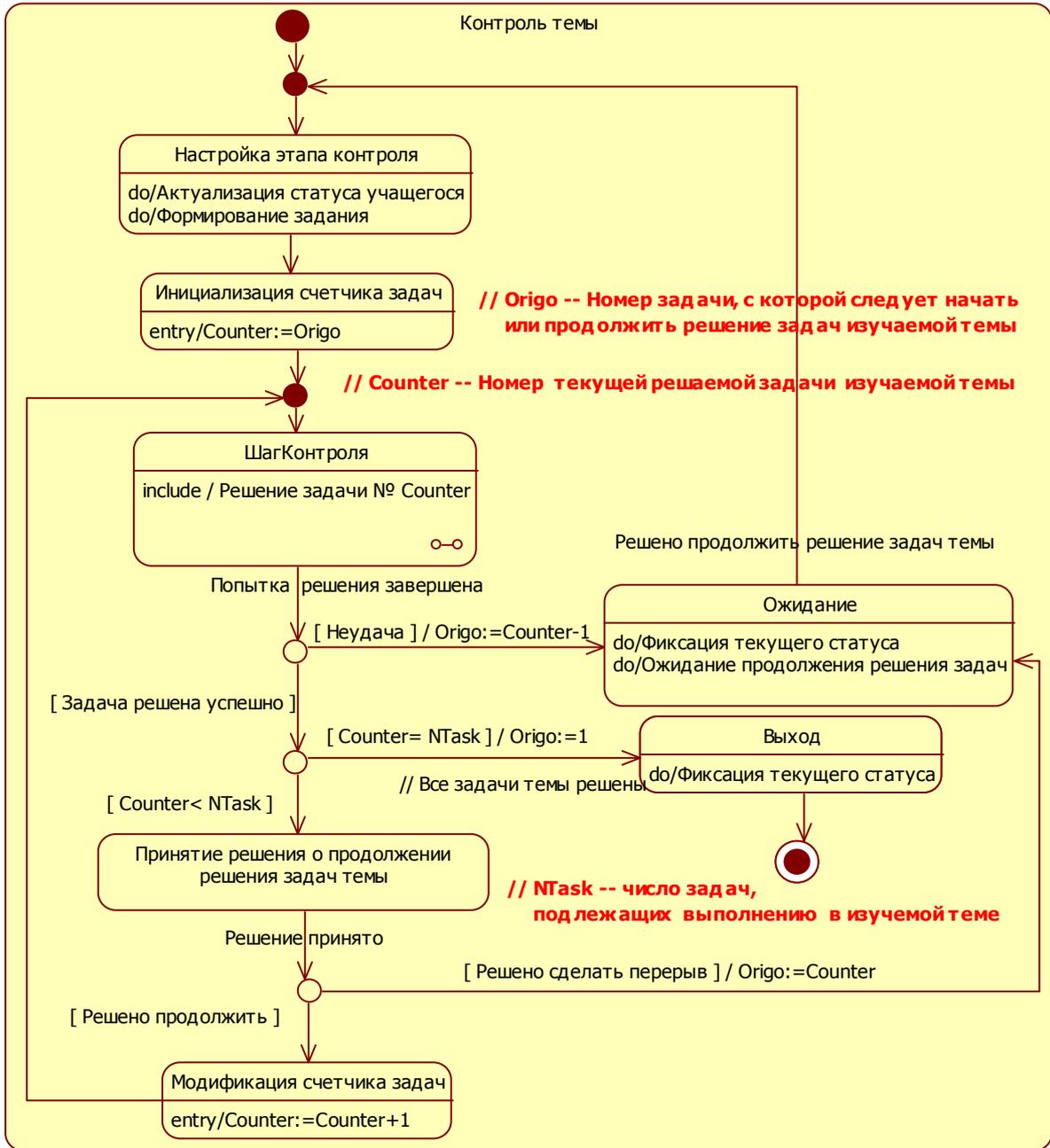


Рис. 1. Автоматная модель процесса контроля результата изучения конкретной темы

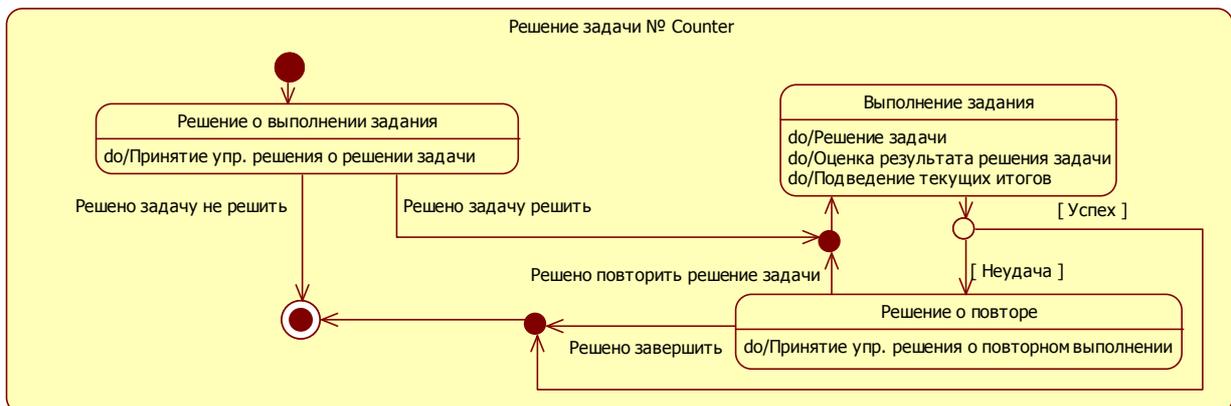


Рис. 2. Автоматная модель процесса решения одной задачи изучаемой темы

Этот автомат моделирует попытку решения учащимся поставленной задачи с номером Counter. Попытка решения задачи может завершиться либо успехом, либо ошибкой в решении с отказом от повторения попыток решения, либо отказом от решения без попыток решить задачу. Последние два варианта выражаются одним условием – «Неудача». Очевидно, что это условие выполняется, если учащийся отказывается от попыток решения задачи либо сразу, либо после неудачных решений. Этот отказ означает, что учащийся принял решение сделать перерыв в процессе аттестации по актуальной теме, поскольку переход к следующей задаче возможен только после успешного завершения решения текущей задачи. По этой причине в случае выхода из состояния «Решение задачи № Counter» с условием «Неудача» осуществляется переход в состояние «Ожидание». В том состоянии, прежде всего, деятельностью «Фиксация текущего статуса» осуществляется запись текущего статуса учащегося в базу данных, а затем осуществляется переход в ожидание принятия учащимся решения о необходимости продолжить процесс аттестации – решения задач изучаемой темы.

Заметим, что в представленных диаграммах в соответствии со стандартом UML простая надпись над стрелкой символизирует триггерное событие (Trigger Event), надпись в квадратных скобках представляет условие перехода (Guard Condition), а надпись с предшествующим слэшем (символом /) выражает действие на переходе (<Behavior-expression). Надписи в рабочем поле диаграммы с двумя предшествующими слэшами – комментарии.

Ожидание наступления благоприятных условий для продолжения учебного процесса, оно может иметь неопределённую длительность. Выход из него осуществляется триггерным событием «Решено продолжить решение задач темы». Заметим, что перед входом в состояние «Ожидание» атрибут Origo, с которым начнётся последующее решение задач, получает значение на единицу меньше значения атрибута Counter, поскольку задача с номером Counter оказывается не решённой.

В случае выхода из состояния «Решение задачи № Counter» с условием «Задача решена успешно» выясняется вопрос о завершённости процесса решения задач текущей темы – проверяется дополнительное условие «Counter = NTask», или вербально: «Равняется ли значение счётчика задач числу отчётных задач в текущей теме». Если это условие оказывается выполненным, то атрибут Origo, с которым начнётся решение задач следующей темы, принимает концептуальное значение 1 и далее осуществляется выход из процесса контроля через состояние «Выход», в котором выполняется деятельность фиксации текущего статуса обучаемого, т. е. занесение атрибутов объекта status в базу данных «Модель учащегося».

Если же условие «Counter = NTask» оказывается не выполненным, т. е. выясняется, что справедлива альтернатива «Counter < NTask», то учащемуся поручается принять решение – продолжить ли решение задач или сделать перерыв. Заметим, что условия «Counter = NTask» и «Counter < NTask» образуют полную группу, поскольку условие «Counter > NTask» не может наступить физически, ибо при Counter = NTask процесс инкрементации Counter останавливается. Если учащийся принимает решение сделать перерыв, то атрибут Origo, с которым продолжится последующее решение задач текущей темы принимает значение счётчика Counter, поскольку последняя задача с номером Counter была решена успешно. Если учащийся принимает решение продолжить, то осуществляется инкрементация Counter и осуществляется переход к состоянию «Решение задачи № Counter» через ссылочное состояние «Шаг контроля».

На рисунке 2 показана автоматная модель процесса решения одной задачи изучаемой темы, представляющая собой составное последовательное (с одним вложенным автоматом) состояние с именем «Решение задачи № Counter». Два состояния вложенного автомата «Решение о выполнении задания» и «Решение о повторе» выполняют управленческую функцию – при нахождении системы в этих состояниях учащийся принимает управленческие решения по поводу выполнения задачи, т. е. отвечает на вопрос – будет ли он решать задачу или нет? Деятельности технического состояния «Выполнение задания» реализуется с участи-

ем учащегося (он решает задачу) и с участием обучающей системы – она оценивает результаты решения и подводит итоги. При этом для разных изучаемых тем могут использоваться различные механизмы реализации системных функций. И это не только разные программные модули, иногда могут использоваться и работники – workers – люди, выполняющие некоторые функции в обучающей системе, работающие «внутри» системы, выполняющие некоторые из её функций.

Если для реализации обучения, в том числе и для контроля, требуются работники, то в состоянии «Настройка этапов контроля» при формировании задания учащемуся сообщаются ограничения по времени выполнения заданий.

Заключение

В статье представлена графовая автоматная модель аттестации учащегося, обучаемого по индивидуальной учебной траектории, отличительной особенностью которой является применение принципа планирования основных фаз учебного процесса самим учащимся. Представленная модель может быть реализована как один из процессов в рамках классической обучающей программно-информационной системы, так в форме отдельной BPMS-системы, реализуемой с использованием BPЕL. Второй вариант имеет предпочтение, если обучение и аттестация осуществляется с использованием работников (workers) системы, т. е. в случае, когда обучающая система является человеко-машинной с существенной долей человеческого фактора. При этом в машинной составляющей обучающей системы особое значение приобретает компонент выстраивания (планирования) бизнес-процесса взаимодействия учащихся с исполнителями ролей консультант/преподаватель/экзаменатор. Естественно, в такой системе свобода выбора времени обучения учащимся существенно уменьшается.

Библиографический список

1. Гормин А.С. Психологические основы барьерной педагогики. Теория развивающего образования для одарённых подростков. – Великий Новгород, НГУ имени Ярослава Мудрого, 2003 – (18 п.л.)
2. Лейтес Н. С. Умственные способности и возраст. М.: Педагогика, 1971. – 277 с.
3. Новые педагогические и информационные технологии в системе образования: учеб. пособие для студ. высш. учеб. заведений / Е.С.Полат, М.Ю. Бухаркина, М.В.Моисеева, А. Е. Петров; под ред. Е. С. Полат. – 3-е изд., испр. и доп. – М.: Издательский центр «Академия», 2008. – 272 с.
4. Шадриков В.Д. Способности и интеллект человека. М.: Издательство современного государственного университета, 2004 – 188 с.
5. Белов В.В., Филоненко И.Н. К вопросу управления процессом обучения в условиях недоосвоения пререквизитов учебных программ // Современные технологии в науке и образовании – СТНО-2020 [текст]: 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. 167 – 172. [04.03.2020-06.03.2020 г. Рязань, Рязанский государственный радиотехнический университет имени В.Ф. Уткина]
6. Кондратьев С.В. Персонафицированное обучение как условие и способ нравственного и интеллектуального развития личности (переход от постиндустриального к гуманитарному обществу) // Евразийское научное объединение. – 2017. – Том 2, – № 9 (31). – С. 129–132.
7. Вдовина С,А., Кунгурова И.М. Сущность и направления реализации индивидуальной образовательной траектории // Интернет-журнал «НАУКОВЕДЕНИЕ». – 2013 – Вып. 6. – 8 с. – <https://naukovedenie.ru/PDF/40PVN613.pdf>

УДК 378.14; ГРНТИ 14.35.09

СЕМАНТИЧЕСКАЯ МОДЕЛЬ РАБОЧЕЙ ПРОГРАММЫ УЧЕБНОЙ ДИСЦИПЛИНЫ

Е.В. Бобылева, И.А. Буланова, А.Н. Пылькин

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

Аннотация. В работе рассматриваются общие принципы формализации предметной области и построение семантической модели.

Ключевые слова: формализация предметной области, семантическая модель, семантическая сеть, графы.

SEMANTIC MODEL OF THE WORKING PROGRAM OF THE EDUCATIONAL DISCIPLINE

E.V. Bobyleva, I.A. Bulanova, A.N. Pylkin

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

The summary. The paper discusses the general principles of formalization of the subject area and the construction of a semantic model.

Keywords: formalization of the subject area, semantic model, semantic network, graphs.

Семантическая модель позволяет вычленять из информационного потока смысловые единицы и связи между ними. Семантическое моделирование в настоящее время применяется к различным областям знаний. Семантическая модель строится в виде семантической сети, которая представляет собой иерархический ориентированный граф.

Отличительной особенностью семантической модели является глубокая структуризация изучаемых понятий предметной области и их представление в виде иерархической модели.

В основе семантической модели лежит логическая структура учебного материала и формализованные знания в виде понятий. [1]

Описание предметной области

Для построения примера семантической модели была выбрана рабочая программа, разработанная на основе ФГОС начального общего образования. В качестве источника данных использовался учебник математики для 1-4 классов авторства М. И. Моро и др.. Учебник входит в федеральный перечень учебников на 2021-2022 учебный год. [2]

Модель состоит из элементов-вершин графа, связанных между собой отношениями-дугами. [3] Элементами могут являться дисциплины, темы, понятия, компетенции и занятия:

- дисциплина – группа, которая содержит список тем (пример дисциплины: 1 класс);
- тема – группа, которая объединяет понятия, компетенции и занятия (пример темы: геометрические величины и их измерение);
- понятие – термин, который входит в тему (пример понятия: треугольник);
- компетенция – навык (умение), который должен получить ученик после прохождения темы (пример компетенции: иметь начальное представление о базовых межпредметных понятиях: числе, величине, геометрической фигуре);
- занятие – форма организации обучения с целью овладения учащимися изучаемыми понятиями.

Отношения могут быть представлены исходящими и входящими связями. Входящие связи могут отражать такие процессы, как понятие «используется при изучении», компетен-

ция «применяется при изучении», а исходящие связи - понятие «изучено», компетенция «приобретена».

Пример обобщенной схемы семантической модели для учебной программы математики 1-4 класса представлена на рис. 1.

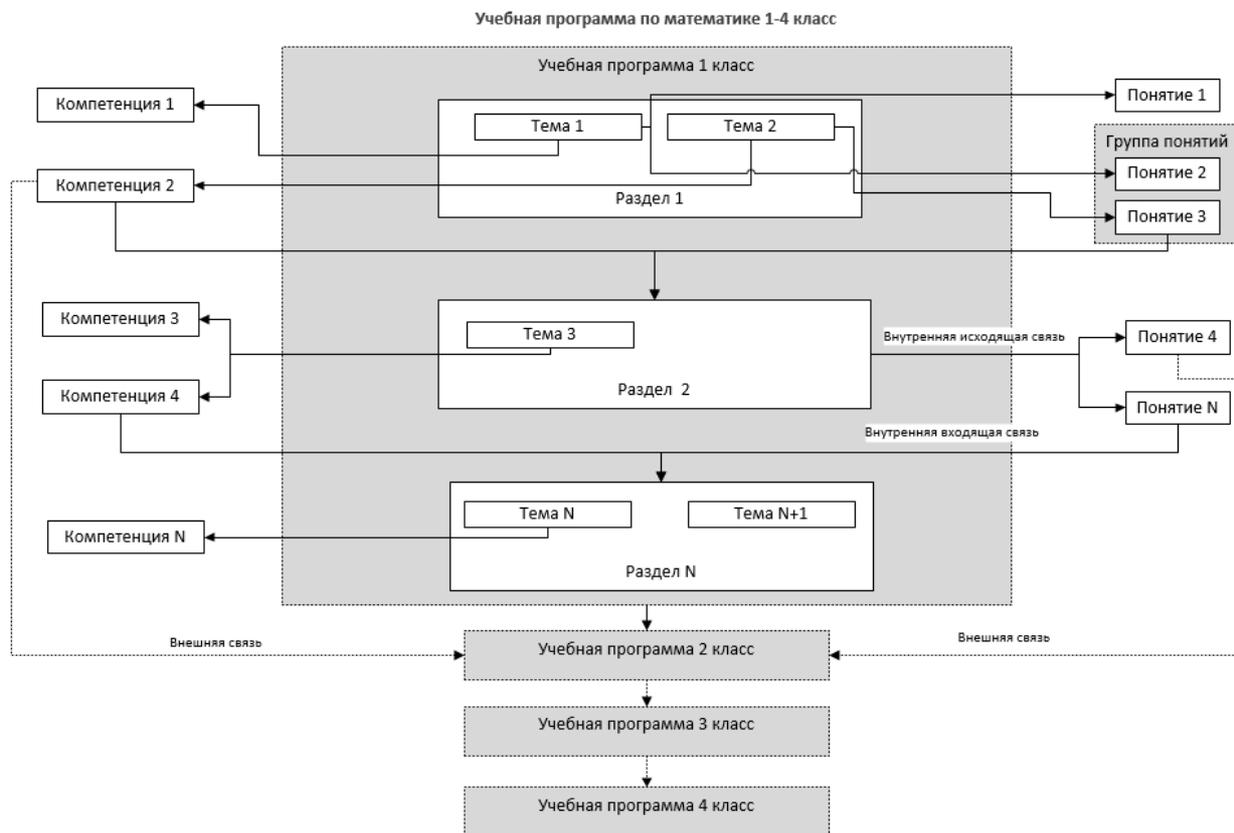


Рис. 1. Обобщенная схема семантической модели

Описание программного комплекса

На рис. 2 представлена общая схема программного комплекса «Семантическая модель рабочей программы учебной дисциплины».

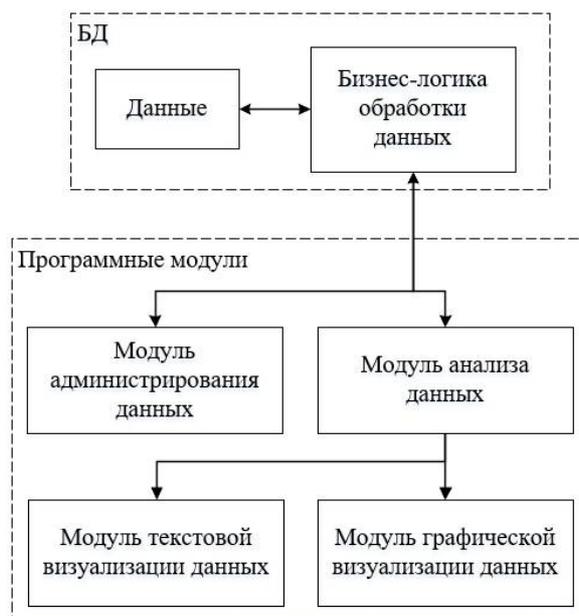


Рис. 2. Общая схема программного комплекса «Семантическая модель рабочей программы учебной дисциплины»

Бизнес-логика базы данных включает в себя следующие функции:

- проверки целостности данных;
- проверки корректности данных;
- преобразования данных (Parsing Data);
- формирования выборок данных (Views).

Взаимодействие комплекса программ с базой данных осуществляется через бизнес-логику базы данных.

Программный комплекс содержит следующие модули:

1. Модуль администрирования данных предоставляет пользовательский интерфейс для управления данными (добавление, изменение, удаление данных).
2. Модуль анализа данных реализует алгоритмы анализа данных в рамках предметной области.
3. Модули визуализации данных.
 - 3.1 Модуль текстовой визуализации данных предоставляет пользовательский интерфейс отображения результатов анализа данных в табличном виде.
 - 3.2 Модуль графической визуализации данных предоставляет пользовательский интерфейс отображения исходных данных или результатов анализа данных в виде графа.

Реализация программных модулей

Модули визуализации данных реализованы в составе программного комплекса в виде desktop-приложения. Скриншот модуля текстовой визуализации представлен на рис.3, фрагмент модуля графической визуализации представлен на рисунке 4.

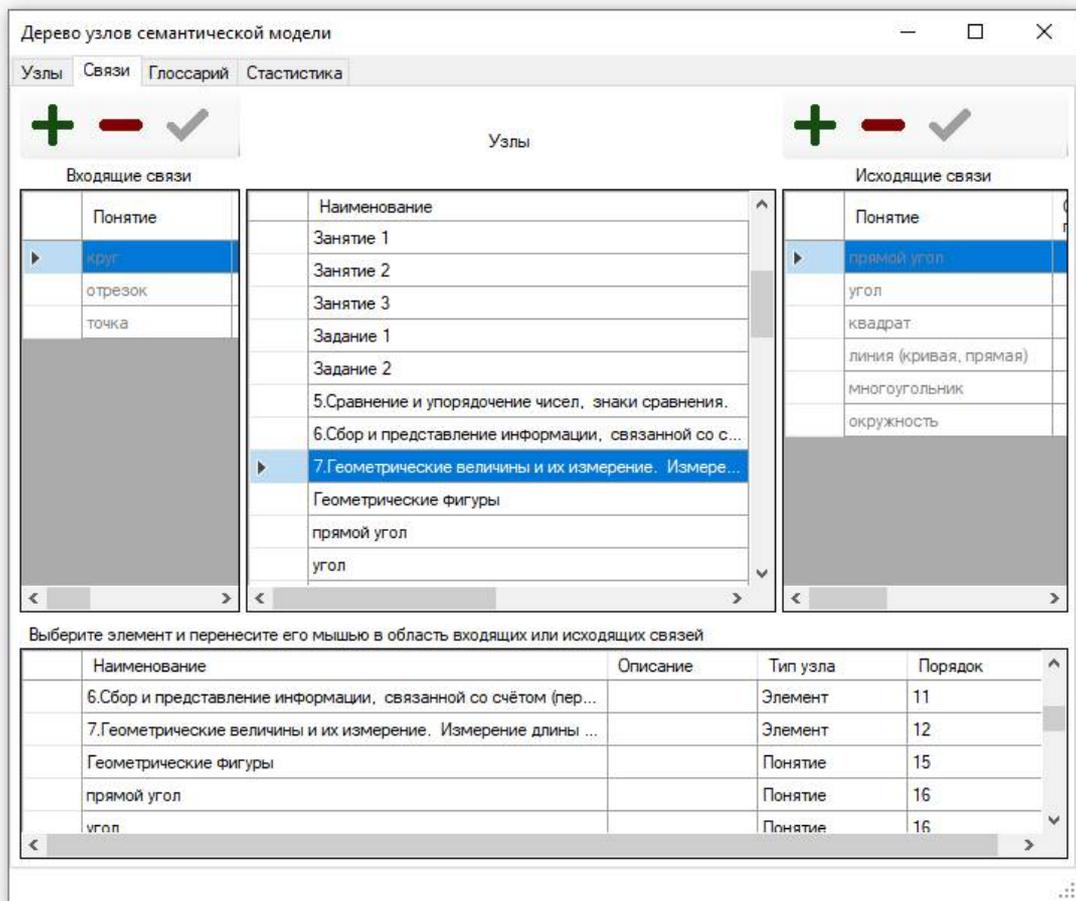


Рис. 3. Скриншот desktop-приложения

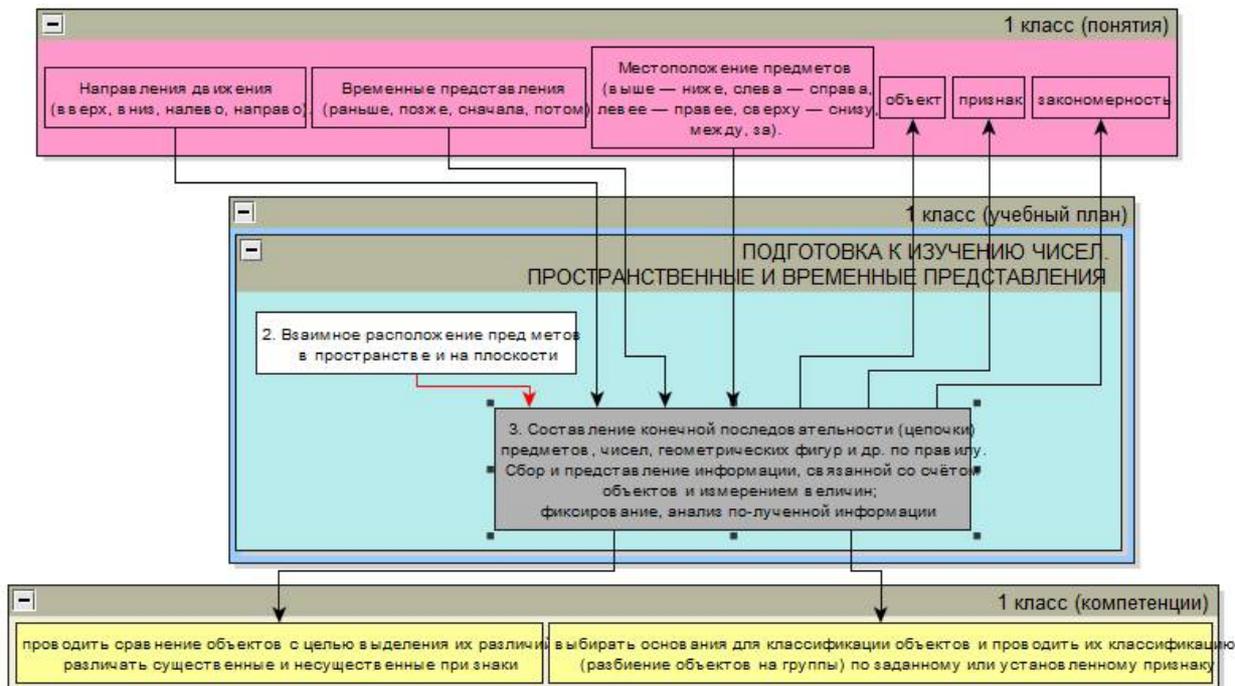


Рис. 4. Фрагмент семантической модели в виде графа

Таким образом, разработанный программный комплекс дает возможность построить семантическую модель рабочей программы учебной дисциплины, которая позволяет проек-

тировать образовательный курс, разрабатывать план учебной программы, определять формат занятий, создавать средства контроля результатов обучения. Интерфейс программного комплекса отвечает современным требованиям эргономики и юзабилити.

Технические требования

Программный комплекс функционирует на персональных ЭВМ в операционных системах Windows XP или выше. Для работы с данными требуется соединение с предустановленным MS SQL Server 2012 или выше.

Применение

Программный комплекс предназначен для эксплуатации в учебных учреждениях специалистами, организующими образовательный процесс.

Библиографический список

1. Шихнабиева Т. Ш., Булаева Н. М., Гусейнова Н. О. Об использовании адаптивных семантических моделей в экологическом образовании // Юг России: экология, развитие. 2008. №3. URL: <https://cyberleninka.ru/article/n/ob-ispolzovanii-adaptivnyh-semanticheskikh-modeley-v-ekologicheskom-obrazovanii> (дата обращения: 03.05.2022).

2. fpu.edu.ru: Федеральный перечень учебников: сайт. – 2021. – URL: <https://fpu.edu.ru/> (дата обращения: 03.05.2022).

3. М. И. Моро и др. Математика. Примерные рабочие программы. Предметная линия учебников системы «Школа России». 1-4 классы : учеб. пособие для общеобразоват. организаций. 5-е изд., перераб. – М. : Просвещение, 2021. – 144 с.

УДК 004.51; ГРНТИ 50.41.29

РАЗРАБОТКА ПОЛЬЗОВАТЕЛЬСКОГО ИНТЕРФЕЙСА ПО «ОНЛАЙН АУКЦИОН»

М.Р. Васильев*, К.И. Кузнецова, М.В. Шилкина*****

*Рязанский государственный радиотехнический университет имени В.Ф. Уткина,
Российская Федерация, Рязань,*

enj1212@yandex.ru, **kkrtstina@mail.ru, *shilkinamarina1210@gmail.com*

Аннотация. В работе рассматривается реализация пользовательского интерфейса системы и функциональных блоков ПО «Онлайн аукцион».

Ключевые слова: онлайн-аукцион (ОА), пользовательский интерфейс (ПИ), регистрация (Р), авторизация (А)

DEVELOPMENT OF THE USER INTERFACE OF THE SOFTWARE "ONLINE AUCTION"

M.R. Vasiliev*, K.I. Kuznetsova, M.V. Shilkina*****

*Ryazan State Radio Engineering University named after V.F. Utkin,
Russian Federation, Ryazan,*

enj1212@yandex.ru, **kkrtstina@mail.ru, *shilkinamarina1210@gmail.com*

The summary. The paper considers the implementation of the user interface of the system and functional blocks of the "Online Auction" software.

Keywords: online auction (OA), user interface (UI), registration (R), authorization (A)

Пользовательский интерфейс онлайн аукциона должен объединять решения как единое целое для поставщиков и покупателей на базе централизованного портала.

Программный комплекс предназначен для выполнения следующих задач: продавец может создавать лоты на продажу товара. Покупатель может делать ставки и участвовать в аукционе на покупку товара.

Система состоит из следующих функциональных блоков:

1. Регистрация, аутентификация и авторизация
2. Функционал для продавца
3. Функционал для покупателя
4. Функционал для оплаты ставки, оплаты товара и т.д.
5. Уведомления о закрытии лотов, открытии лотов продавцом и т.д.
6. Система штрафов и рейтинга

Программа «Онлайн аукцион» должна обеспечивать доступность, удобство.

Для повышения эффективности работы торговой площадки, пользователи должны получать равный и простой доступ к информации и услугам. Поэтому Web-интерфейс должен быть интуитивно-понятный с легкой навигацией по сайту.

Процесс регистрации и персонализации (аутентификации), поиск по аукционам, заключение сделок, процесс заключения договоров должен быть понятен пользователю и не вызывать затруднений.

1. Регистрация, аутентификация и авторизация

Регистрация пользователя в системе производится через ссылку Вход/Регистрация (рисунок 1).

Рис. 1. Регистрация пользователя в системе

Персонализации (аутентификации) реализована в «Личном кабинете». Вход в личный кабинет осуществляется после успешного прохождения процедуры авторизации. Для авторизации служит форма «Вход в личный кабинет» (рис 2).

Рис. 2. Форма. Вход в личный кабинет

2. Разработка дизайна основной страницы

При разработке дизайна основной страницы следует учесть, что пользователь, вероятнее всего, выберет сайт, на котором функционал, который полностью ему понятен. Стоит так же учесть, что на странице не должно быть лишней информации, а содержаться лишь целевая информация по выбранному направлению. Дизайн должен быть выполнен в сдержанных тонах, отображать актуальную информацию.

3. Разработка дизайна страницы лота

3.1 Функционал для продавца и покупателя

Пользовательский интерфейс продавца и покупателя содержит доступ к торговым площадкам (рисунок 3), списку товаров и аукционов (рисунок 4), сообщения о результатах торгов (рисунок 5).

Рис. 3. Торговые площадки

Объявление о проведении торгов

Должник

Наименование: «Агрохимбанк» (Открытое Акционерное Общество)
 Адрес: 119992, г. Москва ГСП-2, Большой Знаменский переулок, д.2 стр. 3
 ОГРН: 1027739282317
 ИНН: 7704044896
 Номер дела: А40-2345/16

Публикатор

Организатор торгов: Торгов Организатор Физлицович

Сообщение

[Скопировать данные из Объявления о проведении торгов.](#)

Вид торгов: повторные торги

Дата и время подачи заявок: начало окончание

Правила подачи заявок:

Дата и время торгов: Форма подачи предложения о цене:

Место проведения: Электронная площадка Другое

Доп. информация:

Список лотов

Номер лота	Описание	Начальная цена, руб.	Шаг	Задаток	Классификация имущества
+ ДОБАВИТЬ ЛОТ					

Рис. 4. Объявление о проведении торгов

Тип сообщения:

Торги на ЭТП

Торговая площадка	Номер торгов	Сообщение ЭТП	Дата публикации сообщения на ЭТП	
«ТЕНДЕР ГАРАНТ»	Trade3	Результаты торгов	19.10.2017 15:49	<input type="checkbox"/> Посмотреть

Рис. 5. Сообщение о результатах торгов

3.2 Функционал для оплаты ставки, оплаты товара

По окончании торгов покупатель получает сведения о результатах торгов в форме «Объявление о проведении торгов» (рисунок 6), которое содержит информацию о номере и дате заключения договора на поставку, перечне товаров/услуг, цене покупки.

Рис. 6. Объявление о результатах торгов

В случае, если покупателя устраивают параметры сделки формируется счет на оплату (рисунок 7). Счет имеет поля: ФИО и реквизиты плательщика.

Рис. 7. Счет на оплату

4. Уведомление о закрытии/открытии лотов продавцом

В перечне аукционов указывается статус аукциона Открыт/Закрыт. Пользователям на указанный ресурс приходят сообщения об открытых/ закрытых аукционах.

5. Система штрафов и рейтинга

В ПО действует система рейтинга продавец/ покупатель. Показатели рейтинга устанавливаются администратором. Действует система штрафов, вплоть до прекращения доступа и отмены регистрации в системе. Система штрафов оговаривается при подписании соглашения на пользование системой.

Библиографический список

1. Байков В.Д., Байков Д.В. Современный самоучитель работы на аукционе. — ДМК-Пресс, 2009. — С. 192. — ISBN 978-5-94074-487-0
2. Юрасов А.В. Основы электронной коммерции = Учебник для вузов. — М.: Горячая линия (издательство). Телеком, 2008. — С. 480. isbn = 978-5-9912-0013-
3. <https://www.bibliofond.ru/view.aspx?id=600160>
4. <http://tendersoft.ru/>
5. <https://a2is.ru/catalog/programmy-dlya-auksionov>
6. <https://web-automation.ru/development/razrabotka-sajta-aukciona/>
7. <https://www.klerk.ru/buh/articles/519153/>
8. https://www.myuniversity.ru/Маркетинг/Интернет_аукцион/174704_2284343_страница2.html

УДК 004.51; ГРНТИ 50.41.29

МОБИЛЬНОЕ ПРИЛОЖЕНИЕ ДЛЯ РАЗМЕЩЕНИЯ ОБЪЯВЛЕНИЙ О ТОВАРАХ СТУДЕНТАМИ

Ж. Ж. Кибамба*, А.Н. Пылькин**

*Рязанский государственный радиотехнический университет им. В. Ф. Уткина
Россия, Рязань, *jacqueskibamba@gmail.com, **pylkin.a.n@rsreu.ru
Демократическая Республика Конго*

Аннотация. В работе рассматриваются общую структуру мобильного приложения операционной системе Android.

Ключевые слова: Android, Activity, Fragment, ViewModel, LiveData, Repository.

MOBILE APPLICATION FOR PLACING ADS ABOUT PRODUCTS BY STUDENTS

J. J. Kibamba*, A. N. Pylkin**

*Ryazan State Radio Engineering University named after V. F. Utkin,
Russia, Ryazan, *jacqueskibamba@gmail.com, **pylkin.a.n@rsreu.ru
Democratic Republic of the Congo*

Abstract. The paper discusses the general structure of a mobile application on the Android operating system.

Keywords: Android, Activity, Fragment, ViewModel, LiveData, Repository.

Часто студенты, которые проживают в общежитиях конкретного университета, после окончания учебы возвращаются в свои родные города, оставляя в своих комнатах свою мебель и другие вещи, которые они продали бы другим студентам, если бы было приложение для размещения объявлений о товарах. В настоящее время существуют несколько мобильных приложений, позволяющих разместить объявления о товарах. Однако ни одно из этих приложений не разработано для студентов, проживающих в общежитии, с помощью которого они смогут продать и купить, не выходя из общежития. В предлагаемой работе сделана попытка для устранения недостатков существующих мобильных приложений для размещения объявлений о товарах.

Мобильное приложение разработано на операционной системе Android [1, 2], с помощью которой можно разработать приложения для смартфонов, планшетов, электронных книг и других устройств. На рисунке 1 представлена общая структура приложения.

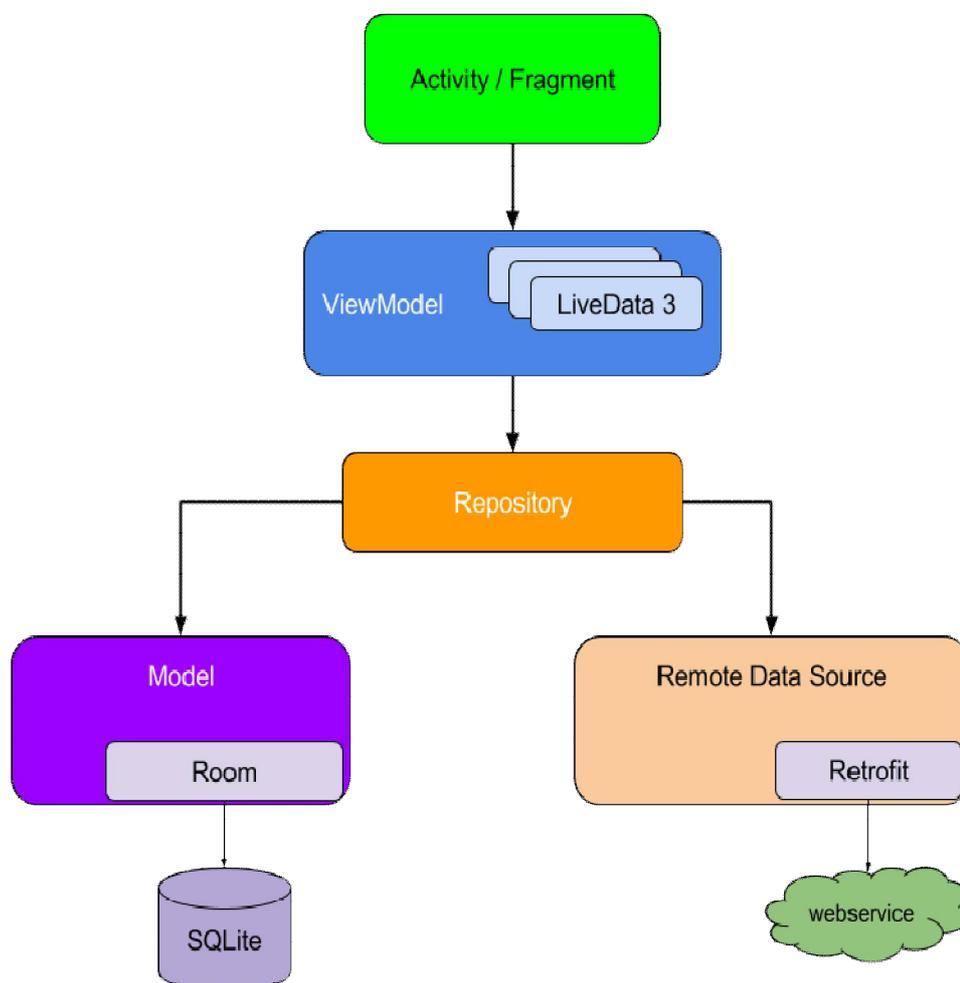


Рис. 5. Общая структура мобильного приложения на ОС Android

Компоненты приложения

Activity – это отдельный экран в Android. Activity позволяет нам разместить все наши компоненты пользовательского интерфейса или виджеты на этом экране.

Fragment - модульная часть activity, у которой свой жизненный цикл и свои обработчики различных событий. Android добавил фрагменты с API 11, для того, чтобы разработчики могли разрабатывать более гибкие пользовательские интерфейсы на больших экранах, таких как экраны планшетов.

ViewModel – предназначен для хранения и управления данными, связанными с представлением, а заодно, избавить нас от проблемы, связанной с пересозданием activity во время таких операций, как переверт экрана и т.д.

LiveData – предназначен для хранения объекта и разрешает подписаться на его изменения. Ключевой особенностью является то, что компонент осведомлен о жизненном цикле и позволяет не беспокоиться о том, на каком этапе сейчас находится подписчик, в случае уничтожения подписчика компонент отпишет его от себя.

Repository – не является специальным компонентом Android. Это простой класс без какой-либо конкретной реализации, который отвечает за выборку данных из всех доступных источников, от базы данных до веб-служб. Он обрабатывает все эти данные, как правило, преобразуя их в наблюдаемые LiveData и делая их доступными для ViewModel.

База данных *Room* - это библиотека *SQLite*, которая облегчает процесс работы с базой данных. Она автоматически записывает готовый код, проверяет ошибки во время компиляции, и, самое главное, она может напрямую возвращать запросы с наблюдаемыми LiveData.

Remote Data Source – Это та часть, где мы взаимодействуем с REST API, так что источник данных используется для удаленного извлечения данных. При извлечении данных мы используем типобезопасную модификацию HTTP-клиента Retrofit.

Мобильное приложение

Был разработан первый компонент мобильного приложения – все файлы фрагментов и основного activity. На рисунке 2 показан скриншот объявлений и заявления, а рисунок 3 показывает скриншот экрана для покупок. То есть студент выделяет нужный продукт на первом экране и сразу переходит на второй экран, где он может прочитать подробнее о продукте и владельце – номер комнаты и имя автора, а также есть возможность написать сообщение студенту, разметившему объявление, если ему нужно купить объявленный товар.

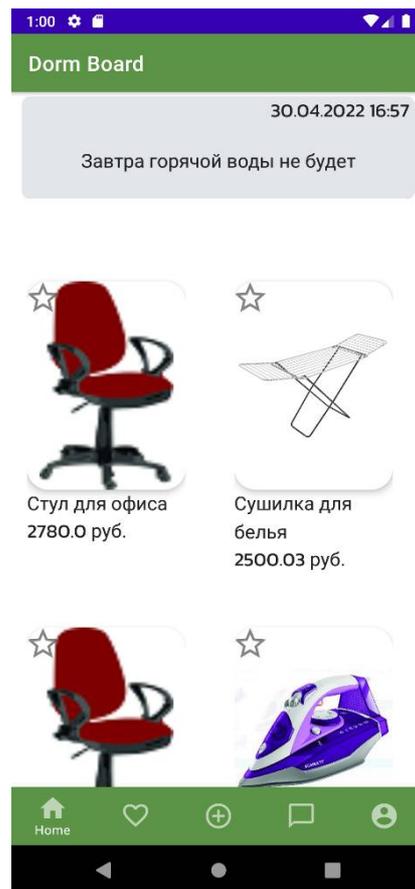


Рис. 6. Скриншот домашнего экрана



Рис. 7 Скриншот экрана для покупок

Выводы

Разработано мобильное приложение для Android, позволяющее студентам университетского общежития получать всю необходимую информацию для покупки-продажи различных предметов обихода, находящихся в пользовании студентов общежития, выполнять заявки на продажу и покупку и реализовывать их, не выходя из общежития.

Библиографический список

1. Архитектура мобильного приложения Android: подробное руководство / [Электронный ресурс] Режим доступа: https://codernet.ru/articles/drugoe/architektura_mobilnogo_prilozhniya_android_podrobnoe_rukovodstvo/. – Дата доступа: 05.05.2022.
2. Руководство по организации архитектуры Android приложения / [Электронный ресурс] Режим доступа: <https://habr.com/ru/post/456256/>. – Дата доступа: 05.05.2022.

УДК 004.62; ГРНТИ 50.41

ОБЗОР КЛАСТЕРИЗАЦИИ ДАННЫХ С ПОМОЩЬЮ ГРАФОВЫХ АЛГОРИТМОВ

А.П. Серов*, А.О. Торжкова**

*Рязанский государственный радиотехнический университет имени В.Ф. Уткина,
Российская Федерация, Рязань, *artseroff@gmail.com, **annet.2001@yandex.ru*

Аннотация. В работе дан краткий обзор кластеризации и рассмотрена кластеризация данных с помощью графовых алгоритмов: выделения связных компонент и минимального покрывающего дерева.

Ключевые слова: анализ данных, кластеризация, графы, алгоритм выделения связных компонент, алгоритм минимального покрывающего дерева.

OVERVIEW OF DATA CLUSTERING USING GRAPH ALGORITHMS

A.P. Serov*, A.O. Torzhkova**

*Ryazan State Radio Engineering University named after V.F. Utkin,
Russia, Ryazan, *artseroff@gmail.com, **annet.2001@yandex.ru*

Abstract. The paper gives a brief overview of clustering and discusses data clustering using graph algorithms: connected component extraction and minimum spanning tree.

Keywords: data analysis, clustering, graphs, connected component extraction algorithm, minimum spanning tree algorithm.

Кластеризация – это процесс анализа данных, в результате которого анализируемые объекты разделяются на группы так, что объекты внутри группы более похожи друг на друга, а объекты из разных группы менее похожи. Такие группы называются кластерами.

Сферы применения кластеризации

Маркетинг. Кластеризация широко используется при анализе рынка для обработки данных, полученных из различных опросов и анкет. Может применяться для исследования пользователей, подбора рекомендаций на основе их интересов, разработки новых продуктов.

Биология и медицина. С помощью кластеризации осуществляются: выделение пространственно однородных групп организмов, сообществ и т.п.; группировка схожих геномных последовательностей в семейство генов, которые являются общими структурами для многих организмов и могут выполнять схожие функции; выявление шаблонов устойчивости к антибиотикам, кластеризация препаратов, симптомов.

Информатика. Кластеризация может быть использована для: повышения степени ответственности ответов поисковым запросам путем группировки веб-сайтов, файлов других объектов по смысловым значениям поискового запроса; разбиения цифрового изображения на отдельные области с целью обнаружения границ или распознавания образов; для анализа данных; также используется в машинном обучении.

Формальная постановка задачи кластеризации

Формально постановку задачи кластеризации можно описать следующим образом:

Имеется конечная выборка объектов, где X – множество объектов. Также задана функция которая вычисляет расстояние между двумя объектами по некоторым критериям (метрике) – $\rho(x_i, x_j)$, $x_i, x_j \in X$. Каждому объекту $x_i \in X$ необходимо поставить в соответствие один из идентификаторов кластеров $u_k \in Y$, где Y – множество кластеров, таким образом, чтобы объекты, которым соответствует один и тот же идентификатор кластера u_k были близки относительно метрики ρ , а объекты с разными идентификаторами кластеров сильно отличались. Соответственно, алгоритм кластеризации – это функция, которая любому объекту $x_i \in X$ ставит в соответствие один из идентификаторов кластеров $u_k \in Y$.

Чаще всего множество Y не известно заранее и требуется определить оптимальное количество кластеров. Этим задача кластеризации отличается от классификации, в которой группы объектов заранее определены на основании некоторых критериев и нужно лишь определить, к какой группе относится каждый объект.

Входные данные представлены в виде выборки объектов, которые имеют набор характеристик. Например, выборка данных медицинских показаний студентов может содержать следующую информацию: ФИО студента, номер группы, факультет, рост, вес, пульс, артериальное давление и т.п. Эти характеристики могут быть двух видов: числовые и категориальные.

Задача кластеризации неоднозначна. При решении задачи кластеризации нельзя дать однозначную оценку качества решения, так как результат сильно зависит от выбора метрики (по каким признакам и каким способом будет определяться степень схожести объектов) и алгоритма кластеризации, который является субъективным. Не существует способа определения идеально подходящих метрики и алгоритма.

Метрики определения расстояния между объектами

Для определения расстояния между объектами прежде всего определяются характеристики (числовые и/или категориальные), которые будут участвовать в анализе. Затем производится нормализация характеристик, которая заключается в приведении всех значений характеристик к одному числовому диапазону. Это необходимо для того, чтобы все характеристики объекта одинаково сильно влияли на расстояние между каждой парой объектов. Чаще всего используются диапазоны $[0; 1]$ или $[-1; -1]$.

Далее представлены некоторые метрики расчёта расстояний между объектами [1].

Евклидово расстояние:

$$\rho(x, x') = \sqrt{\sum_i^n (x_i - x'_i)^2}.$$

Манхэттенское расстояние:

$$\rho(x, x') = \sum_i^n |x_i - x'_i|.$$

Расстояние Чебышева:

$$\rho(x, x') = \max (|x_i - x'_i|).$$

Степенное расстояние:

$$\rho(x, x') = \sqrt[r]{\sum_i^n (x_i - x'_i)^p}.$$

где p отвечает за постепенное взвешивание разностей по отдельным координатам, а r – за прогрессивное взвешивание больших расстояний между объектами.

Классификация алгоритмов кластеризации

Существует множество алгоритмов кластеризации и их модификаций. Для всех них трудно составить единую классификацию, но можно ввести несколько классификаций по тем или иным признакам [2].

Алгоритмы кластеризации можно классифицировать несколькими способами: иерархические и плоские, чёткие и нечёткие. Также можно выделить отдельную группу графовых алгоритмов.

На рисунке 1 представлена классификация алгоритмов кластеризации.

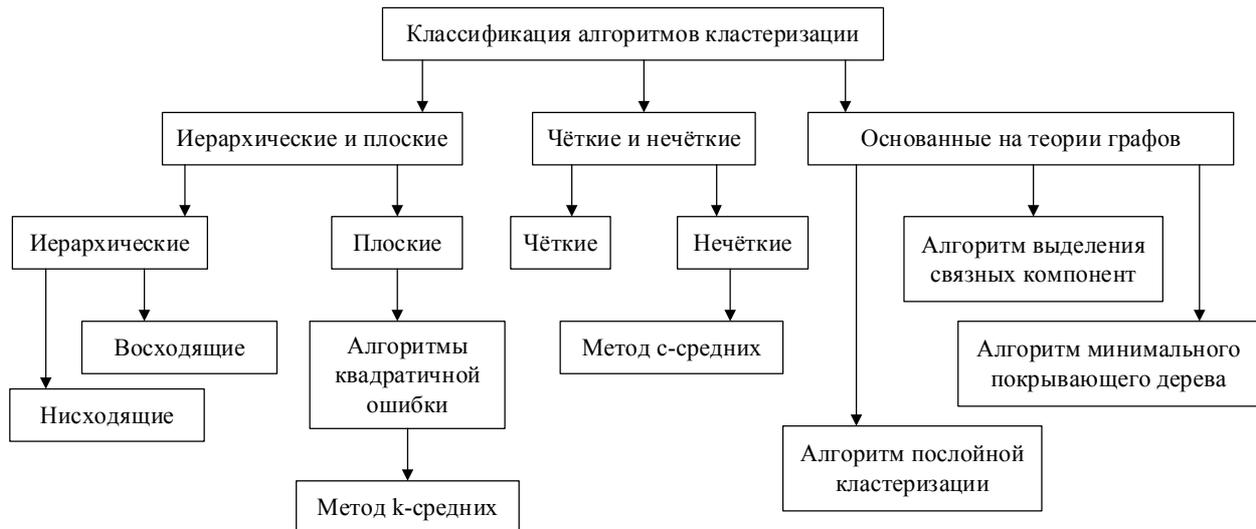


Рис. 1. Классификация алгоритмов кластеризации

Иерархические и плоские алгоритмы характеризуются наличием или отсутствием нескольких уровней кластеризации. Если плоские алгоритмы просто присваивают каждому объекту один из идентификаторов кластера, то иерархические разбивают объекты на кластеры, которые в свою очередь разбиты на более мелкие кластеры и так далее. В результате работы иерархических алгоритмов получается дерево кластеров. Иерархические алгоритмы бывают двух типов: нисходящие, которые разбивают кластеры на всё более мелкие, и восходящие, которые сначала выделяют самые мелкие кластеры, а потом объединяют их во всё более крупные. К плоским алгоритмам относятся алгоритмы квадратичной ошибки, например метод k -средних.

Чёткие алгоритмы ставят каждому объекту в соответствие только один идентификатор кластера, а нечёткие для каждого объекта определяют вероятность принадлежности его к каждому кластеру, то есть каждый объект имеет набор вещественных значений отношения его к каждому из кластеров. К нечётким алгоритмам относится метод c -средних, который является модификацией метода k -средних.

Графовые алгоритмы, как следует из названия, основаны на представлении выборки в виде графа, где узлы графа – это объекты выборки, а рёбра – это расстояния между объектами, определённые метрикой. Отсюда следует, что граф является взвешенным.

Преимущество графовых алгоритмов заключается в том, что они просты в реализации и дают наглядный результат. К недостаткам относится чувствительность к выбросам (неоднородности выборки). К графовым алгоритмам относятся алгоритмы выделения связанных компонент, минимального покрывающего дерева и послойной кластеризации.

Алгоритм выделения связанных компонент

Как следует из сказанного выше, алгоритм выделения связанных компонент основан на представлении выборки в виде графа. В качестве входных данных помимо выборки требуется задать параметр R – пороговое значение весов рёбер. Граф выборки может быть задан как список объектов и матрица расстояний между объектами, где каждая ячейка с индексами i и j хранит вес ребра – расстояние $\rho(x_i, x_j)$, $x_i, x_j \in X$. Так как $\rho(x_i, x_j) = \rho(x_j, x_i)$, то матрица является симметричной, при чём на главной диагонали все элементы будут равны 0.

Алгоритм заключается в последовательном удалении всех рёбер, вес которых больше, чем параметр R . Таким образом в графе останутся рёбра, соединяющие более близкие объекты. В результате граф распадётся на несколько подграфов, которые и будут являться класте-

рами. Также эти подграфы называют связными компонентами, потому что в каждом подграфе каждая вершина должна быть связана ребром с другой вершиной этого подграфа, и напротив не должно существовать ребер, соединяющих вершины разных подграфов.

Результат работы алгоритма сильно зависит от заданного параметра R . Чем он меньше, тем больше ребер будет удалено и тем больше кластеров получится.

Данный алгоритм наиболее подходит для выделения кластеров типа сгущений или лент. Наличие разреженного фона или «узких перемычек» между кластерами приводит к неадекватной кластеризации.

Недостатком алгоритма является плохая управляемость числом кластеров. Приходится многократно решать задачу при разных значениях R , что сильно увеличивает время работы алгоритма.

Алгоритм минимального покрывающего дерева

Алгоритм минимального покрывающего дерева применяется для кластеризации больших наборов произвольных данных и выделяет кластеры произвольной формы.

В процессе работы алгоритма строится минимальное остовное дерево – граф $G = (V, E)$, где V – число вершин, E – число ребер. В алгоритме кластеризации методом минимального остовного дерева задается число кластеров k . Алгоритм может также и работать с числом R (порог расстояния для удаления ребер). Однако, как выше было сказано, управлять числом кластеров с помощью параметра R довольно затруднительно.

Число кластеров k можно также определять графически, если упорядочить все расстояния, образующие каркас, в порядке убывания и отложить их на графике. Резкий скачок вниз где-то на начальном участке графика будет соответствовать количеству наиболее четко выделяемых кластеров.

На вход алгоритма поступает такая же матрица расстояний, как для алгоритма выделения связных компонент. На ее основе строится минимальное остовное дерево, то есть ациклическое дерево, содержащее все вершины графа с минимально возможным весом ребер. Для построения минимального остовного дерева может быть использован алгоритм Краскала. Затем удаляется $k - 1$ ребро остовного дерева с максимальным весом. Полученные k компонент связности являются кластерами.

Алгоритм Краскала заключается в следующем: алгоритм порождает остовное дерево посредством разрастания только одного поддерева, например X_p , содержащего больше одной вершины. Алгоритм начинает работу с включения в поддерево начальной вершины. Поддерево постепенно разрастается за счет присоединения ребер (x_i, x_j) , где $x_i \in X_p$ и $x_j \in X_p$; причем добавляемое ребро должно иметь наименьший вес c_{ij} . Процесс продолжается до тех пор, пока число ребер в A_p не станет равным $V - 1$ (V – число вершин). Тогда поддерево $G_p = (X_p, A_p)$ будет требуемым остовым деревом.

Научный руководитель работы – д.т.н., проф. А.Н. Пылькин

Библиографический список

1. Прикладная статистика: классификация и снижение размерности: Справ. изд. / С.А. Айвазян, В.М. Бухштабер, И.С. Енюков, Л.Д. Мешалкин; Под ред. С.А. Айвазяна. – М.: Финансы и статистика, 1989. – 607 с.: ил.
2. Кластерный анализ: Научн. изд. / И.Д. Мандель. – М.: Финансы и статистика, 1988. – 176 с.: ил.

СЕКЦИЯ «ЭВМ И СИСТЕМЫ»

УДК 621.396; ГРНТИ 47.47

РАЗРАБОТКА АРХИТЕКТУРЫ ВЕБ-ПРИЛОЖЕНИЯ «ОНЛАЙН АУКЦИОН»

Е.А. Потапов*, Н.В. Шаповалов**, А.И. Шишкин***

Рязанский государственный радиотехнический университет имени В.Ф. Уткина,

*Российская Федерация, Рязань, *Egor0133@yandex.ru,*

***xomka2280@gmail.com, ***artem.shishkin0393@gmail.com*

Аннотация. В работе рассматривается реализация архитектуры системы и базы данных веб-приложения «Онлайн аукцион».

Ключевые слова: онлайн-аукцион (ОА), архитектура (ПИ), база данных(БД), микросервис (МС), монолит(МЛ)

DEVELOPMENT OF THE ARCHITECTURE OF THE WEB APPLICATION "ON-LINE AUCTION"

E.A. Potapov*, N.V. Shapovalov**, A.I. Shishkin***

Ryazan State Radio Engineering University named after V.F. Utkin,

*Russian Federation, Ryazan, *Egor0133@yandex.ru,*

***xomka2280@gmail.com, ***artem.shishkin0393@gmail.com*

The summary. The paper considers the implementation of the architecture of the system and database of the web application "Online Auction".

Keywords: online auction (OA), architecture (PI), database (DB), microservice (MS), monolith (ML)

Одними из первых аукционов, в их современном понимании, стали аукционные дома Голландии, в середине 17-ого века. Лотами на них, как правило, становилось имущество, оставленное в наследство, или же имущество обанкротившихся компаний, что позволяло им выплачивать свои долги. В современном мире популярность и количество аукционов возросли, и существуют различные категории лотов с огромным количеством товаров. Из-за развития технологий в конце 20-ого века стали появляться и начали набирать популярность онлайн-аукционы, которые позволяют людям в любое время и из любой точки мира принимать участия в торгах.

В связи с этим, целью данной статьи является разработка веб-архитектуры программной системы для проведения онлайн-аукционов. Система должна реализовывать следующие модели аукционов: стандартный аукцион (побеждает человек с наибольшей суммой ставки), обратный аукцион (снижение стоимости лота от начальной цены, заданной продавцом) и скандинавский аукцион (тип аукциона с оплатой за каждую ставку).

Система основана на архитектурной модели клиент/сервер. На стороне сервера используется модель архитектуры на основе микросервисов в сочетании с моделью многоуровневой архитектуры.

Несмотря на то, что базовая структура дизайна веб-архитектуры непротиворечива, подходы, которые разработчики используют для создания и соединения этих слоев, различаются.

Выбор и разработка архитектуры

Давайте рассмотрим наиболее распространенные подходы к созданию веб-приложения и связыванию всех компонентов страницы вместе.

Монолитная архитектура

Монолитная архитектура представляет собой традиционный подход к разработке веб-приложения. Такие приложения имеют единую кодовую базу. Когда команда хочет внести изменения в приложение, им необходимо обновить весь файл. Этот подход кажется интуитивно понятным на первых этапах, поскольку все компоненты явно вытекают друг из друга.

Это хороший вариант для небольших продуктов и услуг, но рискованный для масштабируемых платформ. Пример данной архитектуры показан на рисунке 1.

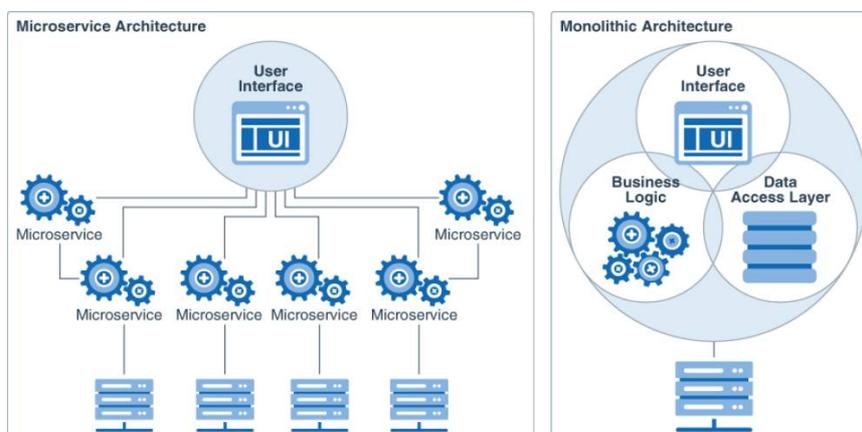


Рис. 2. Сравнение микросервисной и монолитной архитектур

Достоинства:

- простота разработки и развертывания — монолитные приложения существуют с незапамятных времен. Многочисленные инструменты действительно облегчают разработку и развертывание стратегий. Разработчикам необходимо выполнять единый фрагмент развертываемого кода, а не вносить обновления в отдельные объекты;

- производительность — более высокая производительность является важным преимуществом монолитных приложений по сравнению с микросервисами. Приложению на основе микрослужб может потребоваться выполнить 100 различных вызовов API к 100 другим микрослужбам, чтобы загрузить один экран пользовательского интерфейса. В то время как в монолите один вызов API может служить той же цели, поскольку он имеет централизованный код и память.

Недостатки:

- тесная связь — Сервисные модули в монолитных приложениях тесно связаны. Бизнес-логика сильно запутана и затрудняет инкапсуляцию приложения, что затрудняет масштабируемость приложения;

- медленный цикл сборки и выпуска — поскольку кодовая база огромна, это снижает скорость разработки и тестирования приложения.

Микросервисы

Микросервисы — это небольшие развертываемые сервисы, созданные на базе сложных приложений. Микросервис — это не что иное, как более новая версия сервис-ориентированной архитектуры (SOA). Они общаются друг с другом, используя разные методы, и у них также есть преимущество — они не зависят от технологий.

С технической точки зрения микросервисы раскрывают одну бизнес/техническую возможность приложений, которые они инкапсулируют, через несколько небольших сервисов [1]. Различные микросервисы взаимодействуют через конечные точки API/протоколы HTTP, превращая их в распределенную систему.

На рисунке 1 показана распределенная архитектура, которая обеспечивается реализацией микросервисов. Она повышает надежность, масштабируемость и переносимость всего приложения.

Достоинства:

- организованность кода — поскольку вся кодовая база разбита на более мелкие службы, она относительно лучше организована. Микросервисы выполняют определенную работу и не зависят от других компонентов;

- гибкость разработки — с микросервисами отдельные члены команды могут работать над отдельными модулями. Каждый человек может создать модуль и развернуть модуль самостоятельно, тем самым уменьшив разногласия в команде и повысив гибкость программного обеспечения

Недостатки:

- повышенная сложность — хотя отдельные компоненты приложения могут быть относительно простыми, приложение на основе микросервисов в целом может быть невероятно сложным. Способ, которым микросервисы связаны друг с другом, добавляет уровень сложности, которого нет в монолитных приложениях;
- дополнительные расходы — использование микросервисов может сэкономить некоторые расходы, но также, вероятно, потребует дополнительных ресурсов разработки для управления каждым микросервисом и его зависимостями.

В ходе рассмотрения двух архитектур, в нашем проекте будут использованы микросервисы, так как они позволяют легко масштабировать и разрабатывать приложение. Ознакомиться с нашей архитектурой можно на рисунке 2.

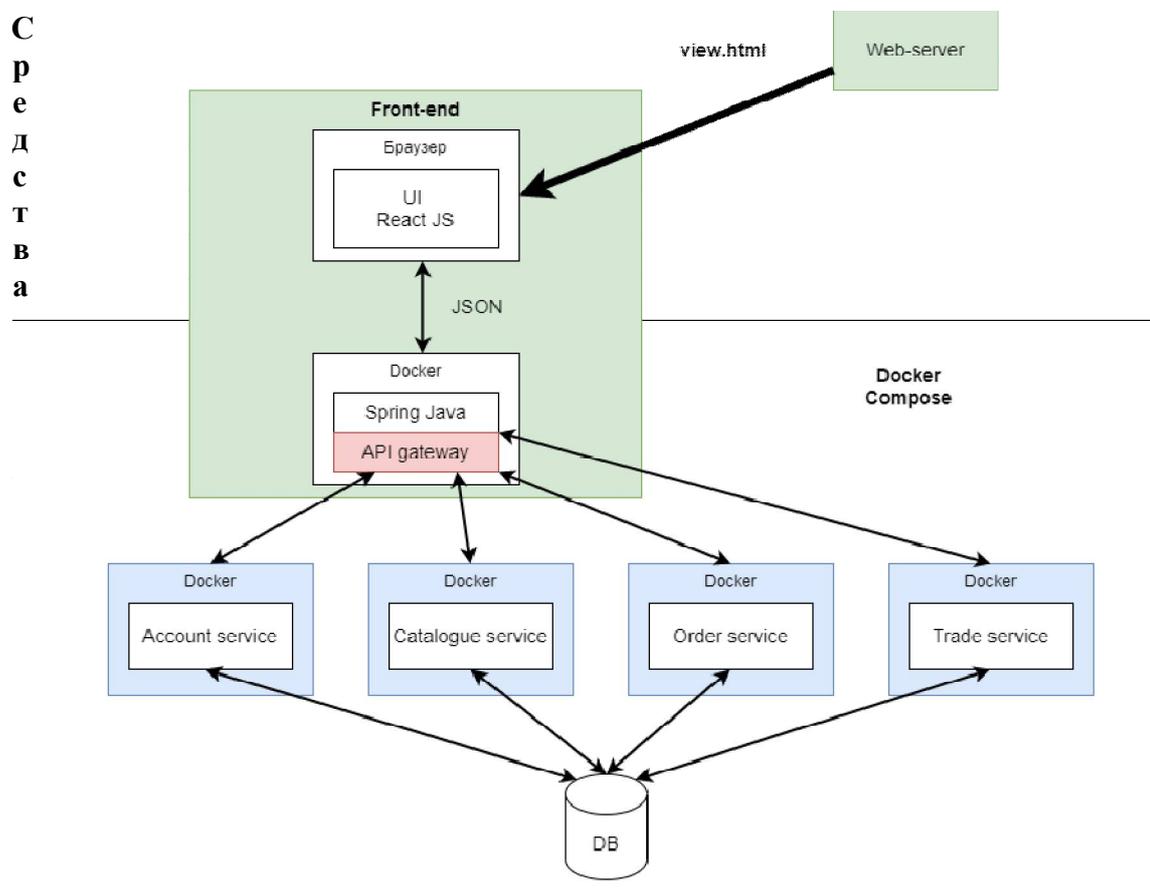


Рис. 2. Архитектура веб-приложения "Онлайн аукцион"

Многие современные программисты предпочитают в своей работе пользоваться Java. Несмотря на то, что спросом пользуется С-семейство, Java тоже не уступает. Связано это с тем, что данный язык является универсальным. Он удобен, понятен и практичен [2].

С самого своего возникновения стал активно развиваться и совершенствоваться. Сейчас Java удобно использовать как для компьютерных программ, так и для мобильных платформ. Это – полноценный язык, поддерживающий объектно-ориентированное программирование. Ключевая особенность Java – возможность создавать веб-приложения и расширения.

У Java следующие особенности:

- наличие функционала для ООП;
- понятный и относительно простой синтаксис – освоить его может даже начинающий программист-любитель;
- хорошая пользовательская поддержка – она дружелюбна к новичкам;
- достаточное количество документации на всех языках, включая русский;
- движок и библиотеки для написания игр и всевозможных веб утилит.

В нашем проекте помимо языка, будут использоваться дополнительные библиотеки и фреймворки, позволяющие ускорить написания нашего проекта.

Выделим основные:

- Spring Boot — один из самых популярных универсальных фреймворков для построения веб-приложений на Java [3].
- Для развёртывания приложения и базы данных воспользуемся Docker.
- Spring Data JPA – часть фреймворка Spring, которая обеспечивает взаимодействие с базой данных.
- MapStruct – для взаимодействия с объектами между слоями приложения.

Разработка базы данных веб-приложения

Для работы с базой данных была выбрана реляционная СУБД PostgreSQL. Почему выбрана PostgreSQL, а не стандартная и более популярная СУБД MySQL?

Во-первых, сроки разработки проекта не критичны, поэтому есть время для детальной настройки СУБД.

Во-вторых, у PostgreSQL есть важное преимущество: он более строгий, делает больше проверок, дает меньше возможности ошибиться, и это в перспективе огромное преимущество. Например, в MySQL приходится писать собственные инструменты для верификации обычной ссылочной целостности базы. И даже с этим могут быть проблемы. В этом смысле PostgreSQL инструмент более мощный, более гибкий, разрабатывать на нем приятнее.

Схема базы данных для веб-приложения «Онлайн-аукцион» показана на рисунке 3.

Система содержит 5 основных сущностей – аукцион (Auction), продукт (Product), лот (Lot), ставка (Bet) и пользователь (User).

Сущность пользователя связана с сущностью Роль (Role), в которой хранится набор допустимых ролей в приложении. Также стоит обратить внимание на хранение пароля в БД. Для правильного хранения паролей в БД будем использовать криптографическую хеш-функцию, основанную на шифре Blowfish. При регистрации клиента мы генерируем соль и используем её для вычисления хеша. В результате получаем строку, содержащую соль и хеш пароля. Полученное значение сохраняем в БД. Пример хеша - \$2a\$12\$3hcF5Cb.RuyuKerAa.4i/OCVptU6WwOsvCdzhL7C34atcLJ.XGhNu:

- \$2a\$12\$— заголовок (алгоритм bcrypt и количество раундов хеширования)
- 3hcF5Cb.RuyuKerAa.4i/O — соль (16 байт)
- CVptU6WwOsvCdzhL7C34atcLJ.XGhNu — хеш (24 байта)

Сущность аукцион характеризуется тремя сущностями: тип аукциона (AuctionType), в которой определяется тип созданного аукциона; настройки цены (PriceSetting), в которой задаются стартовая цена на лот, шаг изменения цены и комиссия, которую забирает аукцион от суммы ставки на лот; настройки даты (DateSetting), в ней задаются дата и время окончания аукциона и интервал времени, по которому закрывается аукцион и определяется победитель.

Сущность продукта связана со вспомогательной сущностью категория, которая определяет принадлежность продукта к определенной категории для обеспечения фильтрации и структурированности каталога приложения.

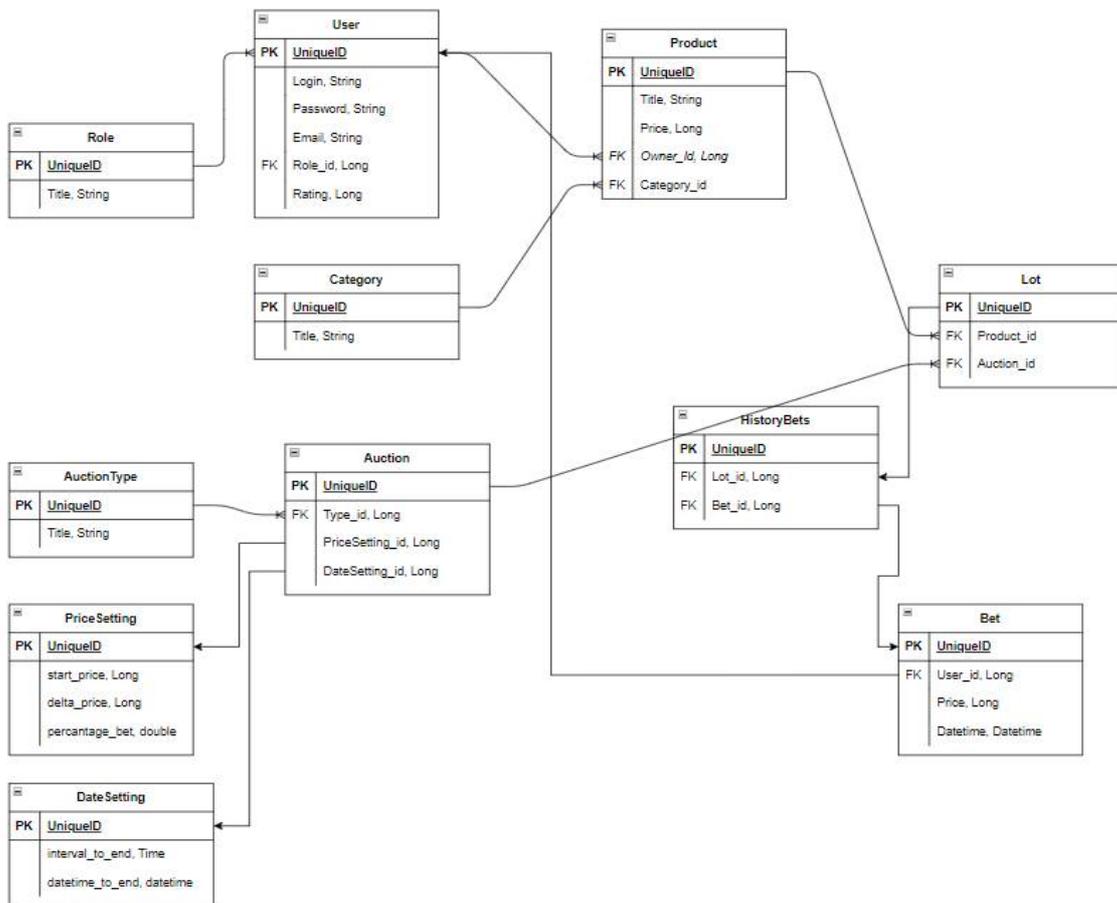


Рис. 3. Схема базы данных приложения

Сущность аукцион характеризуется тремя сущностями: тип аукциона (AuctionType), в которой определяется тип созданного аукциона; настройки цены (PriceSetting), в которой задаются стартовая цена на лот, шаг изменения цены и комиссия, которую забирает аукцион от суммы ставки на лот; настройки даты (DateSetting), в ней задаются дата и время окончания аукциона и интервал времени, по которому закрывается аукцион и определяется победитель.

Сущность продукта связана со вспомогательной сущностью категория, которая определяет принадлежность продукта к определенной категории для обеспечения фильтрации и структурированности каталога приложения.

Сущности лота и ставки связаны между собой и объединяются, в таблице история ставок (HistoryBets). Лот характеризуется продуктом, на который он открыт, и аукционом, который определяет правила торговли этого лота. А ставка характеризуется пользователем, который ее поставил, суммой ставки и датой и временем.

Заключение

В ходе данной статьи были рассмотрены два самых популярных подхода по архитектуре веб-приложений, их преимущества и недостатки, исходя из чего была разработана микросервисная архитектура веб-приложения «Онлайн-аукцион». В статье подробно разбирается разработка базы данных приложения, сильно зависящая от выбранной архитектуры.

Библиографический список

1. Козмина Юлиана, Харроп Роб. Spring 5 для профессионалов / Козмина Юлиана, Харроп Роб
2. Крис Ричардсон. Микросервисы. Паттерны разработки и рефакторинга. / Крис Ричардсон.
3. Блинов И.Н. Java Методы программирования. / И.Н. Блинов

УДК 681.391; ГРНТИ 50.33

АНАЛИЗ МЕТОДОВ ИСПРАВЛЕНИЯ ОШИБОК В СОВРЕМЕННЫХ СИСТЕМАХ ХРАНЕНИЯ ДАННЫХ

Чан Туан Зунг

Рязанский государственный радиотехнический университет имени В.Ф.Уткина,
Российская Федерация, Рязань, dungk4@gmail.com

Аннотация. В работе рассматриваются методы коррекции ошибок в современных системах хранения данных. Приводятся их основные характеристики, обсуждаются достоинства, недостатки и области применения.

Ключевые слова: методы коррекции ошибок, система хранения данных, коды Рида – Соломона, коды БЧХ, коды LDPC.

ANALYSIS OF ERROR CORRECTION METHODS IN MODERN DATA STORAGE SYSTEMS

Tran Tuan Dung

Ryazan State Radio Engineering University named after V.F.Utkin
Russia, Ryazan, dungk4@gmail.com

Annotation. The paper discusses modern methods of error correction in modern data storage systems. The main characteristics, advantages, disadvantages and areas of application in memory are discussed.

Keywords: error correction methods, data storage system, Hamming codes, Reed-Solomon codes, BCH codes, LDPC codes.

Коды коррекции ошибок (ЕСС) используются в памяти для обнаружения и исправления битовых ошибок. С уменьшением технологических узлов и увеличением сложности памяти, частота ошибок по битам продолжает расти. Метод коррекции ошибок - хороший способ восстановить неправильное значение из оставшихся хороших битов. Методы коррекции ошибок позволяют исправлять ошибки в памяти и бороться с повреждением данных.

В настоящее время существует много методов коррекции ошибок в памяти например коды Хемминга, коды Рида-Соломона, коды БЧХ, Low-density parity-check code (LDPC). В данной статье рассмотрим принцип работы, особенности использования, а также достоинства и недостатки этих методов.

Исправление ошибок в памяти выполняется путем добавления битов проверочных на каждую страницу пользовательских данных, так что добавленных битов проверочных будет достаточно для исправления 1 или нескольких битовых ошибок на странице.

Коды БЧХ и коды Рида-Соломона

Коды Рида-Соломона и БЧХ используются для исправления многобитовых ошибок. Коды Рида-Соломона [1] - 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 блока ($n = 2^m - 1$)

k – Длина сообщения

t – Возможность исправления ошибок (т. е. t ошибок могут быть исправлены)

Для кодов БЧХ $(n-k) \leq m * t$. Поскольку $(n - k)$ - это количество битов проверочных, максимальное количество битов проверочных равно $m * t$.

Таким образом, 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, моделирование было выполнено с использованием кодов БЧХ на канале с АБГШ(с аддитивным белым гауссовским шумом) с использованием модуляции BPSK(двоичная фазовая манипуляция). Параметры кодов БЧХ, используемых при моделировании, следующие (таблица 1):

Таблица 1. Параметры моделирования для анализа эффективности кодов БЧХ (BCH)

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 представлен эффективность кодов БЧХ, для разных скоростей кодирования.

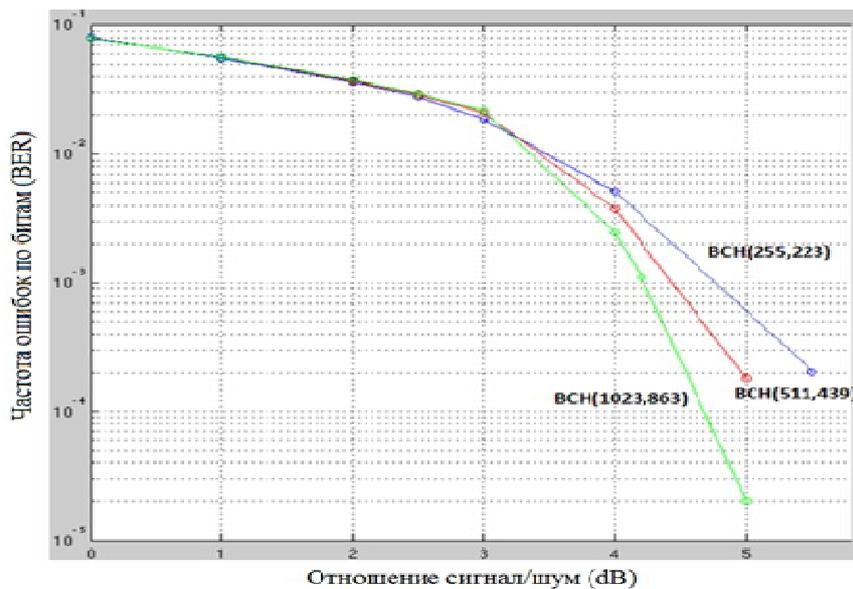


Рис. 2. Эффективность кодов БЧХ

1. Результаты показывают, что по мере увеличения кодовой скорости эффективность кодов БЧХ ухудшается.

2. Можно видеть, что с точки зрения возможностей исправления ошибок БЧХ (511, 439) эквивалентен с БЧХ (255, 223) (оба могут исправить 8 ошибок). Однако БЧХ (511, 439) эффективнее, чем БЧХ (255, 223) для та же возможность исправления ошибок.

Коды Рида-Соломона эффективны для исправления пакетных ошибок, поскольку исправление одного символа исправит несколько кластеризованных битовых ошибок. Однако эффективность кодов БЧХ все еще далеко от предела пропускной способности Шеннона [3].

Коды LDPC

Чтобы проанализировать эффективность кодов LDPC, моделирование было выполнено с использованием кодов LDPC на канале с АБГШ (с аддитивным белым гауссовским шумом) с использованием модуляции QPSK (четырёхпозиционная фазовая модуляция). Параметры кодов БЧХ, используемых при моделировании, следующие (таблица 2):

Таблица 2. Параметры моделирования для анализа эффективности кодов LDPC

Кодовая скорость	Размеры N - матрицы
1/2	32400x64800
2/3	21600x64800
3/4	16200x64800
5/6	10800x64800
7/8	8100x64800

Результаты моделирования для коэффициента битовых ошибок относительно отношения сигнал / шум (рис. 3).

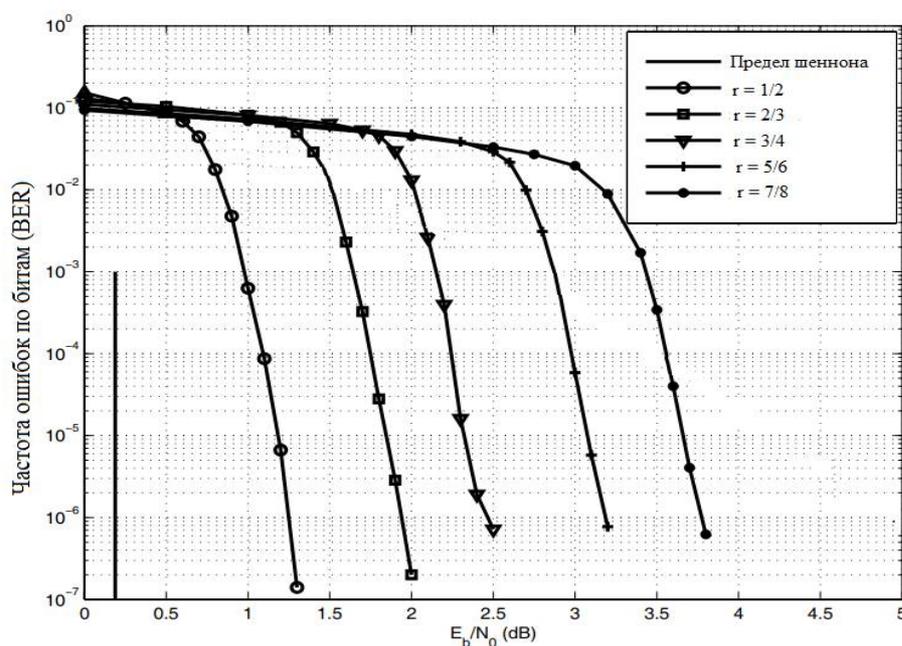


Рис. 3. Эффективность кодов LDPC

Результаты показывают, что по мере увеличения кодовой скорости эффективность кодов LDPC ухудшается. Эффективность кодов LDPC довольно близка к теоретическим пределам, то есть пределу пропускной способности Шеннона. Однако эффективность кодов LDPC все еще далеко от предела пропускной способности Шеннона [4].

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 681.391; ГРНТИ 50.33

ЭФФЕКТИВНОСТЬ ПРИМЕНЕНИЯ СИМВОЛЬНЫХ МНОГОПороГОВЫХ ДЕКОДЕРОВ В СИСТЕМАХ ХРАНЕНИЯ ДАННЫХ

Чан Туан Зунг

*Рязанский государственный радиотехнический университет имени В.Ф.Уткина,
Российская Федерация, Рязань, dungdk4@gmail.com*

Аннотация. В работе рассматриваются эффективность применения символьных многопороговых декодеров в системах хранения данных.

Ключевые слова: метод коррекции ошибок, система хранения данных, qMPTD

EFFICIENCY OF SYMBOL MULTITHRESHOLD DECODERS IN DATA STORAGE SYSTEMS

Tran Tuan Dung

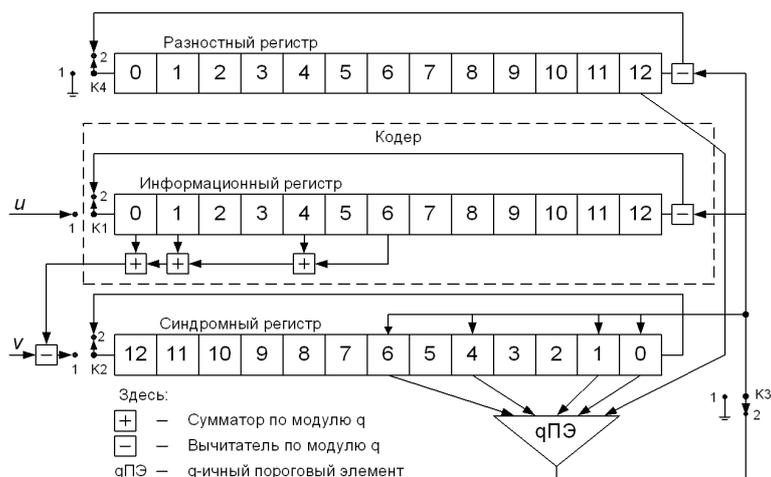
*Ryazan State Radio Engineering University named after V.F.Utkin
Russia, Ryazan, dungdk4@gmail.com*

Annotation. The paper discusses the effectiveness of the use of symbolic multithreshold decoders in data storage systems.

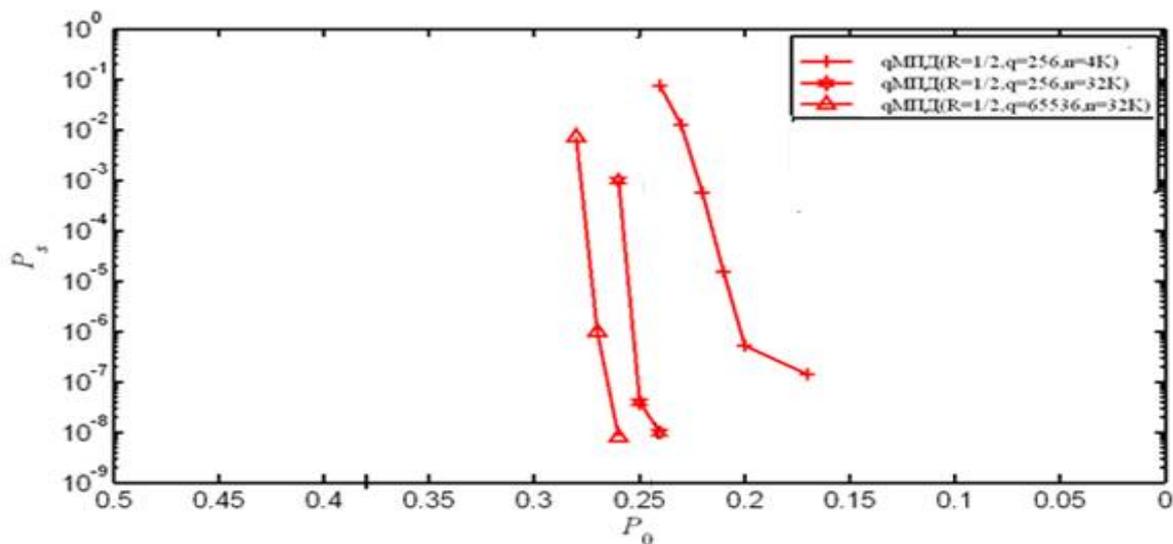
Keywords: error correction method, data storage system, qMTD

Наиболее эффективным средством повышения достоверности цифровой информации при ее передаче по каналам с шумом является применение помехоустойчивого кодирования. В настоящее время существует много методов коррекции ошибок в памяти например коды БЧХ, Коды с низкой плотностью проверок на четность LDPC, символьный многопороговый декодер (qMPTD). Однако наиболее предпочтительными для применения в системах хранения данных являются многопороговые декодеры. В данной работе рассматривается символьный многопороговый декодер (qMPTD). Приводятся его основные характеристики, обсуждаются достоинства, недостатки и области применения.

Принцип работы qMPTD представлен на рис. 1.

Рис. 1. q МПД блочного СОК с $r=1/2$, $d=5$ и $n=26$.

Для представления зависимости эффективности декодирования символьного многопорогового декодера на рисунке 1 представлен график, для разных длин кодов, с кодовой скоростью равной $1/2$.

Рис. 2. График эффективности декодирования q МПД для кодов разных длин

На рисунке 2 кривые 1 и 2 показывают характеристики недвоичных многопороговых декодеров для однобайтовых кодов длиной около 4000 и 32000 символов соответственно. Кривая 3 на этом же рисунке показывает характеристики недвоичных многопороговых декодеров для двухбайтовых кодов длиной около 32000 символов. Отметим, что q МПД позволяют исправлять значительно большую долю ошибок в канале

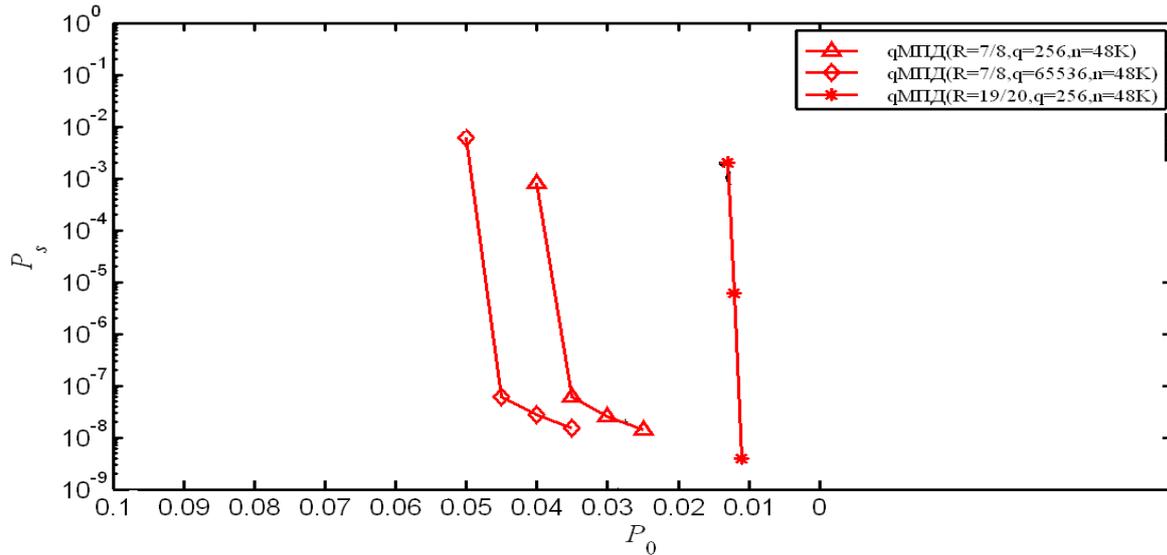


Рис. 3. Результаты моделирования qMPPD с различным кодовым скоростями

Кривые 1 и 2 на рис. 3 показывают характеристики недвоичных многопороговых декодеров для однобайтовых и двухбайтовых кодов, соответственно, при кодовой скорости $R=7/8$. Отметим, что недвоичные многопороговые декодеры для двухбайтовых символов оказываются не сложнее, чем однобайтового. Кривая 3 показывает характеристики недвоичных многопороговых декодеров для однобайтовых символов при кодовой скорости $R=19/20$. Заметим, что и при более высокой кодовой скорости qMPPD оказываются эффективный.

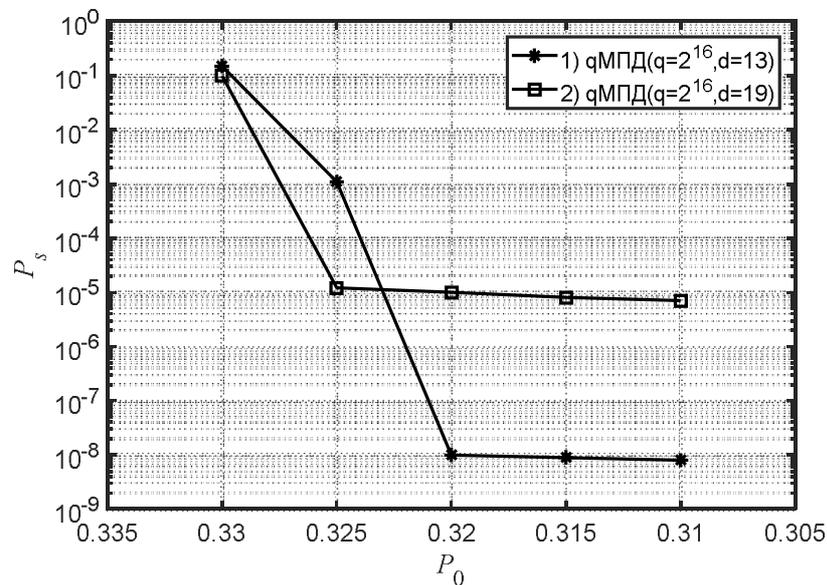


Рис. 4. Результаты моделирования qMPPD с различным кодовым расстоянием

На рисунке 4 представлены графики зависимости полученных дивергентных схем кодирования в q -ичном симметричном канале с алфавитом $q=2^{16}$. Кривая 1 соответствует СОК с $d=13$, а кривая 2 – коду с $d=19$. Задержка решения символьного МПД второго кода составляет 1.2 млн. информационных символов, его скорость декодирования ~ 10 тыс. двухбайтовых символов/с при $l=150$ итерациях. В обоих случаях символьный МПД реализует характеристики ОД для использованных кодов, но при собственной линейной, т.е. теоретически минимальной сложности [3].

Проведенные исследования позволяют сформулировать основные выводы и результаты:

- qМПД позволяют исправлять значительно большую долю ошибок в канале.
- Недвоичные многопороговые декодеры для двухбайтовых символов оказываются не сложнее, чем однобайтового.
- При более высокой кодовой скорости qМПД оказываются эффективный.
- Символьный МПД реализует характеристики ОД для использованных кодов, но при собственной линейной, т.е. теоретически минимальной сложности.
- Недвоичные многопороговые декодеры являются наиболее простым и эффективным методом коррекции ошибок для систем хранения больших объемов цифровых данных.

Библиографический список

1. Многопороговые декодеры. Веб-сайт ИКИ РАН www.mtdbest.iki.rssi.ru.
2. Золотарёв В.В., Овечкин Г.В. О сопоставлении новых методов помехоустойчивого кодирования //18 Международная конференция "Цифровая обработка сигналов и её применение – DSPA 2016". М., 2016, Т.1, с.59-65.
3. Зубарев Ю.Б., Золотарёв В.В., Овечкин Г.В. Теория кодирования как оптимизационная проблема декодирования вблизи границы Шеннона // 21 международная конференция «Цифровая обработка сигналов и ее применение». 2019, Вып. XXI-1. с.10-16.

УДК 004.93'1, 159.942.33; ГРНТИ 28.23.15, 15.01.77

АЛГОРИТМ ВЫЧИСЛЕНИЯ ДЕСКРИПТОРА ЛОКАЛЬНЫХ БИНАРНЫХ ШАБЛОНОВ ПО ТРЕМ ОРТОГОНАЛЬНЫМ ПЛОСКОСТЯМ ДЛЯ РАСПОЗНАВАНИЯ МИКРОВЫРАЖЕНИЙ ЛИЦА

А.Д. Черненко

*Рязанский государственный радиотехнический университет имени В.Ф. Уткина,
Россия, Рязань, anuta201294@yandex.ru*

Аннотация. Обнаружение микродвижений лица — это новая и сложная область исследований в области компьютерного зрения, которая возникла на основе работы, проделанной психологами, изучающими микромимику. Микромимикане настолько прямолинейна, чтобы ее можно было интерпретировать как эмоцию, и требует контекста, когда произошло движение, чтобы понять, можно ли классифицировать движение как микро или макровыражение. Процесс обнаружения выражений лица в компьютерном зрении обычно включает в себя предварительную обработку, извлечение признаков и классификацию. В данной статье рассмотрены и проанализированы принципы работы одного из самых эффективных алгоритмов извлечения признаков – алгоритм вычисления дескриптора локальных бинарных шаблонов по трем ортогональным плоскостям.

Ключевые слова: Микровыражение, движение, локальный бинарный шаблон, LBP, LBP-TOP.

ALGORITHM FOR COMPUTING A DESCRIPTOR OF LOCAL BINARY PATTERNS ON THREE ORTHOGONAL PLANES FOR RECOGNITION OF MICRO-FACIAL MOVEMENTS

A.D. Chernenko

*Ryazan State Radio Engineering University named after V.F. Utkin,
Russia, Ryazan, anuta201294@yandex.ru*

Annotation. Micro-facial movement detection is a new and challenging field of research in computer vision that has been inspired by the work done by psychologists studying micromimicry. Micromimicry is not so straightforward as to be interpreted as an emotion, and requires context when the movement occurred in order to understand whether the movement can be classified as micro or macro. The process of detecting normal facial expressions in computer vision typically involves

pre-processing, feature extraction, and classification. In this paper the principle of operation of one of the most efficient feature extraction algorithms, i.e. the algorithm for calculating the descriptor of local binary templates in three orthogonal planes, are considered and analyzed.

Keywords: Microexpression, movement, local binary pattern, LBP, LBP-TOP.

1. Введение

Научно обосновано, что выражения человеческого лица носят типичный характер, а не связаны с культурой или другими особенностями проживания людей [1]. Доказано, что во время эмоционального эпизода возникает импульс, который не может сознательно контролироваться человеком. Под его влиянием формируется одно из 7 выражений лица, соответствующее эмоциям счастья, грусти, гнева, страха, удивления, отвращения или презрения. Если человек осознает влияние импульса он может попытаться подавить внешнее выражение, однако скрыть появляющиеся микродвижения мимических мышц полностью не представляется возможным. При проведении видеозаписи со скоростью 25 кадров в секунду (*fps*) было обнаружено, что микровыражение длится доли секунды, поэтому для его распознавания из последовательностей изображений с использованием локальных бинарных шаблонов по трем ортогональным плоскостям (*LBP-TOP*) извлекаются ключевые признаки.

2. Алгоритм вычисления дескриптора локальных бинарных шаблонов по трем ортогональным плоскостям

В большинстве методов используется изображение каждого выражения лица, которое отображает его характерные черты на пике эмоции [2-7]. Однако, по мнению психологов [8], анализ последовательности изображений дает более точное и надежное распознавание мимики. Исследования психологов показали, что изменение состояния лица является основополагающим при распознавании мимики. Эксперименты, проведенные Джоном Бассили [9], демонстрируют, что люди распознают выражения лиц по динамическим изображениям лучше, чем по фотографиям.

Обычно видеопоследовательность рассматривается как набор плоскостей XU по оси T . Однако следует отметить, что видеопоследовательность также может рассматриваться как набор плоскостей XT по оси U и плоскостей UT по оси X . При этом плоскости XT и UT предоставляют информацию о пространственно-временных переходах.

Алгоритм вычисления дескриптора локальных бинарных шаблонов по трем ортогональным плоскостям включает следующие этапы.

Этап 1. Разбиение всей исходной последовательности изображений на блоки.

Этап 2. Разбиение каждого из полученных на предыдущем этапе блоков на блоки меньшего размера.

Этап 3. Вычисление локального бинарного шаблона по каждой из плоскостей XU , XT , UT .

Этап 4. Вычисление итоговой гистограммы.

Рассмотрим подробнее выполнение каждого из этапов.

Этап 1. Разбиение всей исходной последовательности изображений на блоки (рис. 1).

Чтобы получить вхождение микрошаблонов с привязкой к их расположению, вводится представление, в котором изображение лица делится на несколько блоков. Данный блочный подход сочетает в себе характеристики уровня пикселей, уровня области и уровня объема. Каждый блок последовательности представлен тремя параметрами [H , W , T], где H – высота блока, W – ширина блока, T – номер блока в последовательности.

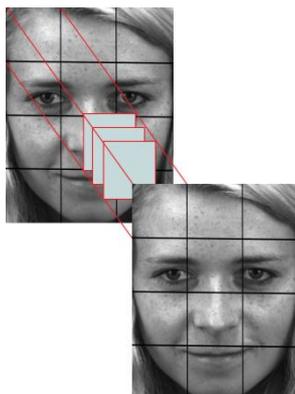


Рис. 1. Разбиение изображений последовательности на блоки

Этап 2. Разбиение каждого из полученных на предыдущем этапе блоков на блоки меньшего размера. В ходе этапа 2 блоки, полученные на этапе 1, делятся на перекрывающиеся или неперекрывающиеся блоки меньшего размера (рис. 2).

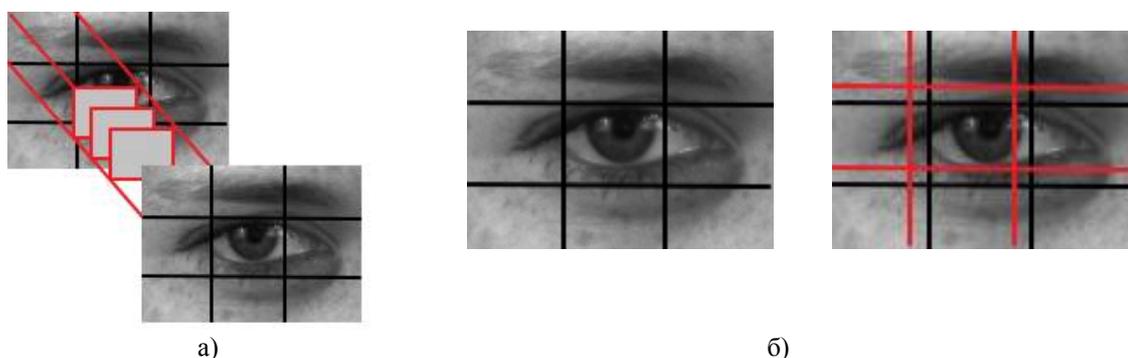


Рис. 2. Области лица: а – разбиение выделенного блока на меньшие блоки; б – разбиение блока на перекрывающиеся или неперекрывающиеся блоки

Этап 3. Вычисление локального бинарного шаблона по каждой из плоскостей XOY , XOT , YOT .

Данный этап включает следующие шаги.

Шаг 1. Преобразование блока изображения по рассматриваемой плоскости из трехмерного в двухмерный.

Шаг 2. Применение алгоритма вычисления LBP к полученному плоскому блоку.

Сначала выполняется выборка соседних точек в объеме. При этом очень важно верно выбрать траекторию расположения этих точек вокруг центрального пикселя. Так, установка радиуса по оси времени равным радиусу по пространственной оси нецелесообразна при анализе изображений выражения лица. Если разрешение изображения более чем 300 на 300 пикселей и частота кадров менее 12, то в области соседних точек, расположенных на окружности с радиусом 8 пикселей по осям X и Y , текстура может сохранять свой внешний вид. Однако в пределах одних и тех же временных интервалов по оси T текстура резко меняется, особенно, если изображения имеют высокое разрешение и низкую частоту кадров. Таким образом, в плоскостях XT и YT следует назначать разные радиусы для выборки соседних точек в пространстве и времени. При данном подходе традиционная круговая выборка расширяется до эллиптической выборки (рис. 3, 4).

В общем случае радиусы на осях X , Y и T , а также количество соседних точек в плоскостях XU , XT и YT могут быть разными, что можно отметить как R_X , R_Y , R_T и P_{XY} , P_{XT} , P_{YT} , как изображено на рис. 4. Соответствующая функция обозначается как

$LBP - TOP_{P_{XY}, P_{XT}, P_{YT}, R_X, R_Y, R_T}$. Предположим, что координаты центрального пикселя $g_{t_c, c}$ равны (x_c, y_c, t_c) , координаты $g_{XY, p}$ задаются как $(x_c - R_X \sin(\frac{2\pi p}{P_{XY}}), y_c + R_Y \cos(\frac{2\pi p}{P_{XY}}), t_c)$, координаты $g_{XT, p}$ задаются формулами $(x_c - R_X \sin(\frac{2\pi p}{P_{XT}}), y_c, t_c - R_T \cos(\frac{2\pi p}{P_{XT}}))$, а координаты $g_{YT, p}$ есть $(x_c, y_c - R_Y \cos(\frac{2\pi p}{P_{YT}}), t_c - R_T \sin(\frac{2\pi p}{P_{YT}}))$. Это отличается от обычного LBP, широко используемого во многих статьях, и расширяет его определение.

При вычислении $LBP - TOP_{P_{XY}, P_{XT}, P_{YT}, R_X, R_Y, R_T}$, учитывается только центральная часть, потому что достаточно большая окрестность не может использоваться на границах в трехмерном пространстве.

После того, как траектория расположения соседних точек выбрана, следует выполнить установку порогового значения, соответствующего значению центрального пикселя, для каждой точки в окрестности с целью получения двоичного кода. Если значение яркости соседнего пикселя больше, чем значение яркости центрального пикселя, тогда соседнему пикселю присваивается 1, иначе 0. Наконец, создается LBP код путем умножения пороговых двоичных значений на веса, присвоенные соответствующему пикселю, и путем суммирования получается результат (рис. 5).

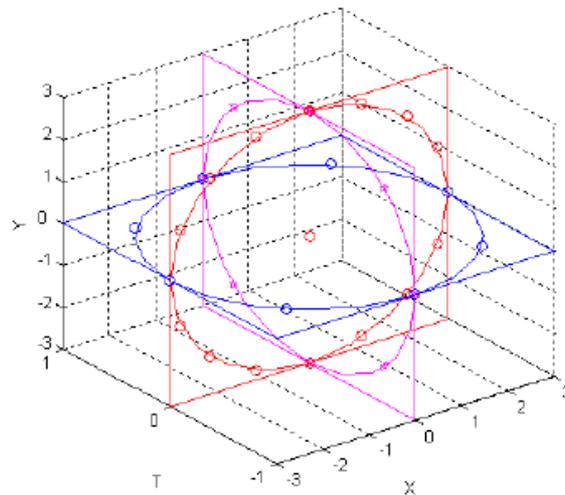


Рис. 3. Различные радиусы и количество соседних точек на трех плоскостях

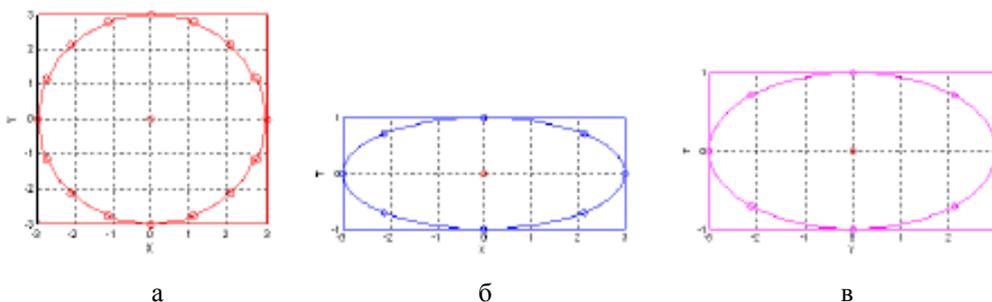


Рис. 4. Подробное представление рисунка 3, $R_X = R_Y = 3$, $R_T = 1$, $P_{XY} = 16$, $P_{XT} = P_{YT} = 8$:
а – XY плоскость; б – XT плоскость; в – YT плоскость

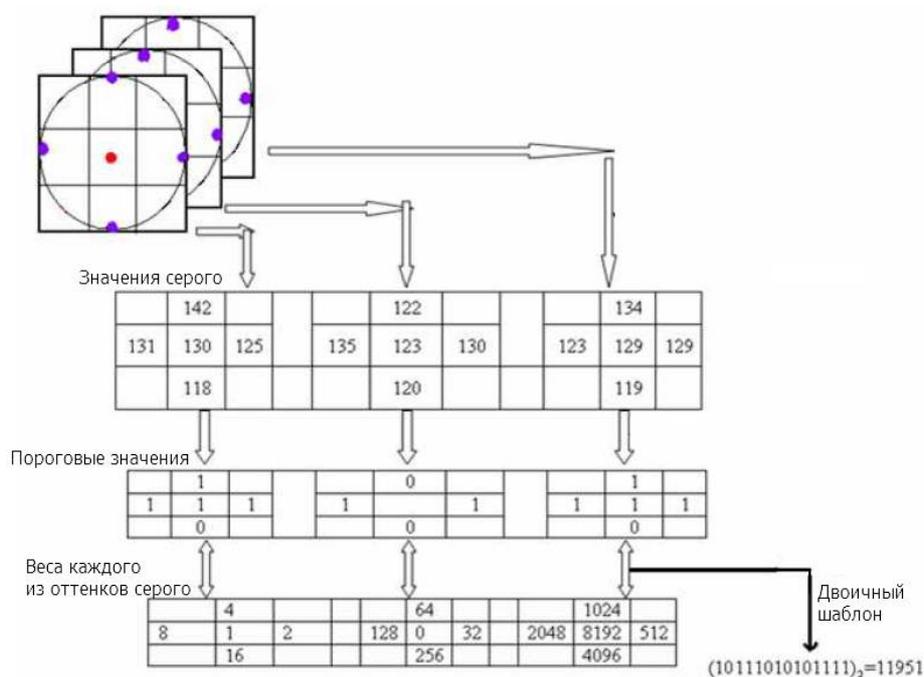


Рис. 5. Алгоритм вычисления локального бинарного шаблона

Шаг 3. Преобразование полученной гистограммы LBP из двумерной в трехмерную.

Таким образом, на основе выполнения предыдущих этапов, у нас есть описание выражения лица на трех разных уровнях. Метки (ячейки) в гистограмме содержат информацию по трем ортогональным плоскостям, описывающим внешний вид и временную информацию на уровне пикселей. Метки суммируются в небольшой блок для получения информации на региональном уровне, выражающей характеристики внешнего вида и движения в определенных местах.

Этап 3. Вычисление итоговой гистограммы.

Этап состоит из следующих шагов.

Шаг 1. Вычисление количества столбцов в итоговой гистограмме.

Шаг 2. Объединение полученных гистограмм в единую гистограмму.

На данном этапе вся информация, полученная на региональном уровне, объединяется, путем объединения гистограмм по каждой из плоскостей XY , XT и YT , которые обозначаются как $XY-LBP$, $XT-LBP$ и $YT-LBP$, и получается статистика по трем разным плоскостям для построения глобального описания движения и выражения лица (рис. 6) [9].

Гистограмма может быть выражена следующим образом:

$$H_{i,j} = \sum_{x,y,t} I\{f_j(x,y,t) = i\}, i = 0, \dots, n_j - 1; j = 0, 1, 2 \quad (1),$$

где n_j – количество разных отметок, созданных функцией LBP на j плоскости, где $j = 0$ для XY , $j = 1$ для XT , $j = 2$ для YT ; $f_j(x,y,t)$ выражает LBP код центрального пикселя на j -й плоскости.

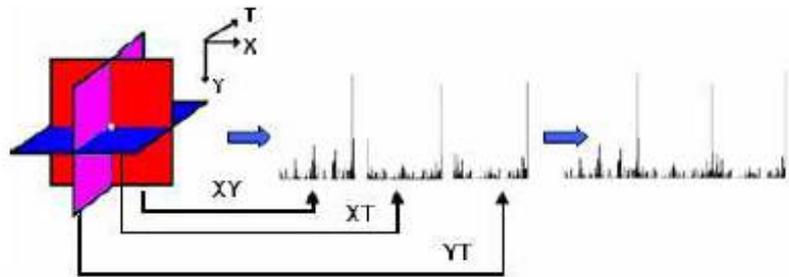


Рис. 6. Получение статистики по трем плоскостям и объединение в единую гистограмму

3. Заключение

В данной статье рассмотрен алгоритм объединенных гистограмм, вычисленных по трем ортогональным плоскостям, который упрощает извлечение признаков совместной встречаемости при большом числе соседних точек. Данный алгоритм прост в вычислительном отношении и надежен с точки зрения вариаций оттенков серого и поворота, что делает его очень многообещающим для реальных прикладных задач. Обработка выполняется локально, захватывая информацию о переходе, что означает, что алгоритм также может использоваться для сегментирования динамических изображений лица. Таким образом, можно избежать проблем, связанных с последовательностями, содержащими более одной динамически изменяющейся области изображения. Также данный алгоритм устойчив к ошибкам в выравнивании лица и не требует подверженной ошибкам сегментации областей лица, например губ. Кроме того, перед выполнением этапов, представленных в статье, для изображений с лицами не требуется нормализация шкалы серого.

Библиографический список

1. D. Chetverikov and R. Preteri, "A Brief Survey of Dynamic Texture Description and Recognition," Proc. Int'l Conf. Computer Recognition Systems, pp. 17-26, 2005.
2. X. Feng, M. Pietikainen, and A. Hadid, "Facial Expression Recognition with Local Binary Patterns and Linear Programming," Pattern Recognition and Image Analysis, vol. 15, no. 2, pp. 546-548, 2005.
3. C. Shan, S. Gong and P.W. McOwan, "Robust Facial Expression Recognition Using Local Binary Patterns," Proc. IEEE Int'l Conf. Image Processing, pp. 370-373, 2005.
4. G. Littlewort, M. Bartlett, I. Fasel, J. Susskind, and J. Movellan, "Dynamics of Facial Expression Extracted Automatically from Video," Proc. IEEE Workshop Face Processing in Video, pp.80-88, 2004.
5. M. Pantic and L.L.M. Rothkrantz, "Automatic Analysis of Facial Expressions: The State of the Art," IEEE Trans. Pattern Analysis and Machine Intelligence, vol. 22, no.12, pp. 1424-1455, 2000.
6. B. Fasel and J. Luetin, "Automatic Facial Expression Analysis: A Survey," Pattern Recognition, vol. 36, pp. 259-275, 2003.
7. C. Shan, S. Gong and P.W. McOwan, "Robust Facial Expression Recognition Using Local Binary Patterns," Proc. IEEE Int'l Conf. Image Processing, pp. 370-373, 2005.
8. J. Bassili, "Emotion Recognition: The Role of Facial Movement and the Relative Importance of Upper and Lower Areas of the Face," Journal of Personality and Social Psychology, vol. 37, pp. 2049-2059, 1979.
9. Guoying Zhao and Matti Pietikainen, "Dynamic texture recognition using local binary patterns with an application to facial expressions," IEEE Trans. Pattern Analysis and Machine Intelligence, vol. 14, pp. 915-928, 2007.

УДК 004.93'12; ГРНТИ 28.23.37

ПРИНЦИП РАБОТЫ АЛГОРИТМА ТРЕКИНГА ДЛЯ ЗАДАЧИ РАСПОЗНАВАНИЯ ЛЮДЕЙ В ТРАНСПОРТНЫХ СРЕДСТВАХ

Е.Р. Муратов, А.С. Епифанов

*Рязанский государственный радиотехнический университет имени В.Ф. Уткина,
Российская Федерация, Рязань, alexlaw00@mail.ru*

Аннотация. В работе рассматриваются принципы, лежащие в основе алгоритма трекинга. Приводятся их основные особенности и параметры существующих алгоритмов, которых можно использовать для реализации различных этапов трекинга объектов.

Ключевые слова: компьютерное зрение, машинное обучение, детектирование, трекинг, ReID, YOLO, SORT, MOT.

OPERATING PRINCIPLE OF THE TRACKING ALGORITHM FOR THE PROBLEM OF RECOGNITION OF PEOPLE IN VEHICLES

E.R. Muratov, A.S. Epifanov

*Ryazan State Radio Engineering University named after V.F. Utkin,
Russia, Ryazan, alexlaw00@mail.ru*

The summary. The paper investigates the considerations underlying the tracking algorithm. By using their main features and the parameters of existing algorithms, they can be used to implement various targets.

Keywords: computer vision, machine learning, detection, tracking, ReID, YOLO, SORT, MOT.

Задача подсчёта числа людей в транспортных средствах является нетривиальной задачей по сбору различной бизнес-статистики. Сложность такой задачи заключается в однозначном определении человека в кадре, притом необходимо учитывать, что процесс проезда человека может занимать какое-то время и, как следствие, придётся увеличить устойчивость всей системы относительно временной компоненты. Одним из перспективных решений такой проблемы является использование алгоритмов трекинга, способных детектировать видеопоток и составлять трекетты (последовательности боксов) для различных объектов во времени, что повысит надёжность системы. В данной статье рассматриваются принципы, на которых базируются алгоритмы трекинга.

Алгоритм трекинга в своей основе содержит несколько других алгоритмов, взаимная комбинация между которыми определяет основные параметры построения трека (скорость, точность). Рассмотрим каждый из этих базовых алгоритмов.

Алгоритмы детекции объектов в кадре

Любой алгоритм трекинга начинается с детектирования объектов в кадре. Результатом детектирования является фиксированное положение объекта. Данные алгоритмы очень чувствительны к ракурсам камеры, яркости, каким-либо помехам (снег, дождь) и т.п., что в конечном счёте может сказываться на качестве всего трекинга. Создание правильных условий для процесса детектирования может в несколько раз повысить точность работы всей дальнейшей системы.

Существует множество алгоритмов детектирования, в зависимости от рассматриваемых объектов или результирующих параметров полученной системы трекинга. Например, если за основу трекинга мы возьмём детектирование лица человека, то можно использовать одну из следующих категорий алгоритмов: эмпирические (метод гистограмм); алгоритмы, использующие характерные инвариантные признаки (метод группировки признаков); алгоритмы, использующие шаблоны разработчика (метод детектирования лица при помощи трехмерных форм) или алгоритмы, использующие обучающиеся системы (РСА, нейронные сети, SVM, ААМ и т.д.) [1]. У каждого из этих подходов свои преимущества и недостатки,

своя скорость и особенности, которые следует учитывать при построении полноценной системы трекинга.

В настоящее время одними из наиболее сбалансированных решений для задачи детектирования объектов являются нейронные сети YOLOv3- tiny, YOLOv4 или EfficientDet [2,3] (рисунок 1).

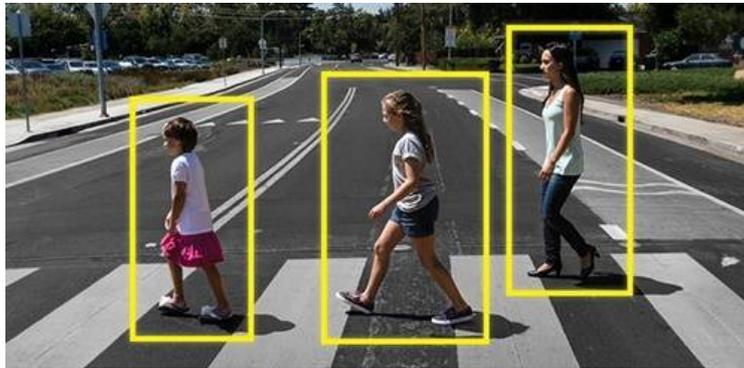


Рис. 1. Пример работы детектора YOLO

Алгоритмы распознавания объекта в кадре

Для алгоритмов трекинга данный этап также называется задачей реидентификации (ReID), поскольку рассмотренный на предыдущем кадре объект должен быть правильно идентифицирован на следующем кадре.

Распознавать объект можно по различным отличительным признакам. Например, распознавание человека можно производить по лицу, по телу или по одежде (рисунок 2). Распознавание автомобиля можно по его типу или номеру. Распознанные объекты можно преобразовать в хэш-код (дескриптор), который затем используют для повторной идентификации (реидентификации) объекта. Для алгоритмов трекинга это особенно полезно, если произошёл какой-либо сбой и трек прервался: механизм хэширования позволит быстрее восстановить работоспособность.

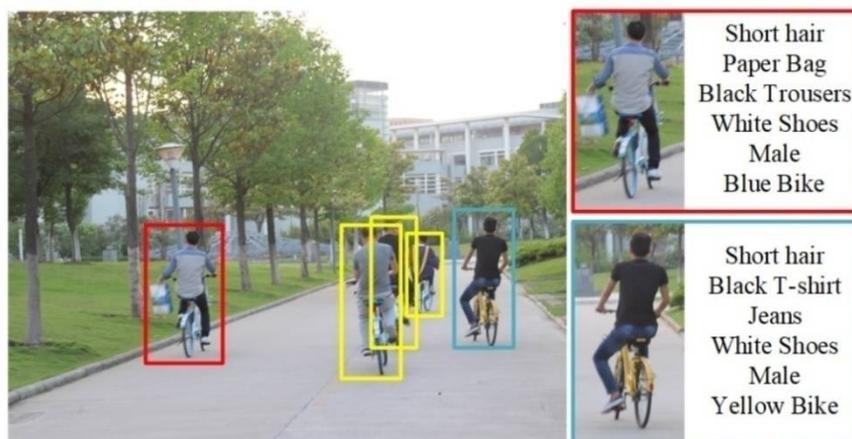


Рис. 2. Распознавание людей по одежде

Для распознавания также используются различные алгоритмы, в зависимости от специфики объекта. Если удастся сделать качественную систему трекинга, основанную на распознавании лица, то её точность будет очень высокой. Однако на практике такого достичь очень сложно, и распознавание лица выступает одной из дополнительных метрик, повышающих общее качество системы. Добиться точности в 100% для алгоритмов ReID невоз-

можно даже в настоящих системах. Исследования показывают, что максимальный уровень точности, на которой приходится рассчитывать, это 94-98% [4].

Конкретно для задачи распознавания людей в транспортных средствах в рамках алгоритма трекинга можно совместить определение по лицу и определение по одежде. Методов получения дескриптора лица множество. Классические из них: метод гибкого сравнения на графах (Elastic graph matching), нейронные сети, скрытые марковские модели (СММ, НММ), метод главных компонент или principal component analysis (PCA), Active Appearance Models (AAM), Active Shape Models (ASM) и т.д. Для того, чтобы распознать человека по одежде, можно использовать следующие подходы: нейронные сети с глубоким обучением (AlexNet, VGG, GoogleNet, Capsule Network и т.д.), рекуррентные и рекурсивные нейронные сети (RNN и RvNN), PAR-алгоритмы (ACN, DeepSAR, DeepMAR, MTCNN), алгоритмы обучения на основе учебной программы (MTCT, CILICIA), алгоритмы на основе графической модели (DCSA, A-AOG, VSGR) [5] и т.п. Каждый из этих методов обладает своими особенностями реализации, преимуществами и недостатками, которые необходимо учитывать при построении общей системы.

Алгоритмы сопоставления объектов по физическим параметрам

В любом случае задача реидентификации даст на выходе набор определённых параметров, выделяемых тем или иным способом. Но эти параметры не являются исчерпывающими для полноценного трекинга. Можно также определять характеристики, которые сложны для выделения нейросетью. К таким характеристикам относятся: положение центра масс (рассматриваемые объекты идентичны, если центр масс будет ближе к планируемой траектории), близость позы (как правило, ReID алгоритмы не учитывают позу человека, но это также может оказаться хорошим показателем), скорость объекта, размеры объекта (выделенного бокса, оценивают идентичность размеров объекта в соседних кадрах) и т.д.

Задачу сопоставления боксов объектов можно решить на основе геометрических признаков (метод IoU), для которых боксы сопоставляются по наибольшей области пересечения с двух соседних кадров (рисунок 3). Такой подход не является устойчивым в случае ложных перекрытий боксов посторонними объектами. Поэтому к данному подходу можно добавить Венгерский алгоритм для назначения детекций и фильтры Калмана для определения скорости. Однако можно разработать и более сложные системы, которые сопоставляют по скелету (учитывая насколько они совпадают по внешней форме), по положению на земле или на полу.

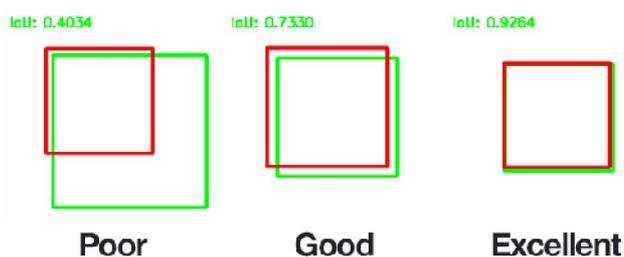


Рис. 3. Сопоставления боксов объектов по методу IoU

Алгоритмы межкадрового трекинга

В данную категорию принято относить алгоритмы оптического трекинга. Представителями такого подхода являются Predator tracker, KCF трекер, нейронные сети с глубоким обучением GOTURN, некоторые сиамские сети (SiamMask_E, SiamFC++, SiamRPN++) [6], разные LSTM-сети с длинной памятью (THOR). Однако оптический трекинг выбирает оптимальный минимум по метрике близости, не учитывая скорость и направление, что может быть критичным в ряде задач.

На настоящий момент хорошо распространено использование алгоритма трекинга SORT [7] и его модификации DeepSORT [8], использующей нейронную сеть для реидентификации объектов. Данные алгоритмы включают в себя весь перечень рассмотренных выше этапов, реализуя мощные системы трекинга, работающие в 20 раз быстрее современных аналогов. В основе алгоритма SORT лежат фильтры Калмана и Венгерский алгоритм, что делает его простым и эффективным в своей реализации. На некоторых наборах данных он работает даже быстрее своего последователя – DeepSORT.

Ещё одним успешным трекером можно считать FairMOT, который перенимает логику алгоритмов SORT и улучшает её за счёт более тонкого этапа обучения, основанного на архитектуре обнаружения объектов без привязки CenterNet [9]. Это позволяет трекеру FairMOT более эффективно решать задачу отслеживания множества объектов. Именно этот трекер по последним данным считается одним из лучших трекеров по конкурсу MOT с открытым кодом.

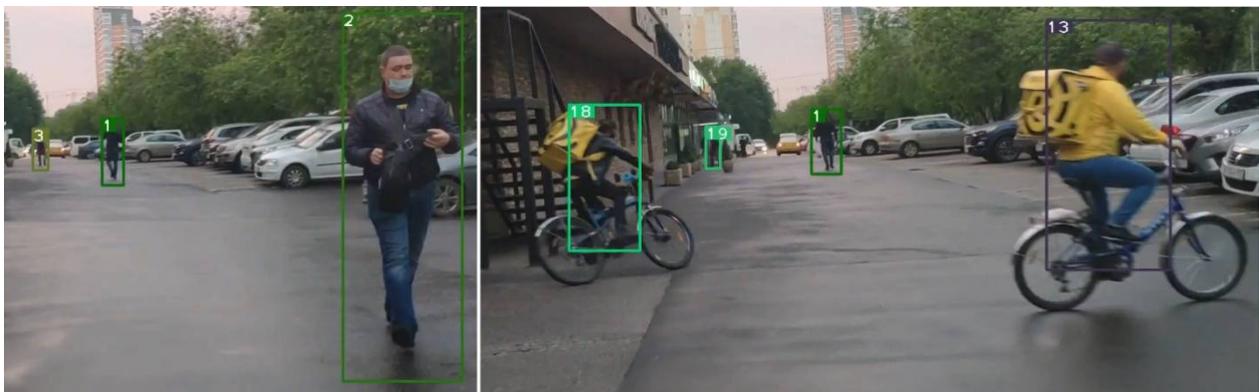


Рис. 4. Пример работы трекера SORT

Для задачи распознавания людей в транспортных средствах использование алгоритма трекинга может повысить качество и надёжность системы, в которой можно использовать как готовые трекаеры, так и собрать свои собственные на основе рассмотренных алгоритмов. Ключевым моментом будет являться оптимальное расположение камер в салоне транспортного средства, от которого напрямую зависит качество алгоритмов трекинга.

Библиографический список

1. Татаренков, Д. А. Анализ методов обнаружения лиц на изображении / Д. А. Татаренков. — Текст : непосредственный // Молодой ученый. [Электронный ресурс] — 2015. — № 4 (84). — С. 270-276. — URL: <https://moluch.ru/archive/84/15524/> (дата обращения: 24.01.2022).
2. You Only Look Once: Unified, Real-Time Object Detection / J. Redmon [et al.] // arXiv:1506.02640v5 [cs.CV], 2016.
3. EfficientDet: Scalable and Efficient Object Detection / Mingxing Tan and Ruoming Pang and Quoc V. Le// arXiv: 1911.09070 [cs.CV], 2020.
4. Spatial-Temporal Person Re-identification / Guangcong Wang and Jianhuang Lai and Peigen Huang and Xiaohua Xie// arXiv: 1812.03282 [cs.CV], 2018.
5. Pedestrian Attribute Recognition: A Survey / Xiao Wang, Shaofei Zheng, Rui Yang, Bin Luo, Jin Tang // arXiv:1901.07474v1 [cs.CV] 2019.
6. Visual Object Tracking on VOT2017/18/ [Электронный ресурс] URL: <https://paperswithcode.com/sota/visual-object-tracking-on-vot201718> (дата обращения: 24.01.2022).
7. Simple Online and Realtime Tracking / A. Bewley [et al.] // arXiv:1602.00763v2 [cs.CV], 2017.
8. Simple Online and Realtime Tracking with a Deep Association Metric / N. Wojke [et al.] // arXiv: 1703.07402v1 [cs.CV], 2017.
9. FairMOT: On the Fairness of Detection and Re-identification in Multiple Object Tracking/ Zhang [et al.] // International Journal of Computer Vision, 2021.

УДК 621.397; ГРНТИ 50.39.27

РЕАЛИЗАЦИЯ ДВУХПРОВОДНОГО ПРОТОКОЛА ПЕРЕДАЧИ ДАННЫХ НА VERILOG

М.Б. Никифоров, Н.В. Калинин

*Рязанский государственный радиотехнический университет имени В.Ф. Уткина,
Российская Федерация, Рязань, nikit.kalinin.2000@mail.ru*

Аннотация. В данной статье рассматривается разработка I2C-подобного протокола передачи данных на языке описания аппаратуры Verilog на одноплатном компьютере MYS-7Z020-C в среде Vivado.

Ключевые слова: ПЛИС, протокол передачи данных.

IMPLEMENTATION OF A TWO-WIRE DATA TRANSFER PROTOCOL ON VERILOG

M.B. Nikiforov, N.V. Kalinin

*Ryazan State Radio Engineering University named after V.F. Utkin,
Russia, Ryazan, nikit.kalinin.2000@mail.ru*

The summary. This article discusses the development of an I2C-like data transfer protocol in the Verilog hardware description language on a single-board computer MYS-7Z020-C in a Vivado environment.

Keywords: FPGA, data transfer protocol

На данный момент в микропроцессорной технике существует множество протоколов передачи данных, но одним из самых популярных остается протокол I2C, разработанный компанией Philips. С помощью данного интерфейса осуществляется обмен данными между периферийными устройствами различных микроконтроллеров и интегральных схем.

Общий принцип работы протокола

Шина представлена двумя проводами – шина данных SDA и тактовый сигнал SCL. Эти шины всегда подтянуты вверх (то есть логический уровень на них в не состоянии равен 1, а не 0). Прием или передача сигналов реализуется прижиманием линии в 0. Как правило, применяются резисторы с сопротивлением равным 10 кОм. Допустимо и применение резисторов с большим сопротивлением, однако следует учесть, что линия будет дольше восстанавливается в единицу, что в свою очередь снижает скорость передачи данных. Скорость передачи данных в таком интерфейсе равна от 10кбит/с до 100кбит/с.

На рисунке 1 представлена схема подключения устройств к интерфейсу. Устройства делятся на два типа:

1. Ведущий (master) – генерирует тактовый сигнал и управляет процессом передачи данных. Всегда выступает инициатором обмена.
2. Ведомый (slave) – может откликаться на сигналы ведомого. Всего устройств может быть 127.

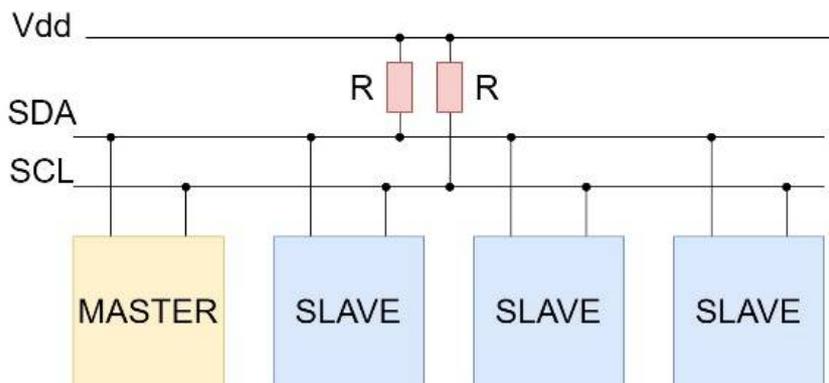


Рис. 1. Схема подключения устройств по интерфейсу

Вся передача состоит из (рис. 2):

1. START сигнала – спад SDA при SCL = 1.
2. Посылки 8 битов от ведущего к ведомому или приема ведущим от ведомого.
3. Сигнала подтверждения А (или ACK). 0 – ответ, 1 – нет ответа.
4. STOP сигнала – фронт SDA при SCL = 1.

Начало работы начинается с формирования START-сигнала. Далее если данные идут от ведущего к ведомому, то последний генерирует тактовый сигнал на SCL и выдает биты на SDA. На каждом фронте SCL ведомый считывает значение. В случае передачи от ведомого к ведущему, ведущий генерирует такты на SCL, при этом прослушивая линию SDA — считывает данные. Если SCL уходит в 0, то ведомый выставляет на SDA бит, который мастер считывает, когда поднимет SCL обратно. Передача заканчивается все STOP последовательностью.

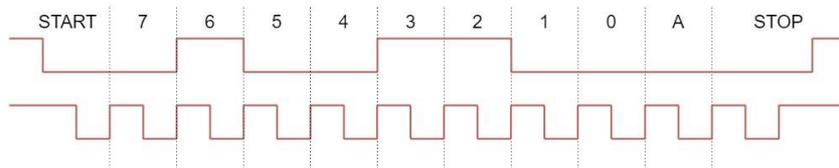


Рис. 2 Пример передачи сигналов по шинам

В интерфейсе есть 2 типа операции – чтение и запись. Запись подразумевает передачу вначале адреса устройства, адреса ячейки памяти и самих данных (рис. 3).

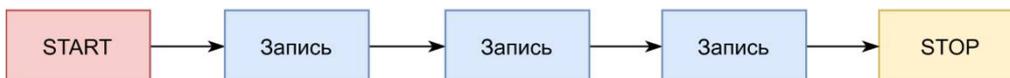


Рис. 3. Операция записи

Чтение реализуется несколько иначе – сначала передается адрес устройства и адрес ячейки памяти в нем, потом по SDA подается повторный START сигнал. Ведомое в это время понимает, что ведущее ожидает от него данные. Далее по SDA передается адрес устройства, после чего ведомое прослушивает линию SDA.



Рис. 3. Операция записи

Реализация интерфейса на языке Verilog

Интерфейс был реализован по схеме автомата – есть несколько связанных состояний, которые выполняют определенную функцию (рис. 4). Основная функция именуется как «Хаб» - она определяет, какая операция будет выполнена следующей, и проводит подготовку данных перед операцией.

Функция «Запись» опускает линию SDA в 0 (1 образуется сама из-за подтягивающего резистора) в случае, если требуется передать по интерфейсу 0. «Чтение» прослушивает линию данных и сохраняет биты SDA. После каждой «Записи» или «Чтения» один раз срабатывает функция «Холостой такт» на линии SCL – этот такт дает возможность ведомому ответить.

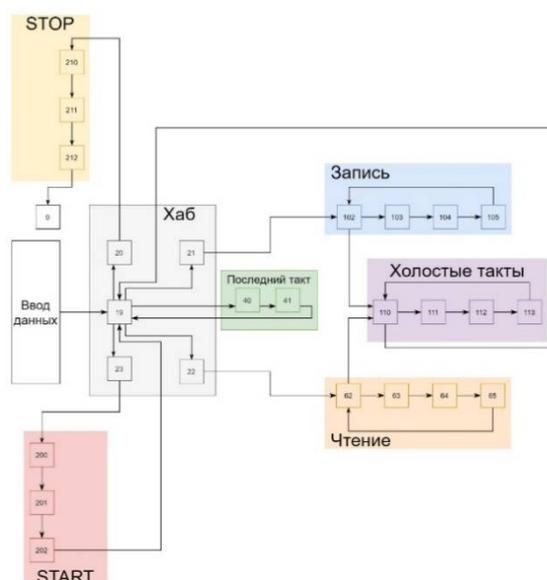


Рис. 4. Схема автомата

Во время реализации возникла одна интересная особенность – на ПЛИС сигнал STOP нельзя подавать сразу после ACK - в конце операции нужно добавить еще один холостой такт (рис. 5). В противном случае периферийное устройство ответит (то есть ACK = 0), но линию SDA ведомый не отпустит, то есть что бы не делали другие устройства, 0 останется на линии SDA до тех пор, пока ведомый не получит еще 1 такт. Таким образом линию SDA можно заблокировать.

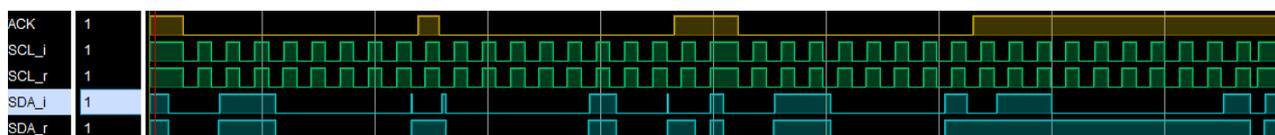


Рис. 5. Временные диаграммы операции чтения

В данном случае был опрошен датчик температуры, встроенный в одноплатный компьютер MYS-7Z020-C. На временной диаграмме видно, как устройство отвечает по шине (придавливание шины SDA_i в 0, в то время как шина SDA_r (шина SDA на резисторе) подтянута в 0).

Библиографический список

1. Новиков А.И., Муратов Е.Р., Никифоров М.Б., Колчаев Д.А. Mathematical modelling of an airborne technical vision system development// E3S Web of Conferences 224, Topical Problems of Agriculture, Civil and Environmental Engineering (TPACEE 2020). Москва. Volume 224, 2020 P. 147-153.
2. Хлуденев А.В. Встроенный логический анализатор на ПЛИС // НТК Оренбург, 2010, С. 261-265
3. Троицкий А.М. Методы тестирования и отладки ПЛИС инструменты проектирования ПЛИС // RESEARCH SUCCESS 2021. Нижний Новгород, 2021, С 36-44.
4. Беляков, П.В. Методы оптимизации в задаче вычисления оптического потока в системе на кристалле / П.В. Беляков, М.Б. Никифоров // Известия Тульского государственного университета. Технические науки. – 2019. – Вып. 2. – С. 45-52.
5. Братулин А.В. Метод вычисления плотного оптического потока на ПЛИС в реальном времени /А.В. Братулин, М.Б. Никифоров, П.В. Беляков, Е.Ю. Холопов // Современные информационные технологии и ИТ-образование. – 2019. – Т. 15. - № 2. – С 320-330.

УДК 62-835:621.316.71

СТЕНД ДЛЯ ПОСТРОЕНИЯ УСТРОЙСТВ НА МИКРОПРОЦЕССОРЕ С ИСПОЛЬЗОВАНИЕМ ПЛИС

А.Ю. Лаппо

*Тульский государственный университет,
Российская Федерация, Тула, strcorp@mail.ru*

Аннотация. Рассматривается лабораторный стенд, позволяющий обучающимся исследовать микропроцессорные устройства и решать задачи, связанные с дисциплиной «Цифровые устройства и микропроцессоры».

Ключевые слова: микропроцессор, адресное пространство, ПЛИС, обучающий стенд.

STAND FOR BUILDING MICROPROCESSOR DEVICES USING CPLD

A.U. Lappo

*Tula State University,
Russian Federation, Tula, strcorp@mail.ru*

Annotation. There is a laboratory stand that allows students to study microprocessor devices and solve problems related to the discipline "Digital devices and microprocessors".

Keywords: microprocessor, address space, CPLD, training stand.

На кафедре Радиоэлектроники присутствует дисциплина «Цифровые устройства и микропроцессоры». Часть этой дисциплины отводится под изучение микропроцессоров и микропроцессорных устройств. К сожалению, теоретическое изучение микропроцессоров очень сложно даётся студентам, поэтому необходимы способы изучения микропроцессоров на практике.

Существует много обучающих микропроцессорных устройств для обучения, в том числе выпускаемые на заводах в РФ. В основе этих устройств легендарные процессоры Z80, Intel 8080, Intel 8086 и т.д. а также их советские аналоги (КР1858ВМ1 и т.д.). Эти системы построены на элементной базе того времени и разрабатывались для использования без ПК. К недостаткам этих систем стоит отнести:

- Для систем без ПЗУ - необходимость ввода программы в машинных кодах
- Для систем с ПЗУ необходимость извлекать её для процесса прошивки
- Фиксированная архитектура и адресное пространство

Современные системы обучения заточены под программирование микроконтроллеров. Самая распространённая из них – Arduino. Это электронный конструктор и удобная платформа быстрой разработки электронных устройств для новичков и профессионалов. Платформа пользуется огромной популярностью во всем мире благодаря удобству и просто-

те языка программирования, а также открытой архитектуре и программному коду. Устройство программируется через USB без использования программаторов.

Платформа Arduino используется практически во всех учебных заведениях, как самая простая, однако её использование позволяет абстрагироваться от проблем проектирования микропроцессорных систем, что уже является недостатком при изучении именно микропроцессоров. Поэтому и было решено разработать такую систему, которая позволяла бы собрать достоинства платформы Arduino, но при этом позволяла бы озадачить студента по проектированию микропроцессорных устройств:

- Дано устройство с определённым адресным пространством памяти программ, памяти данных и периферией – разработать программу.
- Дана программа, составленная для устройства с определённым адресным пространством. Спроектировать схему соединения процессора со всеми устройствами, чтобы эта программа была работоспособна.

Прежде, чем проектировать стенд были рассмотрены существующие микропроцессорные системы на основе процессоров Z80, Intel8080 и Intel8086 (рис.1). Как видно из схем, все процессоры требуют наличие источника тактирования, причём для процессоров Intel он не такой простой, как для Z80 и для него использована отдельная микросхема. Ещё одно важное отличие – процессор Intel8086 имеет мультиплексированную шину адреса и данных, что требует дополнительные регистры для их разделения. К тому же, распиновка у этих процессоров совершенно разная.

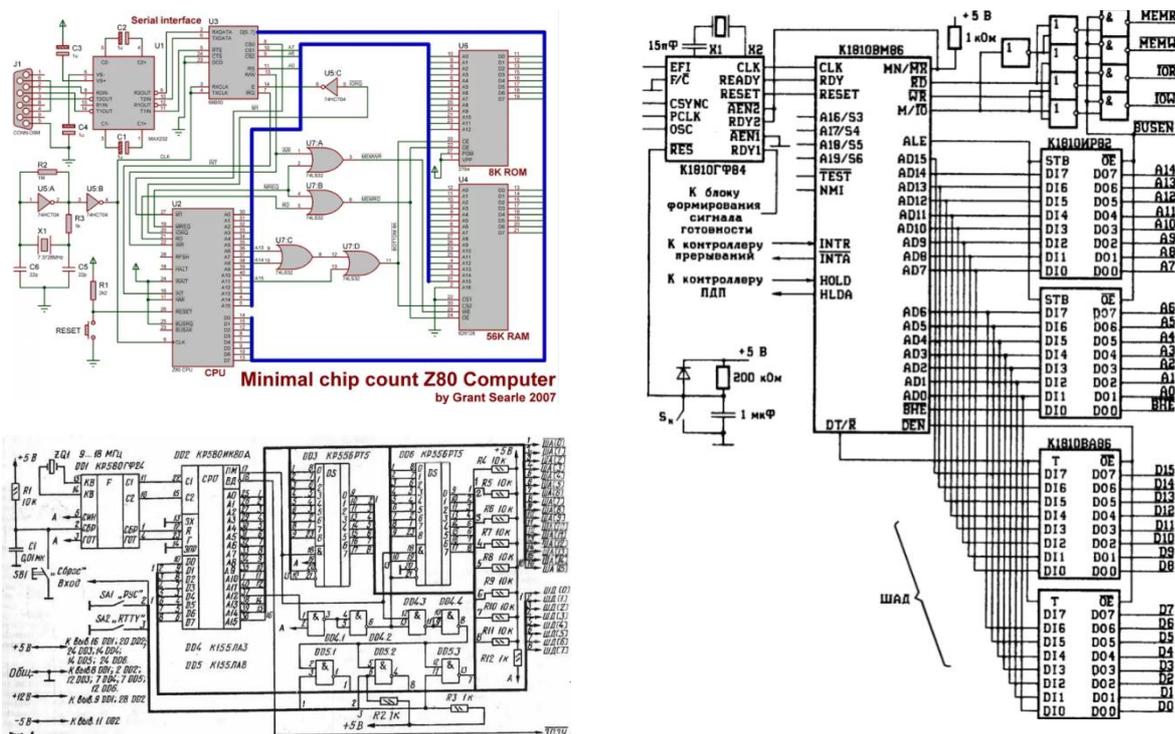


Рис. 1. Схемы (фрагменты) микропроцессорных модулей на основе процессоров Z80, Intel8080 и Intel8086

Для того, чтобы решить эти задачи, предлагается система, структурная схема которой представлена на рисунке 2. В основе системы лежит микросхема ПЛИС. В её задачу входит формирование схемы связей микропроцессора со всеми периферийными устройствами. Так

как было принято решение сделать плату универсальной для любого процессора (в пределах корпуса DIP40) все контакты микропроцессора подключаются в ПЛИС. Статическое ОЗУ и ПЗУ всеми своими сигнальными линиями подключается к ПЛИС. Программирование ПЛИС осуществляется через стандартный JTAG интерфейс. Программирование Flash ПЗУ осуществляется либо при помощи того же JTAG интерфейса через ПЛИС, либо через схему, сформированную внутри ПЛИС.

В качестве периферийных устройств предлагается использовать светодиоды, кнопки и распространённый 2-строчный дисплей 1602. Чтобы было возможно использовать готовые платы периферийных устройств Arduino, сформирован разъем Arduino. Некоторые контакты этого разъёма должны быть аналоговыми, поэтому был добавлен модуль АЦП с возможностью выбора каналов (как в микроконтроллере платы Arduino).

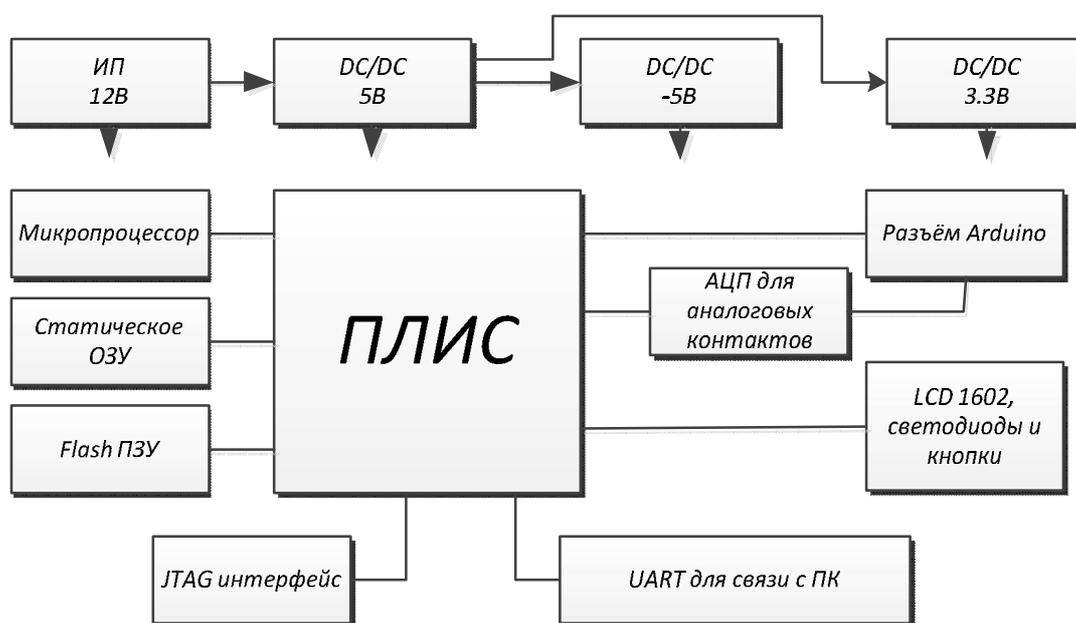


Рис. 2. Структурная схема стенда

Питание платы осуществляется при помощи источника питания 12В. Для работы всей цифровой части необходимо 5В, которые формируются из этих 12В. Для разъёма Arduino один из контактов должен давать 3.3В питания – получается от 5В линии питания. Некоторые микропроцессоры для работы требуют 12В и -5В (Intel 8080, Intel 8086). Именно для этого был выбран источник питания 12В и установлен DC/DC преобразователь на отрицательное напряжение (-5В).

К сожалению, вопросы коммутации микропроцессора с цепями питания ПЛИС не может взять на себя из-за ограничения по напряжению и току, поэтому предполагается, что пользователь должен сам выполнить механическую коммутацию линий питания процессора согласно таблице. Учитывая, что в качестве пользователя будет работать студент, необходимо выполнить необходимые мероприятия, чтобы защитить схему от неправильной конфигурации питания (таблица 1).

Таблица 1. Конфигурация линий питания

Линии питания	Z80	Intel8080	Intel8086
+12В	-	28	-
+5В	10	20	40
Общий	30	2	1, 20
-5В	-	11	-

Ещё одна не менее важная проблема – большинство современных ПЛИС имеют питание 3.3В, поэтому необходимо выполнять согласование 5В логических уровней с 3.3В логическими уровнями.

Библиографический список

1. Анкудинов И.Г. Микропроцессорные системы. Архитектура и проектирование: Учеб. Пособие. – СПб.: СЗТУ, 2003. – 109 с.
2. Grant's 7-chip Z80 computer. URL: <https://web.archive.org/web/20190713224608/http://searle.hostei.com/grant/z80/SimpleZ80.html>. Дата обращения: 20.02.2020
3. КР580ИК80А в любительском дисплее. URL: <https://www.rlocman.ru/shem/schematics.html?di=50300>. Дата обращения: 20.02.2020.
4. Пример схемы на Intel8086. URL: <https://studfile.net/preview/7075014/page:13/> Дата обращения: 20.02.2020

УДК 004.42 ; ГРНТИ 50.41.29

РАЗРАБОТКА ТЕСТОВОГО СТЕНДА ДЛЯ СРАВНЕНИЯ АЛГОРИТМОВ ДЕТЕКТИРОВАНИЯ ГРАНИЦ

А.А. Лучкина, В.Ю. Тарасова

Рязанский государственный радиотехнический университет

имени В.Ф. Уткина,

Россия, Рязань, Valentina2008.91@mail.ru

Аннотация. Рассмотрены основные методы выделения границ на изображении и разработан тестовый стенд, позволяющий увидеть результат применения данных методов. Данный программный стенд является основой для будущего измерительного комплекса.

Ключевые слова: детектирование границ, фильтрация изображения, детектор Кэнни, оператор Собеля

DEVELOPMENT OF A TEST BENCH FOR CAMPARING EDGE DETECTION ALGORITHMS

A.A. Luchkina, V.Yu. Tarasova

Ryazan State Radio Engineering University named after V.F. Utkin,

Russia, Ryazan, Valentina2008.91@mail.ru

Abstract. The main methods for highlighting the boundaries in the image are considered and a test stand is developed that allows you to see the result of applying these methods. This software stand is the basis for the future measuring complex.

Keywords: border detection, image filtering, Canny detector, Sobel operator

Выделение линий, необходимо для получения контуров основных объектов на изображении. Такими объектами могут выступать различные здания, дороги, реки и т. д. За счет перехода от изображения всей зоны обзора к изображениям контуров основных объектов, существенно упрощается обработка данного изображения. Появляется возможность быстрее и точнее сопоставлять изображения одного и того же участка земной поверхности и фиксировать важные изменения в зоне обзора камеры.

Математическое описание изображений

Изображение является функцией двух переменных $I(x, y)$, где x, y - это пространственные координаты, а I - яркость (интенсивность) изображения в точке с координатами (x, y) .

Если переменные x, y и I - принимают значения из конечного множества (принимают неотрицательные целочисленные значения), то говорят о цифровом изображении (ЦИ).

С математической точки зрения – ЦИ – это матрица

$$I = \begin{pmatrix} I_{11} & I_{12} & \dots & I_{1N} \\ I_{21} & I_{22} & \dots & I_{2N} \\ \vdots & \vdots & \ddots & \vdots \\ I_{M1} & I_{M2} & \dots & I_{MN} \end{pmatrix}$$

значений функции $I(x, y)$ яркости на дискретной сетке размером $M \times N$ [1].

Фильтрация изображений

Любое преобразование изображения I можно представить как действие некоторого оператора T :

$$I_{ij}' = T(I_{ij}),$$

где I_{ij} – входное изображение, I_{ij}' – выходное (преобразованное).

Размер области, к которой применяется преобразование T берут его в виде $(2k + 1) \times (2k + 1)$, т.е. с нечетным числом строк и столбцов.

Обработку изображения скользящим окном (маской, шаблоном) начинают, как правило, с левого верхнего угла с ячейки $((k + 1), (k + 1))$. Затем окно смещают на один пиксель вправо по строке, вновь выполняют преобразование и т. д [2].

Преобразование T задается матрицей весовых коэффициентов. Например, для $k = 1$ ($2k + 1 = 3$) эта матрица будет иметь следующий вид:

$$T = \begin{pmatrix} t_{-1,-1} & t_{-1,0} & t_{-1,1} \\ t_{0,-1} & t_{0,0} & t_{0,1} \\ t_{1,-1} & t_{1,0} & t_{1,1} \end{pmatrix}.$$

Значения t_{ij} элементов матрицы T , как правило, нормированы, т.е.

$$\sum_{i=-k}^k \sum_{j=-k}^k t_{ij} = 1.$$

Детектирование границ основывается на алгоритмах, выделяющих точки изображения, где наблюдается резкое изменение яркости. Для выполнения данной операции вычисляется градиент функции.

Градиент – это векторная величина, которая показывает направление наискорейшее возрастание некоторой величины. Функцией в нашем случае является двумерная функция яркости изображения. Расчет градиента для функции производится по следующей формуле:

$$\nabla = \begin{bmatrix} \frac{\partial}{\partial x} f \\ \frac{\partial}{\partial y} f \end{bmatrix}.$$

Оператор Собеля

Оператор Собеля – дискретный дифференциальный оператор, вычисляющий приближенное значение градиента яркости изображения. Результатом применения оператора Собеля является вектор градиента яркости в этой точке.

Преобразование выполняется сверткой исходного изображения с ядром 3×3 для вычисления приближенных значений производных по горизонтали и вертикали [3].

Пусть A – это исходное изображение, а G_x и G_y – два преобразованных изображения, на которых точка содержит приближенные производные по x и по y . Они вычисляются следующим образом:

$$G_x = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix} * A; \quad (1)$$

$$G_y = \begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix} * A, \quad (2)$$

где $*$ - двумерная операция свертки.

Оператор Кэнни

Оператор Кэнни является многоступенчатым алгоритмом для выделения границ на изображении. Данный детектор считается в настоящее время одним из лучших, довольно трудно найти метод, который работал бы качественнее [5].

Алгоритм Кэнни состоит из нескольких этапов:

- 1) Для начала исходное изображение переводится в полутоновое.
- 2) Вторым этапом происходит сглаживание получившегося изображения путем наложения фильтра Гаусса двумерной операцией свертки.
- 3) Далее идет поиск градиентов. Алгоритм Кэнни использует четыре фильтра для выявления горизонтальных, вертикальных и диагональных границ, поскольку контуры объектов находятся в различных направлениях.

После применения оператора обнаружения границ, получается значение для первой производной в горизонтальном и вертикальном направлениях, с помощью которых можно получить угол направления границы.

4) Следующим этапом идет подавление немаксимумов. Данная операция реализуется посредством сохранения всех локальных максимумов градиента изображения, в то время как остальные значения «зануляются».

5) Заключительный этап алгоритма – это двойная пороговая фильтрация. Идет применение порога, определяющий нахождение границы в данной точке изображения. С увеличением порога происходит игнорирование более слабых границ, т.е. чем меньше порог, тем больше будет выделяться лишние данные на изображении.

Разработка тестового стенда

Для сравнительного анализа данных методов была написана программа на языке C#, интерфейс которой представлен на рисунке 1, где расположены четыре ползунка: медианный фильтр и фильтрация по Гауссу, позволяющие сделать предварительную обработку зашумленного изображения, а также операторы Кэнни и Собеля. Для оператора Кэнни положение ползунка позволяет установить величину порога, а для остальных – размер накладываемой маски (начальный размер – 3×3).

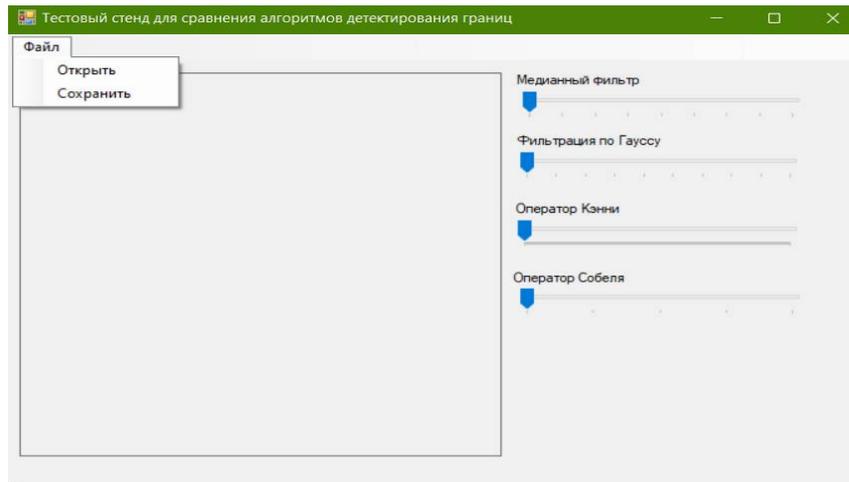


Рис.1. Интерфейс программы

Результаты работы программы

Пример работы оператора Собеля с помощью формулы (1) приведен на риунке 2.

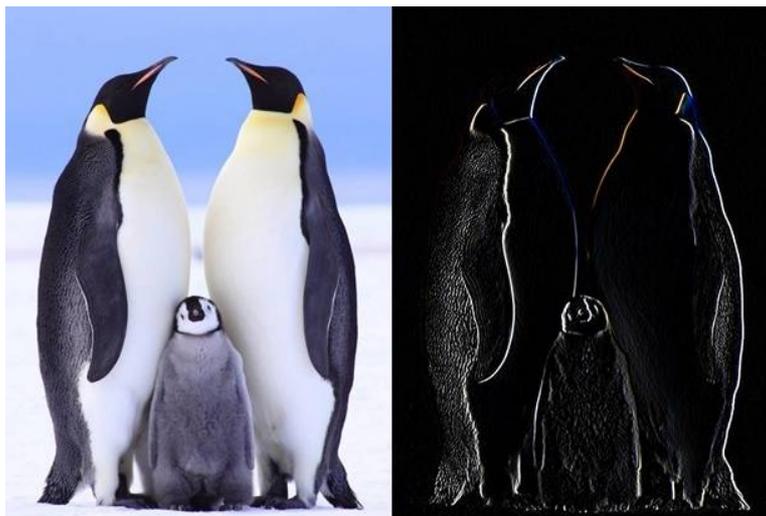


Рис. 2. Слева исходное изображение, справа преобразованное оператором Собеля, используя формулу (1)

Как видно из рисунка, более четкое выделение получили вертикальные линии, а горизонтальные либо очень слабо выделены, либо их нет совсем. Обратная ситуация получается при применении оператора Собеля уже с использованием формулы (2), где более четкое выделение будет наблюдаться у горизонтальных линий (рис. 3).



Рис. 3. Слева исходное изображение. Справа преобразованное методом Собеля, используя формулу (2)

Применение формул (1) и (2) по отдельности не позволяет достигнуть удовлетворительной точности нахождения границ на изображении. В таком случае допускается наложение масок Собеля из формул (1) и (2) последовательно на одно изображение. Пример последовательного наложения данных масок представлен на рисунке 4.



Рис. 4. Слева исходное изображение. Справа преобразованное оператором Собеля, используя наложение двух матриц

Пример работы оператора Кэнни и сравнение двух преобразованных изображений разными методами представлены на рисунках 5 и 6 соответственно.



Рис. 5. Слева исходное изображение. Справа преобразованное методом Кэнни



Рис. 6. Слева применение оператора Собеля. Справа применение оператора Кэнни

Проведя визуальный анализ рисунка 6, можно сделать вывод, что использование оператора Кэнни дает более качественный результат, который заключается в непрерывности и плавности найденных границ. Данный вывод подтверждается такими метриками, как чувствительность, специфичность, ошибки первого и второго рода, которые в свою очередь имеют более высокие показатели у метода Кэнни, чем у метода Собеля.

Библиографический список

1. Выделение границ. — Текст : электронный // Wikipedia : [сайт]. — URL: https://ru.wikipedia.org/wiki/Выделение_границ (дата обращения: 20.02.2022).
2. Детектор границ Канни. — Текст : электронный // Habr : [сайт]. — URL: <https://habr.com/ru/post/114589/> (дата обращения: 20.02.2022).
3. Алгоритмы выделения контуров изображений. — Текст : электронный // Habr : [сайт]. — URL: <https://habr.com/ru/post/114452/> (дата обращения: 20.02.2022).
4. Матричные фильтры обработки изображений. — Текст : электронный // Habr : [сайт]. — URL: <https://habr.com/ru/post/142818/> (дата обращения: 20.02.2022).

УДК 004.932; ГРНТИ 28.23.15

ИССЛЕДОВАНИЕ БИТОВОГО ТРАФИКА ПРИ ПЕРЕДАЧЕ ЛАНДШАФТНЫХ ИЗОБРАЖЕНИЙ

Н.Н. Гринченко, А.А. Вьюгина, С.Н. Баранова

Рязанский государственный радиотехнический университет имени В.Ф. Уткина,
Российская Федерация, Рязань, baranova.sv.n@gmail.com

Аннотация. Целью работы является исследование возможности передачи ландшафтного изображения через зашумленный канал связи с минимальной потерей информативности. Битовая последовательность представляется в виде плоскостей в соответствии с разрядами и над измененным массивом данных производится наложение шумов различной интенсивности. Результатом эксперимента является сравнение выходной последовательности без дополнительного этапа кодирования и с ним.

Ключевые слова: ландшафтные изображения, битовый трафик, битовые плоскости, код Хэмминга, канал связи.

RESEARCH OF BIT TRAFFIC IN THE TRANSMISSION OF LANDSCAPE IMAGES

N.N. Grinchenko, 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. The aim of the work is to investigate the possibility of transmitting landscape images through a noisy communication channel with minimal loss of informativeness. The bit sequence is represented in the form of planes in accordance with the bits and noise of varying intensity is superimposed over the modified data array. The result of the experiment is a comparison of the output sequence without an additional coding step and with it.

Keywords: landscape images, bit traffic, bit planes, hamming code, communication channel.

С каждым годом увеличивается объем информации, которая передается между различными устройствами по сети. Большую часть данных составляют различного типа изображения, несущие различное количество информации. В таком случае возникает необходимость в ускорении передачи данных. При этом возможны появления искажений сигнала при передаче по линиям связи [1]. Перечисленные проблемы образуют задачу по исследованию битового трафика, которая сводится к анализу возможностей уменьшения количества передаваемых данных.

Первым шагом в исследовании стало разделение изображения на битовые плоскости, под которыми понимается разбиение яркости пикселя на отдельные биты и создание отдельной матрицы для каждого разряда [2]. При выполнении данного преобразования исходная матрица яркостей, состоящая из значений, хранимых в виде байта, увеличивалась в 8 раз при переходе на хранение в виде бит.

$$\mathbf{G}_{N \times N} = \mathbf{G}_{(N \cdot 8) \times N} \quad (1)$$

Затем для каждой плоскости выбираются соответствующие значения матрицы в зависимости от номера битовой плоскости k . Количество битовых плоскостей зависит от количества бит, необходимых для хранения одного пикселя. В данном случае под один пиксель был использован 1 байт, соответственно количество битовых плоскостей k может принимать значения от 1 до 8. Формирование матрицы битовой плоскости представлено на формуле 2.

$$\mathbf{B}_{N \times N}^k = \begin{pmatrix} g_{k,1} & g_{k+8,1} & \dots & g_{k+8 \cdot N,1} \\ \vdots & & \ddots & \vdots \\ g_{k,N} & g_{k+8,N} & \dots & g_{k+8 \cdot N,N} \end{pmatrix}. \quad (2)$$

В итоге были получены срезы изображения по каждому из восьми разрядов. Пример перехода к битовым плоскостям для изображения размером 256x256 представлен на рисунке 1.

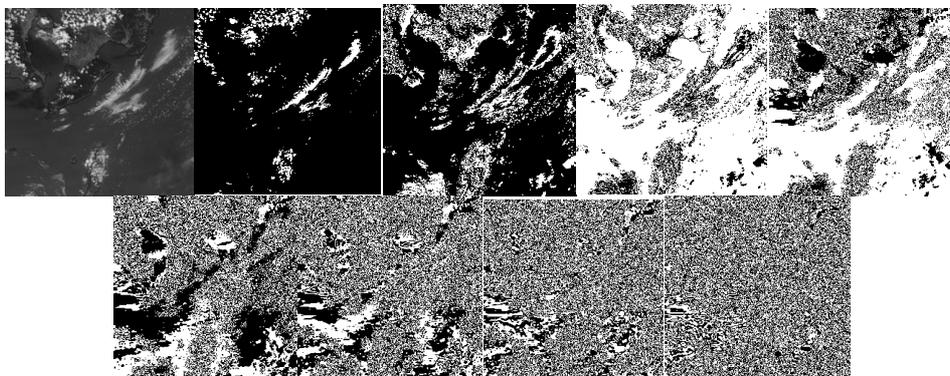


Рис. 1. Битовые плоскости изображения

При отображении матрицы отдельных разрядов можно обратить внимание на большую информативность старших бит. При этом младшие биты отвечают за оттенки серого, благодаря чему итоговое изображение имеет большую сглаженность. Сравнение исходного изображения и без 0-го и 1-го бита яркости пикселей отображены на рисунке 2 (самые явные искажения выделены в круг).

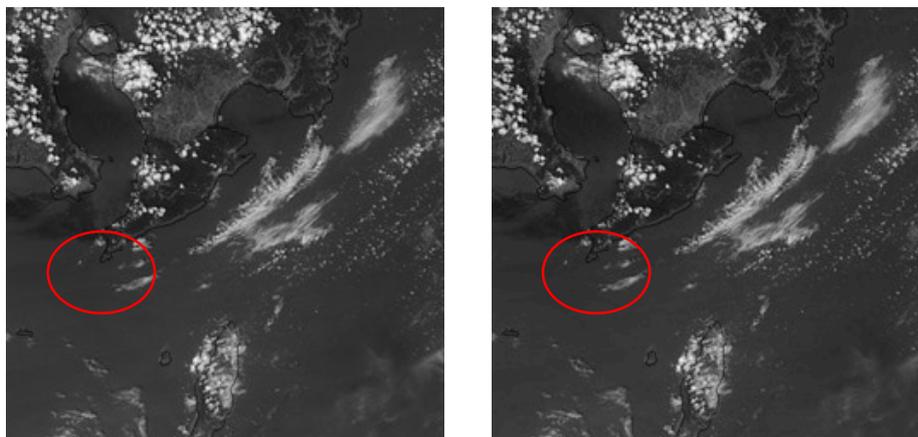


Рис. 2. Сравнение изображений

Таким образом, можно предположить, что при передаче изображений по линиям связи основную опасность в искажении на принимаемой стороне представляет влияние шумов на старшие биты передаваемых данных. В таком случае для защиты от шумов при передаче информации следует использовать кодирование [3]. Однако накладывание кода на всю последовательность сильно утяжеляет всю цепочку данных и увеличивает время передачи.

Вторым шагом в исследовании битового трафика было изменение последовательности передаваемых данных: полученные битовые плоскости соединены от матрицы со старшими разрядами к младшим. В итоге получены 2 равноценные последовательности: с исходным и измененным следованием бит (рисунок 3).

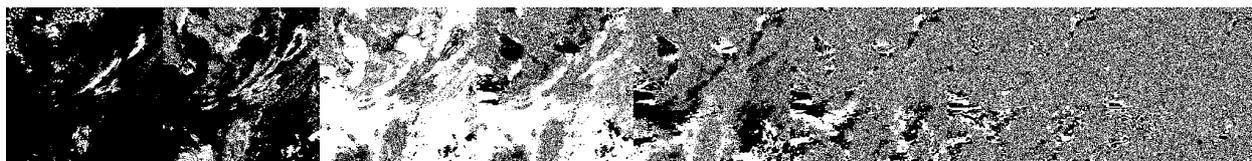


Рис. 3. Битовые последовательности по битовым плоскостям

При таком расположении разрядов по сравнению с исходным появляется большая вероятность попадания шума на значащий разряд. Однако сосредоточенность старших бит в начале последовательности позволяет наложить более сильную защиту только на часть всех данных в сравнении с необходимостью защиты всей информации в исходной последовательности пикселей.

Наложение шумовой составляющей при рассмотрении битовых последовательностей можно представить как сумму двух матриц (3). Так как исходная последовательность представлена в битовом виде, то шум может быть представлен последовательностью 0 и 1, при этом сумма элементов двух матриц также будет оставаться в значениях 0 и 1.

$$\mathbf{A}_{M \times N} = \mathbf{B} + \mathbf{Z} = \begin{pmatrix} b_{1,1} + z_{1,1} & b_{2,1} + z_{2,1} & \dots & b_{M,1} + z_{M,1} \\ \vdots & & \ddots & \vdots \\ b_{1,N} + z_{1,N} & b_{2,N} + z_{2,N} & \dots & b_{M,N} + z_{M,N} \end{pmatrix} \quad (3)$$

где \mathbf{B} – совокупная матрица битовых плоскостей, которые соединены от матрицы с большим разрядом к матрице с меньшим разрядом, \mathbf{Z} – матрица шумов.

Третьим шагом стало исследование гипотезы. В качестве защиты данных был взят код Хэмминга. Его основная мысль заключается в разбиении на классы кодируемой комбинации, каждая из которой представляет собой сумму по модулю 2 кода и образующих элементов данного класса [4]. Для исправления любой полученной ошибки необходимо определить класс смежности, каждому из которых соответствует некоторая контрольная последовательность символов. В двоичном коде при возникновении ошибки в определенном разряде опознавателя появляется 1.

В качестве эксперимента были взяты исходные сообщения в размере 8 и 16 бит и кодовые последовательности 4 и 5 бит соответственно. Единственным ограничением при использовании кода Хэмминга является то, что в сообщении будет обнаружено только одна ошибка [4]. В таком случае при увеличении шумовой составляющей существует вероятность искажения нескольких бит одного сообщения. При длине закодированного сообщения 12 бит выделяют 4 проверочных разряда: 1, 2, 4, 8. В таком случае группы проверочных разрядов формируются в соответствии с формулой 4.

$$\begin{aligned} a_1 &= a_3 \oplus a_5 \oplus a_7 \oplus a_9 \oplus a_{11} \\ a_2 &= a_3 \oplus a_6 \oplus a_7 \oplus a_{10} \oplus a_{11} \\ a_4 &= a_5 \oplus a_6 \oplus a_7 \oplus a_{12} \\ a_8 &= a_9 \oplus a_{10} \oplus a_{11} \oplus a_{12} \end{aligned} \quad (4)$$

Вид изображения с добавленным кодом Хэмминга представлен на рисунке 4.



Рис. 4. Изображение с кодом Хэмминга

При таком подходе длина сообщения будет увеличена в 1,5 раза при использовании 4 бит кода на каждые 8 бит. Однако результаты передачи по зашумленному каналу и восстановление изображения показывают, что наложение дополнительного кода на значащие разряды позволяет, не сильно перегружая исходное сообщение, добиться результата, который не теряет информативности. Рисунок 5 представляет восстановление изображения без кода и с кодом Хэмминга соответственно.

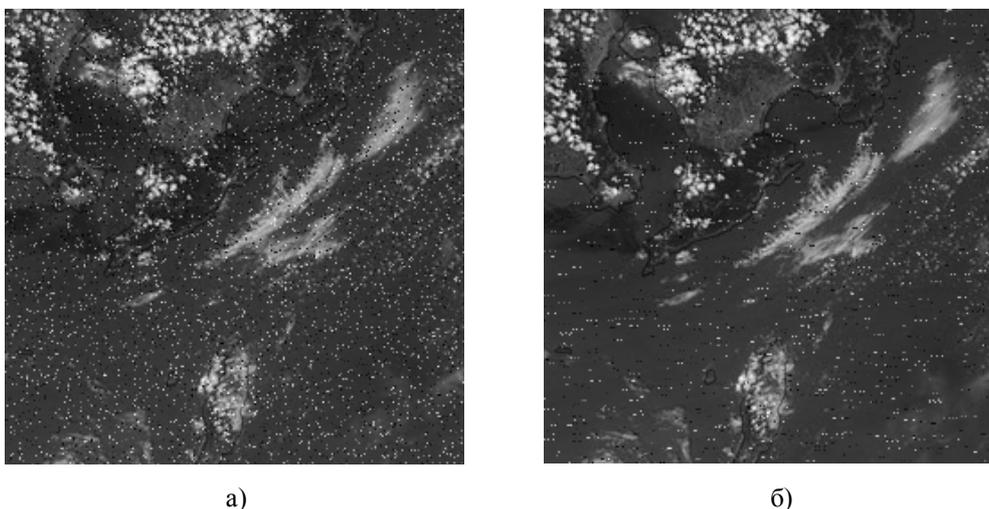


Рис. 5. Восстановленное изображение: а) без кода; б) с кодом Хэмминга

Для проведения эксперимента по кодированию и наложению шумовой составляющей на передаваемое изображение был использован пакет прикладных программ Matlab. Вся работа состояла из нескольких обособленных этапов:

1. Представление исходного изображения в виде сообщения из последовательности бит.
2. Разделение матрицы на битовые плоскости.
3. Кодирование битовых плоскостей с помощью кода Хэмминга.
4. Изменение последовательности исходного сообщения на порядок в виде битовых плоскостей.
5. Наложение шумовой составляющей.
6. Восстановление изображения.

Для выполнения шага 1 была разработана функция перевода матрицы байт в битовое представление, в ходе чего результирующая матрица была увеличена в 8 раз. Для преобразования байта в набор бит использована внутренняя функция среды Matlab `decimalToBinaryVector()` [5]. Для выполнения шага 3 также была написана функция по добавлению кода Хэмминга для кода 4 при длине сообщения 8 (уже существующие функции кодирования не позволяют работать с последовательностью 12 бит). Алгоритм вычисления кода подробно описан в главе 2 [4].

Исследование проводилось на нескольких изображениях различного размера: от 32x32 до 256x256 пикселей. Шумовая составляющая, наложенная на исходную последовательность, варьировалась от 3 до 15%. Во всех случаях СКО не превышало 0,1 для кодированного изображения [6]. Также стоит отметить, что при кодировании части битовых плоскостей, соответствующие значащим битам, результат не теряет информативности.

Таким образом, исследование битовой последовательности в рамках данной работы указывает на возможность передачи сообщений через зашумленный канал с минимальной нагрузкой в виде дополнительной информации. Такой подход позволяет выделять группы значащих значений и выделять на такую группу большее количество ресурсов для защиты

данных, учитывая для конкретной задачи, насколько серьезными могут быть потери в результирующем наборе данных.

Библиографический список

1. S. I. Babaev, A. Bastrychkin, B. V. Kostrov [et al.] Aspects of Binary Images Spectral Analysis // 2019 8th Mediterranean Conference on Embedded Computing, MECO 2019 - Proceedings : 8, Budva, 10–14 июня 2019 года. – Budva, 2019. – P. 8760003. – DOI 10.1109/MECO.2019.8760003.
2. Гонсалес Р., Вудс Р. Цифровая обработка изображений. Издание 4-е. Москва: Техносфера, 2018. – 1184 с.
3. Б. В. Костров, Н. Н. Гринченко, Е. С. Геращенко [и др.] // Моделирование канала передачи аэрокосмических изображений с использованием каскадного кода / Известия Тульского государственного университета. Технические науки. – 2016. – № 2. – С. 124-129.
4. Костров Б.В. Технологии физического уровня передачи данных: учебник для студ. учреждений сред. проф. образования – М.: Издательский центр «Академия», 2017. – 224с.
5. Амос Гилат MATLAB: Теория и практика. 5-е изд./ Пер. с англ. Смоленцев Н.К. – М.: ДМК Пресс, 2016. – 416с.
6. Vyugina, A. A., Baranova, S. N., Silkin, G. D., Khizrieva, N. I., & Bodrov, O. A. (2021). Research of spectral components structure in landscape images spectrum. Paper presented at the 2021 10th Mediterranean Conference on Embedded Computing, MECO 2021, doi:10.1109/MECO52532.2021.9460224

УДК 004.627; ГРНТИ 28.23.15

ФОРМИРОВАНИЕ ТРАФИКА В СИСТЕМАХ ПЕРЕДАЧИ ДАННЫХ

А.С. Бастрычкин, Б.В. Костров, Н.Н. Гринченко, Б.А. Костров

*Рязанский государственный радиотехнический университет имени В.Ф. Уткина,
Российская Федерация, Рязань, kostrov.b.v@evm.rsreu.ru*

Аннотация. В работе предлагается принцип формирования трафика в системах передачи изображений, позволяющий снизить пульсацию трафика в каналах связи и энергетические затраты.

Ключевые слова: системы передачи данных, обработка изображений, преобразование Уолша, квазидвумерный спектр, постоянная составляющая.

TRAFFIC GENERATION IN DATA TRANSMISSION SYSTEMS

A.S. Bastrychkin, B.V. Kostrov, N.N. Grinchenko, B.A. Kostrov

*Ryazan State Radio Engineering University named after V.F. Utkin,
Russia, Ryazan, kostrov.b.v@evm.rsreu.ru*

The summary. The paper proposes the principle of traffic generation in image transmission systems, which makes it possible to reduce traffic ripple in communication channels and energy costs.

Keywords: data transmission systems, image processing, Walsh transform, quasi-two-dimensional spectrum, constant component

За последние тридцать лет цифровая обработка изображений выросла из области научных исследований в техническую область, которая имеет научные и коммерческие применения. Многие из этих приложений связаны с огромными усовершенствованиями в компьютерных технологиях. Электронная обработка изображений теперь легко доступна на большинстве типичных настольных компьютеров в отличие от прошлого, когда только немногие имели такую возможность.

Во всех случаях цифровая обработка изображений связана с компьютерной обработкой изображений или, в более общем смысле, изображений, которые были преобразованы в числовую форму. Эти образы могут исходить из многих источников. Цифровые камеры, которые стали широко доступны, производят цифровые изображения вместо классической пленки. Сейчас любое изображение может быть оцифровано с помощью такого устройства,

как сканер. Что делает доступной реализацию практического применения цифровых изображений при хранении и передаче в информационных системах, в том числе специального назначения.

Для решения задач подобного рода используется переход в пространство спектральных образов передаваемых объектов [1]. Понятие разложения двумерной дискретной функции в ряд по системе ортогональных базисных функций тесно связано с понятием двумерных унитарных преобразований. Последние имеют обширное применение в области обработки изображений.

В результате прямого унитарного преобразования матрицы изображения $b(i, j)$ размером $M \times N$ образуется матрица преобразованного изображения того же размера, элементы которой по определению равны

$$F(u, v) = \sum_{i=0}^{N-1} \sum_{j=0}^{N-1} f(i, j) \phi_{(u,v)}(i, j), \quad (1)$$

где $\phi_{(u,v)}(i, j)$ – ядро прямого преобразования.

Исходное изображение можно получить с помощью обратного преобразования, описываемого соотношением

$$f(i, j) = \sum_{u=0}^{N-1} \sum_{v=0}^{N-1} F(u, v) \phi_{(i,j)}^{-1}(u, v),$$

где $\phi_{(i,j)}^{-1}(u, v)$ – ядро обратного преобразования.

Преобразование является унитарным, если выполняются условия ортогональности. Преобразование называют разделимым, если оба его ядра можно представить в следующей форме:

$$\phi_{(u,v)}(i, j) = \phi_u(i) \cdot \phi_v(j), \quad (2)$$

$$\phi_{(u,v)}^{-1}(i, j) = \phi_i^{-1}(u) \cdot \phi_j^{-1}(v).$$

Здесь разделение происходит на одномерные операторы преобразования столбцов и строк.

Результат воздействия оператора разделимого двумерного унитарного преобразования можно находить в два этапа. Сначала выполняется одномерное преобразование по всем столбцам матрицы изображения, при этом образуется матрица с элементами

$$P(i, v) = \sum_{j=0}^{N-1} f(i, j) \phi_u(j).$$

Затем выполняется второе одномерное преобразование по всем строкам полученной матрицы, в результате которого образуется массив чисел вида

$$F(u, v) = \sum_{i=0}^{N-1} P(i, v) \phi_v(i)$$

Унитарные преобразования также можно записывать, используя векторную форму обозначения изображений. Возможны различные интерпретации унитарных преобразований.

Преобразование изображения можно рассматривать как разложение исходного изображения в обобщенный двумерный спектр. Каждая спектральная составляющая характеризует вклад соответствующей спектральной (базисной) функции в энергию исходного изображения. При такой трактовке понятие частоты можно обобщить так, чтобы оно было применимо не только к синусам и косинусам, но и к другим функциям, на которых основываются преобразования. Таким образом, можно прийти к обобщению спектрального анализа для различных разложений.

В научной литературе описаны и другие унитарные преобразования, основанные на различных ортогональных системах базисных функций. Особый интерес представляют унитарные преобразования, основанные на нетригонометрических ортогональных системах базисных функций.

Преобразование Уолша-Адамара основано на квадратной матрице Адамара, элементы которой равны плюс или минус единице, а строки и столбцы образуют ортогональные векторы[2]. Наиболее просто построить матрицы Адамара размером $N \times N$, где $N = 2^k$, k – целое число. В таком случае преобразование известно также под названием преобразования Уолша. Применение двумерного преобразования для организации передачи изображений через канал связи вряд ли будет целесообразным, поскольку энергия низкочастотных составляющих может отличаться от энергии высокочастотных составляющих на несколько порядков, что неизбежно приведет к значительной пульсации трафика в системе передачи.

Преобразование Уолша является унитарным (1) и разделимым (2). Поэтому для него можно применить одномерный оператор преобразования строк или столбцов, что для случая матричной структуры изображения можно назвать квазидвумерным спектральным представлением. Дополнительно к применению квазидвумерного преобразования для выравнивания трафика можно предложить метод формирования трафика на основе разбиения целого изображения на фрагменты[4].

Дискретное двумерное преобразование Уолша[3] имеет вид

$$F(u, v) = \frac{1}{MN} \sum_{i=0}^{M-1} \sum_{j=0}^{N-1} f(i, j) \text{wal}_{(u,v)}\left(\frac{i}{M}, \frac{j}{N}\right)$$

Обратное дискретное двумерное преобразование Уолша имеет вид

$$f(i, j) = \sum_{u=0}^{M-1} \sum_{v=0}^{N-1} F(u, v) \text{wal}_{(i,j)}\left(\frac{u}{M}, \frac{v}{N}\right)$$

Дискретизация функций Уолша приводит к переупорядоченной матрице Адамара размером $N \times N$, которую будем обозначать \mathbf{H}^W .

Матрица изображения будет определяться значениями яркости его элементов:

$$\mathbf{B} = \begin{bmatrix} b_{11} & b_{12} & \dots & b_{1j} & \dots & b_{1N} \\ b_{21} & b_{22} & \dots & b_{2j} & \dots & b_{2N} \\ \dots & \dots & \dots & \dots & \dots & \dots \\ b_{i1} & b_{i2} & \dots & b_{ij} & \dots & b_{iN} \\ \dots & \dots & \dots & \dots & \dots & \dots \\ b_{N1} & b_{N2} & \dots & b_{Nj} & \dots & b_{NN} \end{bmatrix},$$

где $i = \overline{1, N}$ и $j = \overline{1, N}$.

При разбиении изображения оно будет состоять из отдельных матриц:

$$B = \begin{bmatrix} B_{11} & B_{12} & \dots & B_{1j} & \dots & B_{1K} \\ B_{21} & B_{22} & \dots & B_{2j} & \dots & B_{2K} \\ \dots & \dots & \dots & \dots & \dots & \dots \\ B_{i1} & B_{i2} & \dots & B_{ij} & \dots & B_{iK} \\ \dots & \dots & \dots & \dots & \dots & \dots \\ B_{K1} & B_{K2} & \dots & B_{Kj} & \dots & B_{KK} \end{bmatrix}, \quad (3)$$

где $i = \overline{1, K}$ и $j = \overline{1, K}$; $K = \frac{N}{n}$; n – число элементов в строке фрагмента (число строк фрагмента);

$$B_{ij} = \begin{bmatrix} b_{11}^n & b_{12}^n & \dots & b_{1j}^n & \dots & b_{1n}^n \\ b_{21}^n & b_{22}^n & \dots & b_{2j}^n & \dots & b_{2n}^n \\ \dots & \dots & \dots & \dots & \dots & \dots \\ b_{i1}^n & b_{i2}^n & \dots & b_{ij}^n & \dots & b_{in}^n \\ \dots & \dots & \dots & \dots & \dots & \dots \\ b_{n1}^n & b_{n2}^n & \dots & b_{nj}^n & \dots & b_{nn}^n \end{bmatrix},$$

где $i = \overline{kn - (n - 1), nk}$; $j = \overline{kn - (n - 1), nk}$ – определяют связь номеров элементов исходного изображения с номерами фрагментов, $k = \overline{1, K}$.

Спектральное представление всех фрагментов изображения и их обратное восстановление будет определяться системами однородных уравнений:

$$\left\{ \begin{array}{l} F_{11} = B_{11}H^W \\ \dots \\ F_{1K} = B_{1K}H^W \\ \dots \\ F_{i1} = B_{i1}H^W \\ \dots \\ F_{iK} = B_{iK}H^W \\ \dots \\ F_{K1} = B_{K1}H^W \\ \dots \\ F_{KK} = B_{KK}H^W \end{array} \right. \quad \left\{ \begin{array}{l} B_{11} = F_{11}H^W \\ \dots \\ B_{1K} = F_{1K}H^W \\ \dots \\ B_{i1} = F_{i1}H^W \\ \dots \\ B_{iK} = F_{iK}H^W \\ \dots \\ B_{K1} = F_{K1}H^W \\ \dots \\ B_{KK} = F_{KK}H^W \end{array} \right.$$

Переход к фрагментации изображения позволяет снизить максимальные спектральные составляющие в десятки раз. Однако составляющие спектра соответствующие нулевой частоте в каждом фрагменте превосходят остальные составляющие спектра более чем в десять раз. Устранение данного недостатка можно добиться при использовании метода передачи без постоянной составляющей. Для реализации этого метода необходимо ввести нулевые элементы в первые столбцы матриц фрагментов.

Матрица яркостей элементов фрагментов изображений примет следующий вид:

$$B_{ij} = \begin{bmatrix} 0 & b_{12}^n & \dots & b_{1j}^n & \dots & b_{1n}^n \\ 0 & b_{22}^n & \dots & b_{2j}^n & \dots & b_{2n}^n \\ \dots & \dots & \dots & \dots & \dots & \dots \\ 0 & b_{i2}^n & \dots & b_{ij}^n & \dots & b_{in}^n \\ \dots & \dots & \dots & \dots & \dots & \dots \\ 0 & b_{n2}^n & \dots & b_{nj}^n & \dots & b_{nn}^n \end{bmatrix}. \quad (4)$$

Стыковка соседних фрагментов будет проводиться с наложением соседних фрагментов и связь номеров элементов исходного изображения с номерами фрагментов будет определяться следующим образом:

$$i = \overline{kn - (k - 1) - (n - 1), nk} ; \quad j = \overline{kn - (n - 1), nk}. \quad (5)$$

На рисунке 1 изображены строки данных при различных подходах к формированию спектров изображений.

Верхний график демонстрирует строку спектра изображения при квазидвумерном преобразовании без разделения на фрагменты. Как показано на графике, максимальное значение передаваемой таким образом строки может достигать 15000 – 18000, тогда как остальные составляющие находятся гораздо ниже. Такой подход может приводить к чрезмерным энергетическим затратам на передачу изображения с применением перехода к спектральным пространствам.

На среднем графике показана строка спектра, полученного при использовании метода предварительного разделения изображения на фрагменты(3). Максимальные значения теперь находятся ниже (около 2000), но всё равно они существенно превышают остальные значения, к тому же теперь наблюдается пульсация графика.

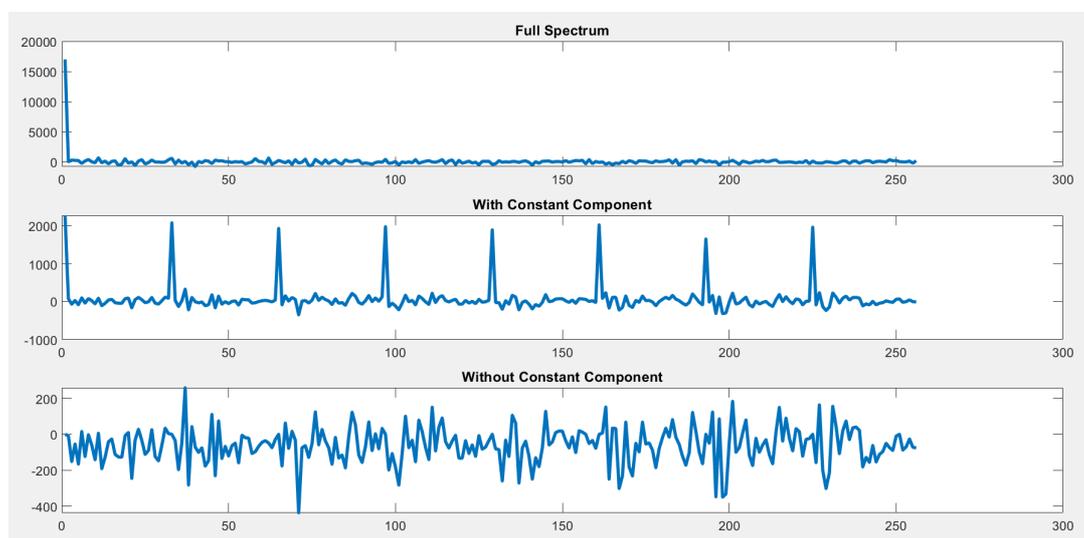


Рис. 1. Сравнение составляющих спектров при применении различных подходов к передаче

Нижний график рисунка 1 иллюстрирует подход разделения изображения на фрагменты в совокупности с применением метода устранения постоянных составляющих при передаче и их восстановления на приёмной стороне(4,5). Данный подход показал наиболее низкие значения максимумов и минимумов (от 200 до - 450) и существенное уменьшение пульсации графика.

Таким образом, получение спектров фрагментов изображений может быть полезно в задачах снижения энергетической нагрузки на каналы связи передачи изображений. Устранение постоянных составляющих в квазидвумерных спектрах фрагментов изображений даёт существенное изменение диапазона передаваемых значений.

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.
4. Baranchikov, A.I., Babaev, S.I., Sumenkov, N.A., Zverev, A.E., Bastrychkin, A.S. The Issues of Orthogonal Basis Dimension Selection in Embedded Image Processing Systems. - 2021 10th Mediterranean Conference on Embedded Computing, MECO 2021, 2021.

УДК 004.738.5; ГРНТИ 19.31

ИНТЕРНЕТ ВЕЩЕЙ В ИНТЕЛЛЕКТУАЛЬНОЙ МОДЕРНИЗАЦИИ БЫТОВОГО ГАЗОВОГО ОБОРУДОВАНИЯ

И.С. Иванов, А.В. Рябинин, М.Б. Никифоров

*Рязанский государственный радиотехнический университет имени В.Ф. Уткина,
Российская Федерация, Рязань, ryabinin7000@gmail.com*

Аннотация. В данной работе рассматриваются методы модернизации бытового газового оборудования, просуммированные под названием «Умная плита», приводятся способы автоматизации работы пользователя с газовой плитой, объединённые в целостную систему.
Ключевые слова: интернет вещей, умный дом, бытовое газовое оборудование, регулирование подачи газа.

INTERNET OF THINGS IN INTELLIGENT MODERNISATION OF HOUSEHOLD GAS EQUIPMENT

I.S. Ivanov, A.V. Ryabinin, M.B. Nikiforov

*Ryazan State Radio Engineering University named after V.F. Utkin,
Russia, Ryazan, ryabinin7000@gmail.com*

The summary. This paper discusses methods for upgrading household gas equipment, summarized under the title "Smart stove", and provides ways to automate the user's work with a gas stove, combined into a complete system.
Keywords: internet of things, smart home, household gas equipment, gas supply regulation.

Введение

Системы автоматизации быта типа «Умный дом» снискали большой успех и прочно зарекомендовали себя как одно из центральных направлений развития современной техники. Несмотря на то, что сама концепция «Умного дома» отнюдь не является новой, технологии в данной сфере и поныне продолжают крайне интенсивно развиваться. Главная причина этого – высокая актуальность подобных систем.

Большой интерес представляет сфера модернизации бытового газового оборудования. Рассматриваемая в данной статье система носит название «Умная плита». Она представляет собой кухонную плиту с расширенным функционалом и более прогрессивным подходом к приготовлению пищи. Умная плита позволит пользователю контролировать процесс готовки с использованием мобильного или десктопного приложения, поддерживать обратную связь

путём отправки уведомлений, автоматически регулировать температуру приготовляемой пищи. Перечисленные функции помогут значительно упростить процесс готовки.

Разработка структуры

Рассмотрим подробнее технологию системы. Контроль системой осуществляется при помощи микроконтроллера Arduino Nano (на базе архитектуры ATmega328), подключаемого к сети через 5-вольтовый блок питания. К блоку через преобразователь 5-3 В подключается Wi-Fi-модуль ESP8266, через который, в свою очередь, происходит подключение к домашней сети Wi-Fi и далее – к мобильному телефону или персональному компьютеру пользователя, что позволяет управлять процессом готовки с приложения. Подключение к Wi-Fi-адаптеру осуществляется по протоколу UART. После базовых приготовлений, осуществляемых вручную (например, поставить кастрюлю с водой на плиту), пользователю необходимо будет выбрать рецепт из списка в приложении. После выбора рецепта соответствующие инструкции будут по Wi-Fi переданы плате Arduino. В тот момент, когда всё готово к приготовлению пищи, пользователь вручную включает газ и запускает из приложения процесс начала готовки.

После запуска плиты контроль за приготовляемой пищей осуществляют инфракрасные (ИК) датчики (термометры). Датчики подключаются к контроллеру по протоколу I2P, причем в выбранных моделях имеется возможность менять адрес подключения, поэтому нет необходимости в использовании мультиплексора. Предполагается использование 4 датчиков MLX90614 – по одному на каждую конфорку плиты. Они должны быть настроены особым образом – быть направленными на еду, но при этом не находиться слишком далеко (оптимальное расстояние – около одного метра). Для увеличения точности путём сужения области измерения, а также для защиты датчиков от загрязнения предполагается использование луп. Датчики считывают температуру, данные о которой передаются микроконтроллеру. Тот, в свою очередь, регулирует подачу газа через 4 сервопривода MG995, подключённых к газовым кранам (когда нужно поставить «медленный» или наоборот «сильный» огонь, т.е. увеличить или уменьшить температуру – в зависимости от рецепта). Сервоприводы подключаются к разным цифровым выходам Arduino. Время отдельных этапов приготовления пищи зафиксировано для каждого конкретного рецепта в приложении; по истечении указанного срока пользователю посылаются уведомления (например: «посолить», «добавить лук» и т.п.), а на плату через Wi-Fi-модуль отправляются инструкции о дальнейших действиях. По окончании процесса готовки плита автоматически выключается.

Размещение компонентов Умной плиты будет проходить следующим образом. Микроконтроллер Arduino и подключённый к нему Wi-Fi-адаптер будут располагаться в любом подходящем месте на расстоянии не менее 30 см от плиты. Такое решение принимается в целях обезопасить плату от термического воздействия. Компактные размеры указанных устройств (1.85 см x 4.2 см x 1.1 см для Arduino Nano и 16 мм x 24 мм x 3 мм для ESP8266) должны полностью исключить проблемы с их размещением. Для безопасности и сведения к минимуму влияния внешних факторов микроконтроллер и адаптер будут помещены в специальный корпус. Четыре сервопривода, подключённых к плате, будут находиться в свободном пространстве газовой плиты под конфорками. К каждому сервоприводу будет подключён металлический стержень с припаянной к нему шестерёнкой на противоположном конце; эта шестерёнка будет сопряжена с другой шестерёнкой, насаженной на газовый кран. При получении соответствующей инструкции плата будет посылать сигнал на сервопривод, который начнёт вращать стержень, приведя в движение через шестерни газовый кран; таким образом будет осуществляться регулирование подачи газа. Сервоприводы будут подключаться к плате через пины; датчики температуры – по протоколу I2P. Датчики будут находиться над плитой на расстоянии около одного метра и будут направлены непосредственно на конфорки (по одному датчику на конфорку).

Управление газом будет осуществляться с помощью пропорционально-интегрально-дифференциальной схемы управления. Для того чтобы контроль температуры происходил плавно, будет применяться ПИД-регулирование.

Структурная схема «Умной плиты» представлена на рисунке 1.

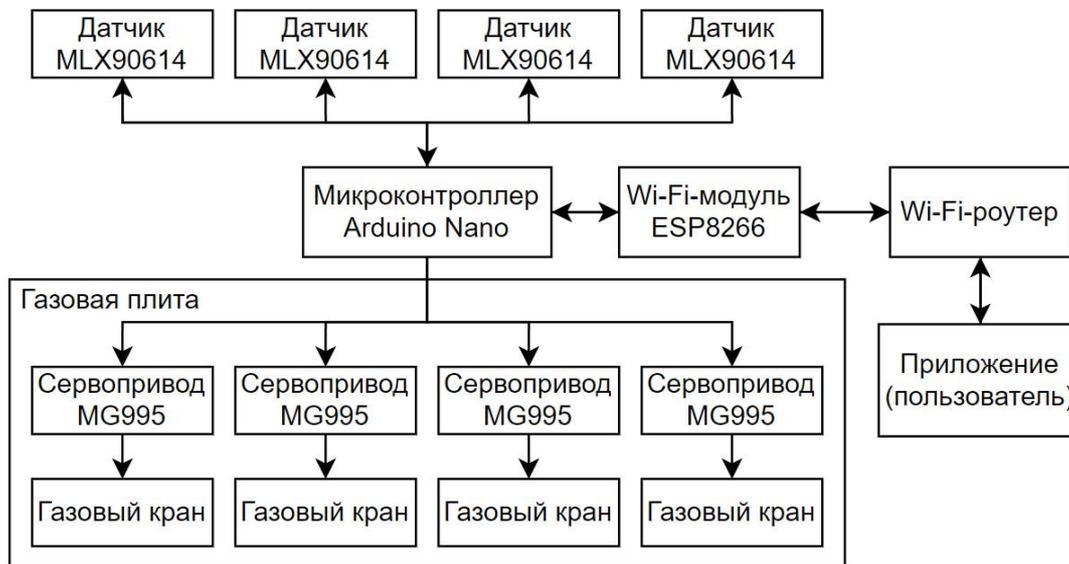


Рис. 1. Структурная схема «Умной плиты»

Выбор элементов

Плата Arduino Nano, построенная на архитектуре ATmega328 и производимая компанией Gravitech [3], широко распространена и может применяться для самых разнообразных задач. Рассмотрим её характеристики:

- рабочее напряжение – 5 В;
- входное напряжение (рекомендуемое) – 7-12 В;
- входное напряжение (предельное) – 30 В;
- цифровые входы/выходы – 14;
- аналоговые входы – 8;
- постоянный ток через вход/выход – 40 мА;
- флеш-память – 32 Кб;
- ОЗУ – 2 Кб;
- EEPROM – 1 Кб;
- тактовая частота – 16 МГц;
- размеры – 1.85 см x 4.2 см x 1.1 см.

Достаточное количество входов и выходов платы исключает затруднения с подключением к ней большого числа устройств. Высокая тактовая частота и значительный объём памяти позволят Arduino обрабатывать полученные сигналы и посылать свои с высокой скоростью. Удобный форм-фактор создаст благоприятные условия для размещения микроконтроллера.

Обратимся к характеристикам Wi-Fi-модуля ESP8266 для Arduino:

- работает с внешней флэш-памятью по интерфейсу SPI;
- объём флэш-памяти – 0,5-4 Мб;
- порты ввода-вывода – 14 (SPI, I²S, UART, 10-bit АЦП);
- питание – 2,2-3,6 В;
- размеры – 16 мм x 24 мм x 3 мм.

Данный модуль может обеспечить стабильное подключение микроконтроллера Arduino Nano к домашней Wi-Fi-сети [4], что обеспечит связь плиты с пользователем через приложение. Кроме того, ESP8266 также обладает компактными размерами, что обеспечит удобное его расположение.

Опишем также датчик температуры. MLX90614 – это инфракрасный термометр в корпусе TO-39 для бесконтактного измерения температуры [5]. В корпусе объединены ИК детектор (MLX81101) и микросхема обработки сигнала (MLX90302). Благодаря применению маломощного усилителя, 17-битного АЦП и мощного DSP-процессора датчики имеют высокую точность и разрешение. Каждый датчик имеет I²S адрес по умолчанию, который, однако, может быть изменен, что является одним из главных преимуществ датчика. Рассмотрим некоторые его характеристики:

- 2 режима работы: режим измерения температуры объекта и режим измерения температуры окружающей среды;
- диапазон измерения температур: -70 – 380°C;
- угол обзора – 80°.

Приложение

Приложение, обеспечивающее связь пользователя с Умной плитой, позволяет отправлять команды на плату Arduino. При начале работы в целях поддержания динамического IP приложение ищет нужное устройство с помощью опросов имеющихся адресов. Предварительно необходимо подключиться к точке доступа плиты, после чего передать seed и пароль для домашней сети. Инструкции для Arduino передаются POST-запросом по адресу контроллера. Приложение отправляет последовательность команд, которая принимается платой, записывается в очередь и обрабатывается.

Приведём пример информации внутри запроса (в удобной для человека форме):

Довести до кипения

Ожидать подтверждения №1

Варить 20 минут при 60°C

Ожидать подтверждения №2

Варить 5 минут при 80°C

Выключить огонь

На определенных этапах Arduino будет высылать ответные запросы, «спрашивая» пользователя о дальнейших действиях. Например, при ожидании добавления специй на устройство пользователя будут приходить уведомления; они будут расшифровываться и показывать подробную информацию о дальнейших действиях.

В приложении, в разделе «Настройки», будет раздел для подключения Arduino, описанный ранее, и общие настройки приложения (звук, размер шрифта и пр.). Раздел с выбором рецептов от других пользователей (они загружаются с сервера или установлены изначально) для ускорения навигации будет содержать вкладку «Избранные рецепты». Кроме того, будет существовать раздел «Создать рецепт». При выборе рецепта будет вариант выбора конфорки; после его осуществления приложение запросит пользователя зажечь огонь и запустить процесс готовки. Информация о начальных действиях повара будет представлена изначально. После в разделе «текущее приготовление» появится данная конфорка, рецепт, таймер до следующего этапа и общий прогресс. На устройство будут приходить уведомления о необходимых действиях со стороны повара.

Целесообразность разработки

Разработка и использование технологии Умной плиты целесообразны ввиду целого ряда причин. Во-первых, обширный функционал плиты значительно уменьшит трудовые за-

траты на приготовление пищи. Теперь обладателю данной технологии не нужно будет выучивать или подолгу искать рецепты, постоянно следить за временем, жертвуя своими личными делами – об этом позаботится сама плита. Во-вторых, система контроля температуры пищи позволит сократить риск возникновения как несчастных случаев, так и незначительных, но всё же нежелательных казусов. В-третьих, огромный потенциал представляет предложение, разрабатываемое под умную плиту, поскольку его функционал не ограничивается простым хранением рецептов и выводом уведомлений. В-четвёртых, умная плита будет иметь не очень высокую цену сравнительно обычной плиты и будет окупаться комфортабельностью процесса приготовления пищи.

Оценим денежные затраты на создание технологии «Умной плиты». Все цены взяты из открытых источников:

- плата Arduino Nano – 465р.;
- Wi-Fi модуль ESP8266 – 176р.
- сервопривод MG995 – 343р. х4
- датчик MLX90614 – 627р. х4
- блок питания 5В – 225р.
- преобразователь напряжения 5-3В – 79р.

Итоговая стоимость составляет 4825 российских рублей. Имеет смысл утверждать, что данная модификация с позиций улучшения плиты относительно денежных затрат экономически обоснована.

Заключение

Результаты проведённой работы позволяют утверждать целесообразность разработки «Умной плиты». Высокая актуальность технологий в сфере «Интернет вещей» в совокупности с насущной потребностью модернизации бытовых приборов делают описанный механизм выгодным решением. Можно утверждать, что предложенная технология является экономически обоснованной, поскольку конечная стоимость «Умной плиты» не будет чересчур высокой сравнительно обыкновенной газовой плиты. Большой потенциал представляет предложение для управления «Умной плитой»: оно позволит не только автоматизировать процесс приготовления пищи, сводя к минимуму действия пользователя, но и управлять газовой плитой из любой точки, где есть доступ к Wi-Fi-сети, используемой адаптером. Технически система представляется вполне реализуемой и возможной к использованию в бытовых условиях.

Библиографический список

1. Кранц М. Интернет вещей: новая технологическая революция. — М.: Эксмо, 2017. — 330 с.
2. Балашов Е. П., Пузанков Д. В. Микропроцессоры и микропроцессорные системы. — М., 1981.
3. Arduino Nano [Электронный ресурс] // Arduino.ru: официальный сайт Arduino на русском. URL: <http://arduino.ru/Hardware/ArduinoBoardNano> (дата обращения: 25.02.2022).
4. WiFi ESP8266 в проектах Arduino [Электронный ресурс] // Arduinomaster.ru: сообщество разработчиков. URL: <https://arduinomaster.ru/platy-arduino/arduino-esp8266> (дата обращения: 25.02.2022)
5. Тепловизор MLX90614 [Электронный ресурс] // Platan.ru: интернет-магазин электроники. URL: <https://www.platan.ru/news/MLX90614.shtml> (дата обращения: 25.02.2022)

УДК 004.62; ГРНТИ 50.41.17

ИССЛЕДОВАНИЕ СТРУКТУРЫ РЕПОЗИТОРИЯ УНАСЛЕДОВАННОЙ ИНФОРМАЦИОННОЙ СИСТЕМЫ ORACLE DESIGNER/2000

Т.И. Кондрашова, А.И. Ефимов

*Рязанский государственный радиотехнический университет имени В. Ф. Уткина,
Российская Федерация, Рязань, tatyanakondrashova00@gmail.com*

Аннотация. Настоящая статья посвящена исследованию структуры БД репозитория CASE-средства Oracle Designer/2000, применявшегося в течении многих лет в деятельности филиала ОКБ «Спектр».

Ключевые слова: репозиторий, информационная система (ИС), база данных (БД), CASE-средства, схема структуры данных (DSD), язык SQL.

INVESTIGATION OF THE REPOSITORY STRUCTURE OF THE LEGACY INFORMATION SYSTEM ORACLE DESIGNER/2000

T. I. Kondrashova, A.I. Efimov

*Ryazan State Radio Engineering University named after V. F. Utkin,
Russian Federation, Ryazan, tatyanakondrashova00@gmail.com*

The summary. This article is devoted to the research of the database structure of the repository of the Oracle Designer/2000 CASE tool, which has been used for many years in the activities of the OKB Spectrum branch.

Keywords: repository, information system (IS), database (DB), CASE tools, data structure schema (DSD), SQL language.

Тенденция развития информационных технологий приводит к постоянному возрастанию сложности информационных систем (ИС). Это способствовало появлению программно-технологических средств специального класса – CASE-средств.

CASE-средства – это программные средства, поддерживающие процесс разработки, создания и сопровождения ИС, проектирования приложений и баз данных (БД), а также многие другие процессы.

Любые среды разработки, в том числе CASE-средства, имеют ограниченный срок службы. Программы, как и люди, стареют. Мы не можем предотвратить старение, поэтому возникает необходимость импорта накопленной информации (модели, репозиторий объектов БД и др.) в современные средства разработки. Эта задача осложняется, если унаследованная (legacy) система не имеет встроенных средств экспорта данных. В этом случае перенос данных должен предваряться исследованием структуры хранилища унаследованной системы. В зависимости от CASE-средства это может быть структура файлов либо структура базы данных репозитория.

Исследование структуры БД Oracle Designer/2000

Информация Oracle Designer/2000 хранится в репозитории - наборе таблиц БД в схеме владельца репозитория «SYSCASE». Репозиторий содержит множество таблиц, но на данный момент мы будем исследовать SDD_ELEMENTS, в которой хранятся элементы (объекты) репозитория и связи (один ко многим) и SDD_STRUCTURE_ELEMENTS - связи (многие ко многим) между ними.

Таблица SDD_ELEMENTS хранит элементы, создаваемые пользователями в процессе работы в Oracle Designer/2000: сущности, атрибуты, ключи, ограничения целостности, таблицы, индексы, триггеры, последовательности и т.п. Каждый элемент занимает одну запись в этой таблице, которая содержит описание элемента, его тип, уникальный номер, а также ссылки на другие элементы репозитория (если элемент тем или иным образом связан с ними).

При создании новой сущности (например, в Entity Relationship Diagrammer'е) в таблице SDD_ELEMENTS создается новая запись (элемент типа сущность). То же происходит и при создании других элементов - таблиц, последовательностей и т.п. Таким образом, SDD_ELEMENTS содержит полный перечень всех созданных в репозитории объектов.

Таблица SDD_ELEMENTS содержит элементы всех папок (APPLICATION SYSTEM) репозитория, причем папки сами являются элементами репозитория и хранятся в этой же таблице.

Таблица SDD_ELEMENT содержит следующие колонки (таблица 1):

Таблица 1 – Значение колонок таблицы SDD_ELEMENTS

Название колонки	Тип данных	Значение колонки
EL_ID	number (38,0)	уникальный номер элемента в репозитории (он же первичный ключ)
EL_TYPE_OF	varchar2(17)	тип элемента
EL_OCCUR_TYPE	varchar2(10)	дополнительная информация о типе элемента
EL_WITHIN_ID, EL_2ND_WITHIN_ID, EL_3RD_WITHIN_ID, EL_4TH_WITHIN_ID, EL_5TH_WITHIN_ID, EL_6TH_WITHIN_ID	number (38,0)	используются как ссылочные на EL_ID других элементов репозитория
EL_ELEM_OWNED_BY	number (38,0)	ссылка на EL_ID той папки, к которой принадлежит элемент
EL_COMMENT	varchar2(200)	комментарий
EL_DATE_CHANGED	date	дата последнего изменения элемента
EL_DATE_CREATED	date	дата создания элемента
EL_NAME	varchar2(100)	имя элемента
EL_SHORT_NAME	varchar2(20)	Короткое имя элемента
EL_USER_CREATED_BY	varchar2(30)	имя пользователя, создавшего элемент
EL_USER_CHANGED_BY	varchar2(30)	имя пользователя, изменившего элемент
EL_VERSION	number (3)	Версия файла
EL_VERSION_DATE	date	дата создания версии

Тип элемента определяется колонкой EL_TYPE_OF. Ниже приведены некоторые ее возможные значения. Для заполнения всех строк таблицы необходимо дальнейшее исследование (не все типы объектов репозитория применяются в работах филиала ОКБ «Спектр»).

Таблица 2 – Возможный тип элемента

Содержание колонки EL_TYPE_OF	Тип элемента
APP	Папка (Application System)
BUN	Должности
ATV	Значения доменов
ABT	Типы данных
UID	Уникальный идентификатор
UPR	Подлежит дальнейшему исследованию
LAN	Язык программирования
RELEND	Связь (подпись на диаграмме)
ATT	Атрибут сущности
COL	колонка таблицы
DOM	Домен
DOI	Подлежит дальнейшему исследованию
DOC	Файлы загрузки (скрипты таблиц и БД)
DIA	Диаграмма
DOG	Подлежит дальнейшему исследованию

Продолжение таблицы 2

Содержание колонки EL_TYPE_OF	Тип элемента
DFI	Файлы БД
ENT	сущность
FIL	файл ОС (OS File)
FUN	функция
GRP	группа пользователей (GROUP)
IND	индекс
INS	Подлежит дальнейшему исследованию
ITM	-“-
MOD	модуль
OCO	ограничение целостности (Constraint)
OCL	кластер
OKY	Подлежит дальнейшему исследованию
ODB	база данных (Oracle Data Base)
ORS	сегмент отката
OSP	Storage Definition
OSQ	последовательность (Sequence)
SYN	синоним
SPV	Подлежит дальнейшему исследованию
OTB	-"-
GRP	-"-
TAB	таблица
TRG	триггер
TSQ	Неограниченная квота
E20	части штаба УВД
E6	Расположение штаба УВД

Дополнительно тип элемента определяется колонкой EL_OCCUR_TYPE. Некоторые ее возможные значения:

Таблица 3 –Дополнительные типы элемента

Содержание колонки EL_TYPE_OF	Содержание колонки EL_OCCUR_TYPE	Тип элемента
OCO	CHECK	Check constraint
	FOREIGN	вторичный ключ
	PRIMARY	первичный ключ
	UNIQUE	уникальный ключ
TAB	TABLE	таблица
	VIEW	представление
OCL	ICL	индекс-кластер
	HCL	хэш-кластер
MOD	CHART	различные типы модулей
	FUNCTION	
	MANUAL	
	MENU	
	PACKAGE	
	REPORT	
	PROCEDURE	
	SCREEN	
	TRG-LOGIC	
UTILITY		

Теперь рассмотрим, создав и структуру таблицы SDD_STRUCTURE_ELEMENTS. Используя эту таблицу наряду с SDD_ELEMENTS, можно получить, например, связи моду-

лей с таблицами, функций с сущностями и т.п. Наиболее важными для нас колонками таблицы SDD_STRUCTURE_ELEMENTS являются (таблица 4):

Таблица 4 – Колонки таблицы SDD_STRUCTURE_ELEMENTS

Название колонки	Тип данных	Значение колонки
STR_TYPE_OF	varchar2(20)	тип связи элементов
STR_USE_OF	number(38,0)	ссылка на элемент
SRT_PART_OF	number(38,0)	ссылка на элемент
STR_INS_IND	varchar2(1)	признак возможности вставки
STR_SEL_IND	varchar2(1)	признак возможности выборки
STR_UPD_IND	varchar2(1)	признак возможности изменения
STR_DEL_IND	varchar2(1)	признак возможности удаления
STR_ARCH_IND	varchar2(1)	признак возможности архивации

Колонка STR_TYPE_OF может принимать следующие значения (таблица 5):

Таблица 5 — Значения колонки STR_TYPE_OF таблицы SDD_STRUCTURE_ELEMENTS

Значение колонки STR_TYPE_OF	Значение колонки STR_USE_OF	Значение колонки STR_PART_OF	Тип связи элементов
FUNENT	ссылка на сущность	ссылка на функцию	функция - сущность
MODFUN	ссылка на функцию	ссылка на модуль	модуль - функция
MODTAB	ссылка на таблицу	ссылка на модуль	модуль - таблица
TABENT	ссылка на сущность	ссылка на таблицу	таблица - сущность

Ссылочные колонки STR_USE_OF и STR_PART_OF содержат ID элемента, на который они ссылаются. Т.е., в нашем случае, они ссылаются на EL_ID таблицы SDD_ELEMENTS.

Колонки STR_INS_IND, STR_SEL_IND, STR_UPD_IND, STR_DEL_IND, STR_ARCH_IND показывают возможные операции одного элемента над другим (например, функции над сущностью).

Для дальнейшего исследования, целесообразным будет воспользоваться языком SQL и создать необходимые для исследования запросы.

Соберем информацию о типах диаграмм, которые хранятся в репозитории. Для данной цели создадим следующий запрос:

```
SELECT DISTINCT el_occur_type
FROM SDD_ELEMENTS
WHERE el_type_of='DIA'
```

Из данного запроса мы получили следующую информацию (таблица 6).

Таблица 6 – Типы хранимых диаграмм в репозитории

Содержание колонки EL_OCCUR_TYPE	Тип элемента
BPA	Модель бизнес-процессов
DSD	Схема структуры данных
ERD	Диаграмма отношений сущностей
FHD	Модель иерархии функции
MDD	Подлежит дальнейшему исследованию
MSD	"-"
MXD	"-"

Чтобы выяснить какие элементы могут входить в состав диаграммы, воспользуемся следующим запросом:

```
SELECT DISTINCT e.el_type_of
```

```
FROM SDD_STRUCTURE_ELEMENTS SE,SDD_ELEMENTS E
WHERE E.EL_ID = SE.STR_USE_OF AND se.str_type_of = 'DEU';
```

Результат представлен в таблице 7.

Таблица 7 – Элементы диаграмм репозитория

EL_TYPE_OF	Значение
ACI	Подлежит дальнейшему исследованию
APP	папка
ATT	Атрибут сущности
EL_TYPE_OF	Значение
BUN	Подлежит дальнейшему исследованию
COL	Колонка таблицы
DFL	Подлежит дальнейшему исследованию
DOI	-"
DST	-"
ENT	-"
EVT	сущность
FUN	функция
IGR	Подлежит дальнейшему исследованию
INS	-"
ITE	-"
LAN	-"
MCL	-"
MCN	-"
MOD	модуль
MTI	Подлежит дальнейшему исследованию
OCO	ограничение целостности (Constraint)
OKY	Подлежит дальнейшему исследованию
RELEND	Конец связи
TAB	таблица
WIN	Подлежит дальнейшему исследованию

На следующем шаге выясним, какие элементы могут входить состав диаграмм с типом DSD (Схема структуры данных). Для этого сначала создадим представления, которые будут содержать id диаграмм и элементов с типом DSD:

```
CREATE VIEW v_t3 AS
SELECT el_id, el_type_of ,el_occur_type
FROM SDD_ELEMENTS
WHERE el_occur_type = 'DSD';
```

```
CREATE VIEW v_t3_1 AS
SELECT
str_use_of, str_part_of, str_type_of
FROM v_t3 , SDD_STRUCTURE_ELEMENTS
WHERE str_part_of =EL_ID
```

Теперь зная необходимую нам информацию, создадим запрос:

```
SELECT
str_part_of, str_type_of, str_use_of, el_type_of, el_name,
el_short_name,
el_date_created, el_date_changed, el_user_created_by
FROM SDD_ELEMENTS, v_t3_1
WHERE EL_ID =STR_USE_OF
```

ORDER BY STR_PART_OF;

Из запроса было выяснено, что в диаграммах типа DSD хранятся элементы с типом ТАВ-таблица и ОСО-ограничение целостности (Constraint).

Исследование показало, что репозиторий имеет сложную структуру, разработанную с отступлениями от нормальных форм. В частности, одни и те же таблицы используются для хранения объектов существенно разных типов, связи между ними хранятся в других таблицах. Можно предположить, что разработчики фирмы Oracle руководствовались соображениями упростить возможное расширение структуры репозитория в будущем. Это обстоятельство существенно осложняет исследование структуры БД Oracle Designer/2000.

Библиографический список

1. Designer/2000 — новое поколение case-продуктов фирмы oracle [Электронный ресурс] — URL: <https://studfile.net/preview/6055204> (дата обращения 27.04.2022)
2. Халимон, В.И. Базы данных: учебное пособие / В.И. Халимон, Г.А. Мамаева, А.Ю. Рогов, В.Н. Чепикова — С-Пб.: СПбГТИ(ТУ), 2017. — 118 с.
3. Основы проектирования баз данных [Электронный ресурс] — URL: <https://metanit.com/sql/tutorial/1.1.php> (дата обращения 04.05.2022)

УДК 621.397;ГРНТИ 50.43.19

СОПРОВОЖДЕНИЕ ОБЪЕКТОВ С ИСПОЛЬЗОВАНИЕМ ФИЛЬТРА MOSSE

А.В. Пчелкин, М.Б. Никифоров

*Рязанский государственный радиотехнический университет имени В.Ф. Уткина
Россия, г. Рязань, alex.pchelkin@yandex.ru*

Аннотация. Объектом исследования является оптико-электронная система обнаружения и сопровождения наземных объектов. Предмет исследования – алгоритм корреляционного фильтра для выделения движущихся объектов в присутствии сложного динамического фона. Цель – разработка способов повышения эффективности решения задачи обнаружения движущихся объектов на сложном динамическом фоне. Исследования выполнены с использованием методов обработки видеоизображений и прогнозирования. Предложен помехоустойчивый метод выделения малоразмерных объектов на сложном динамическом фоне по последовательности кадров. Приведены качественные и количественные результаты работы метода на примере решения задачи обнаружения и сопровождения наземных объектов с изменяемым. Предложенный метод может быть использован при разработке систем обнаружения движущихся объектов по последовательности кадров в оптическом диапазоне.

Ключевые слова: обработка видеоизображений, обнаружение наземных объектов, обнаружение движущихся объектов, сложный динамический фон.

OBJECT TRACKING USING THE MOSSE FILTER

A.V. Pchelkin, M.B. Nikiforov

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

Annotation. The object of the study is an optoelectronic system for detecting and tracking ground objects. The subject of the study is a correlation filter algorithm for selecting moving objects in the presence of a complex dynamic background. The goal is to develop ways to improve the efficiency of solving the problem of detecting moving objects on a complex dynamic background. The research was carried out using the methods of video image processing and forecasting. A noise-resistant method for selecting small-sized objects on a complex dynamic background from a sequence of frames is proposed. The qualitative and quantitative results of the method are presented on the example of solving the problem of detecting and tracking ground objects with a variable. The proposed method can be used in the development of systems for detecting moving objects by a sequence of frames in the optical range.

Keywords: video image processing, detection of ground objects, detection of moving objects, complex dynamic background.

Введение

На сегодняшний день обеспечение визуального сопровождения объектов в пространстве является одной из ключевых задач в сфере систем компьютерного зрения, которая особенно актуальна для мониторинга и контроля точности перемещения объектов в таких условиях, как: недостаточная видимость, сложность рельефа или большое количества объектов, затрудняющих видимость.

Задача поиска движущихся при разработке систем сопровождения объектов является наиболее сложной. Определение областей движения и последующее распознавание объектов внутри полученных областей является одним из возможных подходов к решению вопроса по поиску движущихся объектов. Если требуется построить траекторию движения объекта (восстановить положение объекта на последовательном наборе изображений), то задача поиска будет усложняться. В таком случае выполняется, или поиск новых объектов на «сцене» и их дальнейшее сопровождение, или сопровождение областей движения и дальнейшее распознавание объектов в каждой области полученной последовательности.

Поскольку в последние годы визуальному отслеживанию уделяется много внимания, был разработан ряд надежных стратегий отслеживания предложены, которые допускают изменения внешнего вида цели и отслеживают цели с помощью сложных движений: Incremental Visual Tracking (IVT) [1], Robust Fragments-based Tracking (FragTrack) [4], Graph Based Discriminative Learning (GBDL) [2], and Multiple Instance Learning (MILTrack) [3]. Хотя эти методы эффективны, они являются не очень простыми в плане реализации не просты; они часто включают сложные алгоритмы оптимизации, что приводит к тому, что они с трудом справляются с 25-30 кадрами в секунду, производимыми многими современными камерами (Таблица 1).

Таблица 1. Характеристики алгоритмов отслеживания

Алгоритм	Частота кадров
IVT [1]	7.5 fps
GBDL[2]	в реальном времени
MILTrack[3]	25 fps
FragTrack[4]	в реальном времени
MOSSE Filters	640 fps

В этой статье мы исследуем более простую стратегию отслеживания, где обнаружение производится адаптивными корреляционным фильтром, а отслеживание выполняется с помощью свертки. Методы создания фильтров, такие как обрезка шаблона с изображения, создают сильные вершины для цели, но также ложно реагируют на фон или перекрытие объекта. В результате такие методы не особенно устойчивы к изменениям внешнего вида цели и не справляются со сложными задачами отслеживания. Minimum Output Sum of Squared Error (MOSSE) (рассматриваемый в этой статье) создает фильтры, которые более устойчивы к изменениям внешнего вида и лучше различают цели и фон. В результате получается гораздо более сильная вершина, что приводит к меньшему времени обнаружения и меньшему количеству ошибок. Несмотря на простоту подхода, отслеживание на основе фильтра MOSSE хорошо работает при изменении поворота, масштаба, освещения и частичной окклюзии. Отношение вершины к боковому выступу (PSR), которое измеряет силу вершины корреляции, может использоваться для обнаружения окклюзий или сбоя отслеживания, если объект снова появится с аналогичным внешним видом. В более общем плане, этот усовершенствованный корреляционный фильтр обеспечивает производительность, соответствующую более сложным трекарам, упомянутым ранее; однако подход, основанный на фильтрах, в 20 раз быстрее и может обрабатывать 640 кадров в секунду (Таблица 1).

1. Отслеживание на основе корреляционного фильтра

Под определением «корреляционно алгоритмы» понимается группа алгоритмов поиска объекта на изображении, основанных на использовании эталонного изображения объекта слежения. В научной литературе чаще всего приводится описание нескольких видов корреляционных алгоритмов, отличающихся способом вычисления критериальной функции и методом ее анализа. Часто используются разностные критериальные функции, как одни из наиболее простых для аппаратной реализации. Реже применяются алгоритмы, в которых находится глобальный минимум критериальной функции, вычисленной с использованием произведений эталонного и анализируемого изображений. Кроме того, используются нормированные критериальные функции.

Принцип работы трекеров основанных на таких фильтрах заключается в то что трекер моделирует внешний вид объекта, используя фильтры, обученные на примерах изображений. Выбор цели для отслеживания чаще всего изначально происходит выбирается на основе небольшого окна отслеживания, центрированного на объекте слежения в первом кадре видеопотока. С этого момента обучение отслеживанию и фильтрации работает совместно. Отслеживание цели происходит путем сопоставления фильтра с окном поиска в следующем кадре; потом по соответствующему максимальному значению в выводе корреляции, указывается новое положение цели. После этого происходит обновление на основе полученного ранее нового местоположения.

Чтобы создать быстрый в плане времени вычисления трекер корреляция вычисляется в области Быстрого Преобразования Фурье (БПФ) [15]. Вначале вычисляется 2D преобразование Фурье входного изображения: $F=F(f)$ и фильтра $H=F(h)$. Теорема Свертки говорит о том, что при подходящих условиях преобразования Фурье, свертки двух сигналов является точечным произведением их преобразований, и это ведет к тому, что корреляция становится элементарным умножением в области Фурье. Используя символ \odot для явного обозначения поэлементного умножения и $*$ для обозначения комплексного сопряженного, корреляция принимает вид:

$$G = F \odot H^* \quad (1)$$

Корреляционный вывод преобразуется обратно в пространственную область с использованием обратного БПФ. Узкое место в этом процесс вычисляется прямым и обратным БПФ так, чтобы весь процесс имел время верхней границы $O(P \log P)$, где P -количество пикселей в окне отслеживания.

Но при работе с алгоритмом свертки БПФ возникает проблема, которая исходит из того, что изображение и фильтр сопоставляются с топологической структурой тора. Поэтому при искусственного соединения краев изображения во время свертки создается артефакт, который влияет на результат корреляции изображения. Этот эффект можно уменьшить при выполнении предварительной обработки, в которой значения пикселей преобразуются с помощью функции регистрации в один набор координат, при которых значения пикселей нормализуются, чтобы иметь среднее значение 0,0 и 1,0. А дальше мы умножаем изображение на косинусное окно, которое постепенно уменьшает значения вблизи края изображения до нуля, что приводит к большему акценту вершины центра цели.

2. Фильтр MOSSE

MOSSE-это алгоритм для создания фильтров, подобных ASEF, из меньшего количества обучающих изображений. Для начала его работы требуется набор обучающих изображений f_i и обучающих выходов g_i . Как правило, g_i может принимать любую форму. В этом случае g_i генерируется из наземной истины таким образом, что он имеет компактный ($\sigma =$

2,0) 2D вершина в форме Гаусса, центрированный на цели на обучающем изображении f_i . Дальнейшее обучение проводится в области преобразования Фурье. Преобразование Фурье будет иметь вид:

$$H_i^* = \frac{G_i}{F_i}. \quad (2)$$

Чтобы найти фильтр, который сопоставляет входные данные обучения с желаемыми выходными данными обучения, MOSSE находит значение фильтра H , который минимизирует сумму квадратов ошибок между фактическим выходом свертки и желаемым выходом свертки. Первым шагом в поиске фильтра MOSSE является настройка задачи минимизации, которая будет оптимизирована:

$$H = \min_{H_i} \sum_i |F_i \odot H^* - G_i|^2. \quad (3)$$

В (3) F_i и G_i -входные изображения и соответствующие желаемые выходные данные в области Фурье, и цель состоит в том, чтобы найти фильтр H , который минимизирует сумму квадратов выходной ошибки. Поскольку корреляция в области Фурье представляет собой поэлементное умножение, каждый элемент фильтра H может быть оптимизирован независимо. Таким образом, задача оптимизации может быть преобразована из задачи многомерной оптимизации в задачу, которая оптимизирует каждый элемент H независимо.

$$H_{\omega v} = \min_{H_{\omega v}} \sum_i |F_{i\omega v} H_{\omega v}^* - G_{i\omega v}|^2, \quad (4)$$

где ω и v индексы H .

Эта функция является вещественной, положительной и выпуклой, поэтому у нее будет только один оптимум. Обычно для нахождения оптимумов функции, стабильные точки находят, устанавливая производную равной нулю, а затем решая для интересующей переменной. Нахождение стабильной точки для этой функции отличается тем, что она является вещественной функцией комплексной переменной. Необходимо позаботиться о том, чтобы правильно решить эту проблему. Оказывается, найти устойчивые точки такой функции не так уж сложно. Подводя итог, можно сказать, что это включает в себя переписывание функции как в $H_{\omega v}$, так и в $H_{\omega v}^*$. Затем частное значение $H_{\omega v}^*$ устанавливается равным нулю, рассматривая H как независимую переменную.

$$0 = \frac{\partial}{\partial H_{\omega v}^*} \sum_i |F_{i\omega v} H_{\omega v}^* - G_{i\omega v}|^2. \quad (5)$$

Можно показать, что любая $H_{\omega v}$, удовлетворяющая этому уравнению, является стабильной точкой. Преобразование этого уравнения приводит к:

$$0 = \frac{\partial}{\partial H_{\omega v}^*} \sum_i (F_{i\omega v} H_{\omega v}^* - G_{i\omega v})(F_{i\omega v} H_{\omega v}^* - G_{i\omega v})^*; \quad (6)$$

$$0 = \frac{\partial}{\partial H_{\omega v}^*} \sum_i [(F_{i\omega v} H_{\omega v}^*)(F_{i\omega v} H_{\omega v}^*)^* - (F_{i\omega v} H_{\omega v}^*)G_{i\omega v}^* - G_{i\omega v}(F_{i\omega v} H_{\omega v}^*)^* + G_{i\omega v}G_{i\omega v}^*]; \quad (7)$$

$$0 = \frac{\partial}{\partial H_{\omega v}^*} \sum_i F_{i\omega v} F_{i\omega v}^* H_{\omega v}^* H_{\omega v} - F_{i\omega v} G_{i\omega v}^* H_{\omega v}^* - F_{i\omega v}^* G_{i\omega v} H_{\omega v} + G_{i\omega v} G_{i\omega v}^* . \quad (8)$$

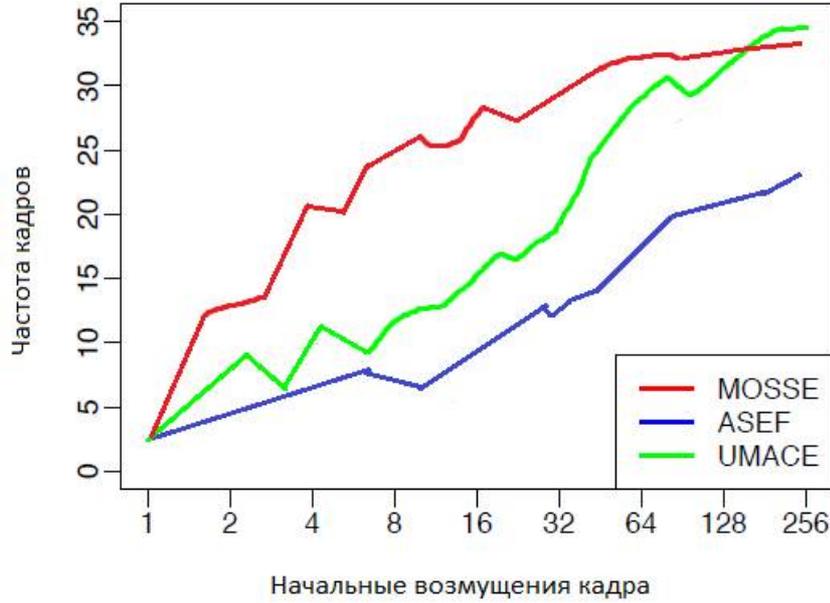


Рис. 1. Результаты сравнения фильтра MOSSE и традиционных

Вычисление частной производной приводит к:

$$0 = \sum_i [F_{i\omega v} F_{i\omega v}^* H_{\omega v} - F_{i\omega v} G_{i\omega v}^*] . \quad (9)$$

Затем мы можем провести суммирование и решить (9) для $H_{\omega v}$:

$$H_{\omega v} = \frac{\sum_i F_{i\omega v} G_{i\omega v}^*}{\sum_i F_{i\omega v} F_{i\omega v}^*} . \quad (10)$$

Наконец, мы перепишем это выражение как:

$$H = \frac{\sum_i F_i \odot G_i^*}{\sum_i F_i \odot F_i^*} . \quad (11)$$

При этом в уравнение (5) для фильтра MOSSE числитель обозначает корреляцию между входным сигналом и желаемым выходом, а знаменатель — энергетический спектр входного сигнала.

Разница между этим фильтром и традиционной реализацией заключается в том, что здесь мы обрабатываем, а затем переводим, а в традиционном вначале перевод, а затем обработка. Чтобы показать, что MOSSE производит лучшую фильтрацию, чем традиционные в качестве примера был проведен эксперимент, в ходе которого варьировалось количество изображений, используемых для обучения фильтров. И в результате на Рисунке 1 показано, что MOSSE создает лучшую фильтрацию при обучении.

3. Реализация фильтра и результаты

Первоначально была создана система отслеживания наземных объектов в видеопотоке на основе MOSSE, которая оценивалась по видео сохранённого с тепловой камеры вертолета. Такое использование позволило протестировать небольшие изменения в конфигурации трекера, и помогло выполнить качественный анализ производительности трекера для различных целей и условий отслеживания. Эти тесты предоставили ценную информацию о работе трекера с фильтром MOSSE представленного в этой статье. Сама камера движется во всех видеороликах, что усиливает беспорядочное движение целей.



Рис. 2. Захват объекта на сопровождение



Рис. 3. Дальнейшее сопровождение объекта



Рис. 4. Сопровождение объекта при перекрытии

Заключение

В этой статье показано, что проблема визуального отслеживания, которая традиционно решается с использованием классификаторов с большим весом, сложных моделей внешнего вида и методов стохастического поиска, может быть заменена эффективными и более простыми корреляционным фильтром MOSSE. В результате получается алгоритм, который прост в реализации, может быть столь же точным и намного быстрее. В этой статье трекер был упрощен для оценки способности фильтра следовать и адаптироваться к сложным сценариям отслеживания. Существует несколько простых способов улучшения этого трекера. Например, если внешний вид цели относительно стабилен, дрейф может быть уменьшен путем случайной повторной настройки фильтра на основе начального кадра. Также, нужно будет учитывать влияние таких факторов, как положение аппарата с которого производится обнаружение и сопровождение, текущая видимость, поле зрения системы ви-

дения, влияние рельефа. И в будущем трекер на основе этого фильтра можно рекомендовать к использованию в авионике. Также существуют идеи использования фильтра MOSSE в будущем для обнаружения и сопровождения в морской авионике.

Библиографический список

1. Росс Д., Лим Дж., Лин Р., Янг М. Инкрементное обучение для надежного визуального отслеживания // *ICCV*, 77(1):125–141, 2008.
2. Чжан Х., Ху У., Мейбанк С., Ли Х. Различительное обучение на основе графиков для надежного и эффективного отслеживания объектов // *ICCV*, 2007.
3. Адам А., Ривлин Э., Шимшони И. Надежное отслеживание фрагментов с использованием интегральной гистограммы // *CVPR*, 2006.
4. Бабенко Б., Янг М.-Х., Конни С. Визуальное отслеживание с онлайн-обучением несколькими экземплярами // *CVPR*, 2009.
5. Алпатов Б.А., Бабаян П.В., Балашов О.Е., Степашкин А.И. Методы автоматического обнаружения и сопровождения объектов. Обработка изображений и управление // *Радиотехника*, 2008. 176 с.
6. Анштедт Т., Келлер И., Лутц Х. Видеоаналитика: мифы и реальность [Текст] // *Security Focus*, 2012 г. – 176 с.
7. Форсайт Д., Понс Ж. Компьютерное зрение. Современный подход. // Издательский дом «Вильямс», 2004. 928с.: ил.
8. Хенрик Б., Джозеф Р., Марк Ф. Машинное обучение. // Издательский дом «Питер», 2017. 338с.: ил.

УДК 002; ГРНТИ 20.01.04

ПРИМЕНЕНИЕ ПРОЦЕДУРЫ ПРОКТОРИНГА ПРИ ПРОВЕДЕНИИ ВСТУПИТЕЛЬНЫХ ИСПЫТАНИЙ В ДИСТАНЦИОННОЙ ФОРМЕ В РГРТУ

Р.В. Хруничев

*Рязанский государственный радиотехнический университет имени Уткина,
Россия, Рязань, hrunichev_robert@mail.ru*

Аннотация. Представлен обзор систем прокторинга, необходимость и условия их применения в том числе при проведении процедуры дистанционного тестирования, а также опыт организации и проведения вступительных испытаний в РГРТУ в период проведения приемных кампаний при приеме на обучение по образовательным программам высшего образования – программам подготовки бакалавриата, специалитета, магистратуры.

Ключевые слова: прокторинг, системы прокторинга, вступительные испытания, дистанционные образовательные технологии, приемная кампания.

APPLICATION OF THE PROCTORING PROCEDURE WHEN CONDUCTING ENTRANCE TESTS IN A REMOTE FORM AT RSREU

R.V. Khrunichev

*Ryazan State Radio Engineering University named after Utkina,
Russia, Ryazan, hrunichev_robert@mail.ru*

Abstract. The article presents an overview of proctoring systems, the need and conditions for their application, including during the remote testing procedure, as well as the experience of organizing and conducting entrance tests at RSREU during the admission examinations for admission to higher education educational programs - bachelor's degree, specialty, master's degree programs.

Key words: proctoring, proctoring systems, entrance tests, distance education technologies, admission campaign.

В последние годы цифровизация всех отраслей народного хозяйства приобрела стремительный характер. Дополнительным стимулом к ускорению процессов внедрения информационно-коммуникационных технологий послужила пандемия.

Национальный проект Правительства Российской Федерации «Цифровая экономика» предусматривает внедрение цифровых технологий для повышения конкурентоспособности страны и улучшению качества жизни населения, что неизбежно накладывает требования к уровню владения соответствующими компетенциями [1].

Образование в структуре нацпроекта занимает особое место как направление программы «Кадры и образование», где предусмотрено стремительное внедрение IT-технологий не только в образовательном процессе, но и в процессе организации приема и управления образованием в целом.

Одновременно с этим одной из целей нацпроекта обозначено создание системы правового регулирования цифровой экономики [1]. Однако, на данный момент в области образования многие информационные процессы должным образом не обеспечены нормативно-правовой базой для регулирования вопросов защиты персональных данных в соответствии с Федеральным законом №152 от 27 июля 2006 года «О персональных данных».

В условиях пандемии при организации приема граждан РФ и иностранных граждан на обучение по образовательным программам высшего образования – программа бакалавриата, специалитета, магистратуры и аспирантуры основные изменения в процедуре проведения приемной кампании, касающиеся внедрения информационно-коммуникационных и дистанционных образовательных технологий были регламентированы Минобрнауки России в 2020 году Приказом №547 от 03 апреля 2020 года «Об особенностях приема на обучение по образовательным программам высшего образования – программам бакалавриата, программам специалитета, программам магистратуры, программам подготовки научно-педагогических кадров в аспирантуре на 2020/2021 учебный год» (далее – Особенности) [2].

Особенностями и последующими нормативно-правовыми актами, изданными Минобрнауки России процедуры, связанные с организацией и проведением вступительных испытаний в дистанционной форме никак не регламентированы, кроме того определялось, что вузы самостоятельно должны разрабатывать локальные нормативные акты для регулирования этих процессов.

Изменения в организации приема Особенности также предусмотрены в способах подачи документов для лиц, поступающих на программы высшего образования – программы бакалавриата и специалитета, магистратуры и аспирантуры.

Среди способов подачи документов предусмотрены следующие:

1. в электронной форме посредством электронной информационной системы организации;
2. с использованием суперсервиса «Поступление в вуз онлайн» посредством федеральной государственной информационной системы «Единый портал государственных и муниципальных услуг (функций)»;
3. лично поступающим.

Право выбора способа подачи документов остается за абитуриентом, а проведение вступительных испытаний в данный момент не исключает дистанционный формат для определенных групп поступающих, которым Правилами приема предоставлено такое право. Однако в 2020 году прием вступительных испытаний мог быть организован исключительно с применением дистанционных образовательных технологий [2].

Процедура идентификации личности и использование систем прокторинга при проведении вступительных испытаний также возложена на вузы. В 2020 году Особенности полностью исключался контакт поступающего с представителями вуза как на этапе подачи документов, так и на этапе организации и проведения вступительных испытаний. Любое взаимодействие с поступающим осуществлялось с применением средств ИКТ в дистанционном

режиме. Режим индивидуальных консультаций по предварительной записи сохранялся и в условиях пандемии.

Для обеспечения возможности организации и проведения вступительных испытаний в дистанционной форме вопрос идентификации личности поступающих решался одним из двух способов:

1. по предварительной записи, поступающий посещал приемную комиссию, где оставлял скан-копию документа, удостоверяющего личность;
2. идентификацию личности проводил нотариус в процессе нотариального заверения копии документа, который в последствии направлялся в приемную комиссию почтовым отправлением.

Вопрос прокторинга, как процедуры контроля за ходом дистанционного испытания на основе онлайн-наблюдения, протоколирования и оценивания поведения пользователей при прохождении онлайн мероприятий, при проведении вступительных испытаний оставался открытым, поскольку организация такой процедуры в сжатые сроки не представляется возможной. На сегодняшний день на рынке программных продуктов предлагается широкий спектр систем прокторинга, которые в асинхронном и синхронном режиме позволяют обеспечить чистоту процедуры проведения вступительных испытаний, среди них [3]:

- 1) Proctor U – стоимость предоставляется только по запросу, обеспечивает дистанционный прокторинг с контролем использования программного обеспечения, но сильно нагружает ЭВМ;
- 2) Examus – стоимость предоставляется только по запросу, позволяет анализировать онлайн-поведение экзаменуемого, обеспечивает защиту пользовательских данных, масштабируемо, требует невысокой пропускной способности канала, но имеет ограниченную поддержку и не подходит при небольшом количестве экзаменуемых;
- 3) AIProctor – стоимость около \$5,5 за один экзамен, есть бесплатная пробная версия, эффективно отслеживает онлайн-поведение, доступно в виде web-приложения, но значительная стоимость, закрытый исходный код, нет мобильной платформы;
- 4) Merittrac – стоимость предоставляется по запросу, имеет облачную и SaaS реализацию, автоматизированный прокторинг, автоматическая оценка действий экзаменуемого, но поставляется в ограниченном функционалом и не имеет бесплатной версии;
- 5) Examity – стоимость предоставляется по запросу, поддерживает мобильное и web-приложение, настраиваемое, но не предусматривает самообучение;
- 6) Talview – стоимость предоставляется только по запросу, есть бесплатная пробная версия, стандартизация процессов оценки, автоматизированный анализ компетенций, но неудобный пользовательский интерфейс, длительное время создания отчетов.

Как видно, представленные на рынке системы прокторинга не являются интеллектуальной собственностью компаний РФ, стоимость обеспечения процедуры прокторинга при проведении вступительных испытаний является значительной.

Следует отметить, что широкий спектр систем прокторинга представлен на программном рынке лишь в последнее время, стремительное развитие которых связано с ширококомасштабным применением дистанционных образовательных технологий во всем мире в условиях пандемии. В 2020 году выбор таких систем был ограничен.

В РГРТУ в 2020 году системы прокторинга при проведении вступительных испытаний не использовались, а контроль за проведением экзаменов возлагался на прокторов, которые в режиме реального времени осуществляли идентификацию, взаимодействие и контроль за соблюдением регламента проведения вступительных испытаний.

Организация вступительных испытаний с применением дистанционных образовательных технологий в поточном режиме не могла быть реализована, поэтому они проводились в малых группах по 25-30 человек в расчете на одного проктора на экзамене.

Вступительные испытания были организованы на платформе zoom [4]. Для поступающих использование данного сервиса бесплатно, платная учетная запись использовалась приемной комиссией для обеспечения возможности длительной (более 40 минут) процедуры проведения вступительных испытаний с обязательной процедурой видеofиксации в соответствии с указанием Минобрнауки России.

Отдельно следует рассмотреть процедуру подготовки к обеспечению возможности проведения вступительных испытаний с применением дистанционных образовательных технологий. Для поступающих на программы бакалавриата и специалитета возникла необходимость создания программ и разработки заданий дистанционного тестирования. Базы тестовых задач были размещены в системе управления обучением на платформе moodle [5]. Для поступающих на программы магистратуры и аспирантуры формой проведения вступительных испытаний является письменный экзамен, в связи с этим, требованием к абитуриенту в условиях дистанционной формы проведения экзамена было наличие оргтехники, позволяющей обеспечить должное качество сканирования (фотографирования) с машиночитаемым распознаванием его реквизитов. Задания вступительных испытаний – вопросы к письменному экзамену также были размещены на платформе moodle, но с иной формой организации обратной связи с абитуриентом.

В данное время обеспечение использования дистанционных технологий при организации приема документов и проведения вступительных испытаний в вузах стало обязательным и является неотъемлемой частью проведения приемных кампаний. Однако использование дистанционных технологий не является теперь исключительным, и абитуриент вправе самостоятельно определять более удобный способ взаимодействия с университетом на этапе поступления.

Библиографический список

1. Цифровая экономика – URL: <https://xn--80aarpampemcchfmo7a3c9ehj.xn--p1ai/projects/tsifrovaya-ekonomika> (Дата обращения: 24.02.2022)
2. Официальный интернет-портал правовой информации. Приказ Министерства науки и высшего образования Российской Федерации от 03.04.2020 № 547 "Об особенностях приема на обучение по образовательным программам высшего образования - программам бакалавриата, программам специалитета, программам магистратуры, программам подготовки научно-педагогических кадров в аспирантуре на 2020/21 учебный год" – URL: <http://publication.pravo.gov.ru/Document/View/0001202004130056>. (Дата обращения: 24.02.2022)
3. 10 лучших программ года для онлайн-прокторинга – особенности, обзоры и цены – URL: https://softwareunplugged.com/review/the-10-best-online-proctoring-sofware-of-the-year-features-reviews-and-pricing/?utm_source=offpage&utm_medium=offpage (Дата обращения: 24.02.2022)
4. Сервис zoom – URL: <https://zoom.us/PRICING>. (Дата обращения: 26.02.2022)
5. Система управления обучением moodle – URL: <https://moodle.org/?lang=ru> (Дата обращения: 26.02.2022)

УДК 621.397; ГРНТИ 47.63

ОБ АКТУАЛЬНОСТИ СОВМЕЩЕНИЯ РЕАЛЬНЫХ И ВИРТУАЛЬНЫХ ИЗОБРАЖЕНИЙ В СИСТЕМАХ ТЕХНИЧЕСКОГО ЗРЕНИЯ ЛЕТАТЕЛЬНЫХ АППАРАТОВ

А.В. Левитова, А.А. Логинов

*Рязанский государственный радиотехнический университет имени В.Ф. Уткина,
Российская Федерация, Рязань, levitova.nastya99@gmail.com, loginal@mail.ru*

Аннотация. В данной работе рассматривается совмещение реальных и виртуальных изображений. Приводятся основные цели совмещения изображений и области применения совмещения изображений.

Ключевые слова: летательный аппарат (ЛА), реальное изображение (РИ), виртуальное изображение (ВИ), цифровая карта местности (ЦКМ), система технического зрения (СТЗ), корреляционно-экстремальные навигационные системы (КЭНС).

ABOUT RELEVANCE OF COMBINING REAL AND VIRTUAL IMAGES IN FLYING MACHINE OBJECTS TECHNICAL VISION SYSTEMS

A.V. Levitova, A.A. Loginov

*Ryazan State Radio Engineering University named after V.F. Utkin,
Russia, Ryazan, levitova.nastya99@gmail.com, loginal@mail.ru*

The summary. This paper considers the combination real and virtual images. The main goals of image alignment and areas of application of image alignment are given.

Keywords: flying machine (FM), real image (RI), virtual image (VI), digital terrain map (DTM), technical vision system (TVS), correlation-extreme navigation systems (CENS).

Данная статья посвящена обсуждению актуальности задачи совмещения реальных и виртуальных изображений (РиВи) в авиационных системах технического зрения (СТЗ) и основана на результатах научных исследований, проводимых научно-образовательным центром «СпецЭВМ» кафедры ЭВМ РГРТУ в данном и смежных направлениях и на международном опыте использования таких систем на летательных аппаратах (ЛА) различного типа.

Определение термина совмещение реальных и виртуальных изображений

Реальное изображение (РИ) – изображение, получаемое от сенсоров СТЗ летательного аппарата (телекамера, тепловизор, лидар, радиолокационная станция (РЛС)) в реальном времени.

Виртуальное изображение (ВИ) – изображение поверхности, синтезированное по трёхмерной цифровой карте местности (ЦКМ), навигационно-пилотажным данным ЛА и направлениям осей визирования сенсоров, с учётом их технических характеристик, например, углов зрения.

Результат совмещение реальных и виртуальных изображений на борту ЛА может быть использован в двух аспектах [1].

Первый, позволяет за счёт создания корреляционно-экстремальной навигационной системы (КЭНС) получить ещё один источник определение координат ЛА, что особенно важно при постановке помех спутниковой навигационной системе (СНС). Совместная комплексная обработка информации (КОИ) поступающей от всех навигационных приборов комплекса навигации ЛА, таких, как инерциальная навигационная система (ИНС), СНС, система воздушных сигналов (СВС), доплеровский измеритель скорости и угла сноса (ДИСС), радиовысотометр (РВ), магнитометр и т.д. позволяет получить более точные и достоверные данные о пространственном и угловом положении ЛА. В этом аспекте КЭНС используется как ещё один независимый автономный источник координат ЛА. Наличие нескольких источников одних и тех же навигационных данных обеспечивает высокую живучесть комплекса

навигации. При выходе из строя отдельных источников информации снижаются лишь точностные показатели без потери основных функций. Кроме этого, наличие нескольких источников одних навигационных данных позволяет использовать специальное программно-математическое обеспечения, основанное на оптимальных алгоритмах комплексной обработки инерциальной, воздушной, радиотехнической, спутниковой, корреляционно-экстремальной и другой навигационной информации. Алгоритмы базируются на Калмановской фильтрации погрешностей измерения навигационных параметров и уменьшают погрешности за счет исключения их систематической составляющей. Алгоритм КОИ построен на основе оптимальной обработки избыточной навигационной информации в процессе полета. В качестве алгоритма оценки используется фильтр Калмана, осуществляющий оптимальную обработку разностей параметров (географических координат и составляющих скорости), численных ИНС, измеренных СНС, КЭНС и другими системами. Алгоритм позволяет идентифицировать источники погрешностей автономного счисления и осуществлять компенсацию погрешностей различных навигационных систем в определении скорости и курса при отсутствии информации от СНС (при «затенении» антенны СНС на участках маневрирования, при отказе СНС или постановке помех). При использовании КОИ появляется возможность иметь высокоточную и надежную информацию по координатам, составляющим скорости и курсу.

Второй аспект связан с приборной визуализацией для экипажа ЛА закабинного пространства по данным СТЗ и ЦКМ. Это очень важно в плохих условиях видимости, так как значительно увеличивает безопасность пилотирования особенно на низких высотах и посадке. Формирование комбинированного изображения состоит из двух этапов:

- геометрическое совмещение РиВИ с учётом трёхмерности поверхности;
- слияние геометрически совмещенных РиВИ, которое должно обеспечить наилучшую эргономичность итогового комбинированного изображения для увеличения ситуационной осведомлённости экипажа ЛА или оператора наземного пункта управления беспилотным ЛА.

По каждому сенсору СТЗ и данным ЦКМ формируется соответствующее изображение, которое помогает экипажу ориентироваться в изменяющейся окружающей обстановке при полете и посадке. Комбинированное изображение, синтезированное по данным от СТЗ и ЦКМ, позволяет экипажу не анализировать каждое изображение в отдельности для принятия адекватного решения.

Два этапа формирования комбинированного изображения очень отличаются по трудоёмкости [2]. Задачу геометрического совмещения значительно сложнее эффективно реализовать в реальном времени, чем слияние.

Несомненно, что геометрическое совмещение РиВИ является также основным и в КЭНС, но коррекция навигационных параметров может проводиться значительно реже, чем частота формирования комбинированного изображения для экипажа ЛА. Поэтому, наиболее критическим этапом является геометрическое совмещение РиВИ для второго аспекта совмещение реальных и виртуальных изображений на борту ЛА, то есть при решении задачи визуализации закабинного пространства. В этом плане актуальность совмещения реальных и виртуальных изображений в СТЗ ЛА следует в первую очередь анализировать для решения задачи визуализации итогового комбинированного изображения.

Следует отметить, что не все варианты (способы) получения комбинированного изображения требуют точного геометрического совмещения РиВИ.

Необходимость геометрического совмещения РиВИ связана с наличием погрешностей в пространственном и угловом положении сенсоров. Вследствие чего, ракурс виртуального

сенсора не совпадает с реальным. Иногда задачу геометрического совмещения РиВИ называют досовмещением РиВИ.

Цель (назначение) совмещения изображений

В широком смысле цель совмещения - увеличение ситуационной осведомленности экипажа или оператора в различных условиях видимости.

В узком смысле, когда речь идёт только о геометрическом совмещении, целью является попиксельное сопоставление РиВИ с учётом трёхмерности поверхности с допустимой погрешностью [3], для их последующего слияния. Исходное виртуальное изображение может значительно не совпадать с реальным, в первую очередь, из-за погрешностей в навигационно-пилотажных данных ЛА, во вторую, из-за погрешностей позиционирования сенсоров СТЗ относительно ЛА и погрешности построения ЦКМ.

Погрешности совмещения изображений зависит от типа сенсора

Для лидара и РЛС погрешность слабо зависит от координатных и угловых погрешностей позиционирования локатора и самое главное не зависит от высоты, вследствие того, что РИ, получаемые от них, имеют координату наклонной дальности.

Для ТВ и ТПВ камер погрешность очень сильно зависит от координатных и угловых погрешностей позиционирования сенсоров и растёт с уменьшением высоты. Кроме этого, с уменьшением высоты уменьшается площадь обзора реальной поверхности и соответственно виртуальной, что приводит к катастрофической потере информативности РиВИ, так необходимой для надежного досовмещения РиВИ.

Области применения совмещения изображений

1. Посадка. Точность и гарантированность совмещения определяют насколько можно доверять ВИ взлетно-посадочной полосы (ВПП), не видя РИ ВПП (или посадочной площадки).
2. Маловысотный полет, полет в рельефе.
3. Уменьшение погрешности определение географических координат некартографических объектов на поверхности.
4. Коррекция координат ЛА на базе технологии КЭНС.

Для первой и второй задачи совмещение актуально только в плохих условиях видимости, когда плохое РИ дополняется ВИ.

Третья задача решается только в хороших условиях технического зрения (хорошее РИ).

На посадке при использовании современных точных навигационных систем (ИНС+СНС) и наличии качественной ЦКМ досовмещение РиВИ для лидара и РЛС в принципе не требуется на любой высоте, а для ТВ и ТПВ камер на высоте не менее 150-200 м.

Расхождение между ТВ/ТПВ изображением и ВИ на высоте менее 100 м, когда нужно принимать решение об уходе на второй круг, достигает недопустимых значений. Казалось бы здесь самый актуальный момент решения задачи досовмещения РиВИ.

Но если техническое зрение (РИ) на данных высотах позволяет принимать решение о посадке, то ВИ уже не нужно. Наличие только ВИ и отсутствие РИ не позволяет принимать решение о продолжении посадки. Недостаточное совмещенное комбинированное изображение на высотах 60-150 м не вызовет затруднений у экипажа, так как доля РИ в комбиниро-

ванном изображении при снижении постоянно растет, а доля несовмещенного ВИ уменьшается до нуля.

Нахождение ВПП на траверзе ЛА с необходимой погрешностью обеспечивается использованием только ВИ на высоте не менее 150-200 м (3 км до границы ВПП, 30 сек до касания ВПП).

Для стационарных ВПП и посадочных площадок комбинированное видение, основанное на совмещении ТВ/ТПВ и виртуального изображений даже для небольших высот, эффективно решается путем формирования ВИ не по ЦКМ, а по ортофотопланам, полученным по предварительным аэрофотосъемкам. Это позволяет заранее выбрать на ВИ характерные точки и затем искать их на РИ, с последующим применением для совмещения РиВИ не корреляционных методов, а методов многопроекционной геометрии, которые легко реализуются в реальном времени.

Маловысотный полет, полет в рельефе эффективно обеспечиваются для вертолета с помощью лидара или РЛС для самолета с помощью РЛС без решения задачи досовмещения РиВИ, так как расхождение РиВИ находится в допустимых пределах для этих сенсоров.

Применение комбинированного ТВ/ТПВ и виртуального изображений для решения этой задачи на очень малых высотах 15-50 м недопустимо, то есть нельзя плохое РИ подменить или дополнять ВИ, так как они значительно не совпадают. Здесь можно работать только на базе РИ и задачу досовмещения решать не надо.

Уменьшение погрешности определение географических координат некартографических объектов на подстилающей поверхности при использовании современных точных навигационных систем (ИНС+СНС) с помощью совмещения РиВИ практически несущественно. Определение географических координат некартографических объектов на подстилающей поверхности может эффективно решаться только по данным навигационной системы и РИ.

Проведенные в НОЦ «СпецЭВМ» исследования различных алгоритмов совмещения РиВИ показывают, что вероятность правильного совмещения сильно зависит от конкретного сюжета подстилающей поверхности и для ТВ/ТПВ сенсоров от высоты полета. Для высоты полета более 500 м можно ориентировочно назвать вероятность равную 0,6-0,7.

Библиографический список

1. S. Elesina, O. Lomteva. Increase of image combination performance in combined vision systems using genetic algorithm // Proceedings of the 3rd Mediterranean Conference on Embedded Computing. –Budva, Montenegro.–2014. PP. 158-161.
2. Костяшкин Л.Н., Логинов А.А., Никифоров М.Б. Проблемные аспекты системы комбинированного видения летательных аппаратов // Известия ЮФУ. Технические науки. 2013. № 5. С. 61-65.
3. Руководство Р-315 «По минимальным стандартам характеристик авиационных систем (MASPS) для систем улучшенного видения, систем искусственного видения, комбинированных систем искусственного видения и бортовых систем увеличения дальности видения» Авиационного Регистра Межгосударственного Авиационного Комитета (АР МАК).

УДК 001.85, ГРНТИ 47.43

СОВЕРШЕНСТВОВАНИЕ НОРМАТИВНО-МЕТОДИЧЕСКОЙ ДОКУМЕНТАЦИИ ПО ВОПРОСУ ПРОВЕРКИ ЭФФЕКТИВНОСТИ ЭКРАНИРОВАНИЯ ЭКРАНИРОВАННЫХ СООРУЖЕНИЙ

А.А. Конкин, Б.В. Костров, М.Б. Никифоров, М.С. Никифоров, Е.В. Лынов
*Рязанский государственный радиотехнический университет имени В.Ф. Уткина,
Российская Федерация, Рязань, nikiforov.m.s@grpz.ru*

Аннотация. В работе рассматривается проблема проведения оценки эффективности экранирования экранированных сооружений в соответствии с действующей нормативно-методической документацией, выявляются несоответствия в положениях нормативно-методических документов, и описываются предложения по их устранению.

Ключевые слова: экранированное сооружение, оценка эффективности экранирования, затухание сигнала, измерительное оборудование, нормативно-методическая документация.

IMPROVEMENT OF NORMATIVE AND METHODOLOGICAL DOCUMENTATION ON THE PERFORMANCE TEST SCREENING OF SCREENED STRUCTURES

A.A. Konkin, B.V. Kostrov, M.B. Nikiforov, M.S. Nikiforov, E.V. Lynov
*Ryazan State Radio Engineering University named after V.F. Utkin,
Russia, Ryazan, nikiforov.m.s@grpz.ru*

Annotation. The paper regulates the problems associated with evaluating the effectiveness of screening of shielded states in accordance with the current regulatory and methodological documentation, identifies inconsistencies in the provisions of regulatory and methodological documents, and the accepted proposals for their elimination.

Keywords: shielded structure, shielding efficiency assessment, signal attenuation, measuring equipment, regulatory and methodological documentation.

Достижение требуемого результата при выполнении поставленной задачи во многом зависит от задания требований к выполнению данной задачи, а также глубины проработки вопроса и достаточности его нормативного регулирования.

Сегодняшнее общество характеризуется быстрой изменчивостью. Если нормативные документы, разработанные в период 80-90х годов XX века оставались актуальными на протяжении многих десятилетий, то сегодня стандарты могут потерять свою актуальность за несколько лет. Следует отметить, что отсутствие актуальных стандартов существенно затрудняет осуществление работ по единому сценарию, а возникающие при проведении работ несоответствия зачастую устраняются не на основании юридически легитимного документа, а на уровне взаимных договорённостей или признания чьего-то авторитета (головного института, регулятора и т.д.), причём зачастую такие договорённости могут оказаться ни коим образом не задокументированы.

Постановка задачи

На протяжении всего жизненного цикла экранированного сооружения (далее - ЭС) необходимо подтверждать как эффективность его экранирования в целом, так и элементов его конструкций, в частности. Связано это с тем, что с течением времени за счет воздействия внешних факторов данная характеристика ухудшается, вследствие чего могут не соблюдаться требования в части эффективности экранирования, установленные для конкретного ЭС. Для подтверждения эффективности экранирования и дальнейшей беспрепятственной эксплуатации ЭС необходимо периодически проводить оценку эффективности экранирования как ЭС в целом, так и элементов его конструкции. С учетом неактуальности нормативно-методической документации по вопросам проведения оценки эффективности экранирования ЭС необходимо разработать способ проведения оценки эффективности экранирования ЭС,

который обеспечит достоверность измерений, а также позволит сократить временные затраты на калибровку измерительного оборудования при изменении рабочей частоты, на которой проводится проверка.

Порядок оценки эффективности экранирования описан и регламентирован действующим на территории Российской Федерации ГОСТ 30373-95/ГОСТ Р 50414-92 «Совместимость технических средств электромагнитная. Оборудование для испытаний. Камеры экранированные. Классы, основные параметры, технические требования и методы испытаний» (далее - ГОСТ). С момента издания указанного ГОСТа прошло уже более четверти века. Несмотря на кажущуюся простоту, в процессе проведения работ и практической оценки эффективности экранирования ЭС в соответствии с указанным ГОСТом был выявлен ряд положений, которые не были или на момент выхода ГОСТа просто не могли быть учтены в нём.

Перечислим некоторые из них:

- ГОСТ описывает проведение проверки эффективности экранирования ЭС в полосе частот от 0,01 до 37500 МГц, что не отвечает современным потребностям, так как ЭС на сегодняшний день могут использоваться для испытания изделий в полосе частот, выходящей за пределы полосы частот, описанной в ГОСТе;

- ГОСТом предусмотрено, что для ЭС должны быть предусмотрены технологические проходы шириной не менее 1 метра. В то же время не учитывается вариант расположения элементов ЭС вплотную к стене, что нередко встречается на практике, когда одна или несколько стен ЭК выполнены вплотную к ограждающей конструкции здания (пункт 2.1.4);

- ГОСТом не определён конкретный временной интервал проведения периодической проверки эффективности экранирования ЭС (пункт 3.1);

- Перечень рекомендуемых стандартных средств измерения, предусмотренных ГОСТом (приложения 4-7), нуждается в актуализации, так как приведенные в нем средства измерения на сегодняшний день являются устаревшими и слишком громоздкими, а некоторые и вовсе сняты с производства;

- ГОСТом не описан порядок проведения проверки эффективности ЭС при их облицовке или установке внутрь ЭС радиопоглощающего материала (далее - РПМ).

Теоретическая часть

С целью приведения положений стандарта к соответствию с задачами, решаемыми на практике в настоящее время, а также с целью сохранения актуальности положений стандарта в долгосрочной перспективе, предлагается провести его актуализацию. При этом в части корректировки положений стандарта по проблемным вопросам, описанным выше, авторы предлагают рассмотреть следующие решения:

1. Расширить полосу частот, в которой проводится проверка эффективности экранирования ЭС.

Уже сегодня Федеральное государственное унитарное предприятие «Всероссийский научно-исследовательский институт физико-технических и радиотехнических измерений» (ФГУП «ВНИИФТРИ») осуществляет работы по первичной, периодической и повторной аттестации безэховых и полубезэховых камер, безэховых экранированных камер, экранированных камер и сооружений, а также антенных и радиолокационных полигонов, открытых и альтернативных испытательных площадок. Работы по оценке эффективности экранирования проводят в диапазоне частот от 20 Гц до 110 ГГц. При этом нормативными документами, на которые ссылается ФГУП «ВНИИФТРИ» при проведении вышеуказанных работ являются ГОСТ Р 50414-92/ГОСТ30373-95, ГОСТ РВ 6601-001-2008, ГОСТ РВ 6601-002-2008, в которых не определен порядок проведения работ в указанном диапазоне.

2. Предусмотреть в стандарте случай расположения ЭС в помещении, при которой технологический проем между ЭС и ограждающими конструкциями зданий, в которых они

располагаются, будет отсутствовать, либо когда стеной/потолком ЭС будет выступать стена/потолок сооружения, в котором находится ЭС, облицованные экранирующим материалом.

На практике при проведении работ по проверке эффективности экранирования ЭС нередко возникают ситуации, когда ЭС расположено вплотную к ограждающим конструкциям здания. Такое расположение может быть следствием того, что расположение ЭС с соблюдением требований в части технологических зазоров между ЭС и ограждающими конструкциями было невозможным по каким-либо объективным причинам. В таком случае становится неочевидным, каким образом необходимо проводить работы по проверке эффективности экранирования ЭС.

3. Определить период проведения регламентных проверок для ЭС всех классов.

В настоящее время остается также не затронутым вопрос о периодичности проведения оценки эффективности экранирования ЭС. Такая неопределенность способствует тому, что организации, единожды проведя подобную работу и получив протокол, могут вполне легитимно предъявлять его в случае проверок. Однако, как уже было сказано ранее, с течением времени за счет воздействия внешних факторов эффективность экранирования ЭС ухудшается, и поэтому, чтобы поддерживать данную характеристику на необходимом уровне, следует определить период, в течение которого будет действовать протокол по результатам проверки эффективности экранирования ЭС и по истечении которого необходимо проводить повторную проверку.

При определении требований по периодичности проведения проверок эффективности экранирования ЭС, авторы предлагают принимать во внимание следующие факторы:

- материалы, из которого выполнено ЭС и его износостойкость;
- задачи, под которые спроектировано ЭС;
- частотный диапазон, работы в котором предусмотрены для данного ЭС;
- класс ЭС.

4. Актуализировать перечень стандартного измерительного оборудования, которое может быть использовано для проведения оценки эффективности экранирования ЭС.

Вследствие ввода в действие ГОСТа более четверти века назад, следует понимать, что указанное в нем рекомендуемое измерительное оборудование, большей частью, является устаревшим, а некоторое и вовсе снято с производства. В то же время, даже в случае наличия такого оборудования, возникает вопрос об удобстве его использования ввиду немалых габаритных размеров и сравнительно небольшого рабочего диапазона частот, что может породить ситуацию, при которой для выполнения работ по проверке эффективности экранирования ЭС может понадобиться по несколько единиц каждой позиции (генератор ВЧ-сигналов, измерительный приемник и антенны). Современное же измерительное оборудование проще в освоении, имеет более широкие рабочие диапазоны частот и сравнительно меньшие габаритные размеры.

Также предлагается наряду с рекомендованным перечнем измерительного оборудования привести перечень требований к оборудованию, используемому для проведения проверок эффективности экранирования ЭС, в котором будут указаны лишь характеристики такого оборудования в зависимости от класса ЭС. При этом единственным требованием к такому оборудованию должно быть включение его в Государственный реестр средств измерений и наличие у него свидетельства о поверке.

5. Описать порядок проведения измерения эффективности экранирования ЭС, облицованного изнутри РПМ.

Обычно, РПМ облицовывают безэховые и полубезэховые камеры на этапе строительства. Однако стоимость строительства новой безэховой и полубезэховой камеры может составлять от нескольких миллионов до нескольких миллиардов рублей в зависимости от задач, для решения которых строятся такие камеры. В таком случае нередко используют уже построенные ранее камеры, которые облицовывают изнутри РПМ для доведения характери-

стик ЭС до минимально необходимых. В то же время остается открытым вопрос о том, каким образом в дальнейшем подтвердить эффективность экранирования такой ЭС, облицованной изнутри РПМ.

6. Разработать новые методы проведения оценки эффективности экранирования ЭС.

Выполнение работ по проверке эффективности экранирования ЭС зачастую предполагает, что работник столкнется с описанными выше ситуациями. Этим фактом обуславливается необходимость разработки новых способов проведения оценки эффективности экранирования ЭС, в которых будет целесообразно учесть и минимизировать такие ситуации, а также уменьшить трудозатраты на выполнение работ.

Практическая часть

Ввиду наличия указанных недостатков становится очевидным тот факт, что необходимо пересмотреть требования и нормативную документацию в части проверки эффективности экранирования ЭС. Кроме того, совершенно очевидно, что на данный момент очень трудно учесть все возможные параметры существующих ЭС, поскольку различные ЭС проектируются и возводятся под выполнение конкретных задач, что в свою очередь делает невозможным предъявление единых требований как в части конструкции, так и в части проведения проверок. На такой случай необходимо также предусмотреть ряд общих требований, которые могут быть предъявлены к ЭС.

Нельзя также не отметить и тот факт, что методы проведения измерений также нуждаются в приведении к реалиям сегодняшнего дня, учитывая изменения, произошедшие с одной стороны в области развития контрольно-измерительной техники, используемой при проведении работ по оценке эффективности экранирования элементов ЭС, а с другой стороны изменения требований, предъявляемых заказчиком к техническим характеристикам самих ЭС. Оптимизация комплекса работ, проводимых с ЭС, может быть достигнута, в том числе, посредством применения стандартных функций современной измерительной аппаратуры, с целью сокращения времен проведения измерений при сохранении необходимой точности и достоверности результатов.

Так, коллективом авторов был предложен способ автоматизированного измерения уровней электромагнитных излучений в элементах конструкции экранированного сооружения в требуемой полосе частот.

В описанном авторами способе с помощью источника сигнала формируют сигнал с последующим его излучением посредством передающей антенны на элемент конструкции экранированного сооружения, с другой стороны которого с помощью приемной антенны принимают просочившийся сигнал, после чего выполняют его обработку, используя измеритель уровня электромагнитной энергии. При этом перед проведением измерений производят калибровку приемного тракта, при выполнении которой измеряют уровень (мощность) сигнала от источника сигнала, а также проводят теоретический расчёт затухания сигнала в свободном пространстве. Калибровка приёмного тракта предполагает измерение уровня (мощности) сигнала от источника сигнала, имитирующего работу изделия, его составной части или технологического оборудования, работы с которым проводятся в ЭС. Максимальное полученное значение уровня (мощности) сигнала принимается за рабочий уровень сигнала на расстоянии 1 метр от изделия.

Затем проводится теоретический расчёт затухания электромагнитного сигнала в свободном пространстве по формуле 1:

$$L_{\text{св.п}} = 20 \lg(d) + 20 \lg(f) + 32,45$$

где $L_{\text{св.п}}$ – затухание сигнала в свободном пространстве (dB);

d – расстояние от изделия до места установки антенны (км);

f – частота сигнала (МГц).

Суммарное затухание (L_{Σ}) от источника сигнала до приемной антенны будет рассчитываться по формуле 2:

$$L_{\Sigma} = L_{\text{ЭС}} + L_{\text{св.п}} + L_{\text{пт}}$$

где $L_{\text{ЭС}}$ – минимальное значение эффективности экранирования экранированного сооружения, обеспечивающее требуемую защиту от утечки сигнала;

$L_{\text{св.п}}$ – значение затухания сигнала в свободном пространстве;

$L_{\text{пт}}$ – значение затухания сигнала, вносимое приемным и передающим трактами, определенное при калибровке системы.

Полученное таким образом значение затухания сигнала L_{Σ} устанавливается как минимально необходимое для возможности проведения работ в ЭС.

При проведении измерений в качестве передающей и приемной антенн используют антенны с широкой диаграммой направленности, при этом излучаемый сигнал формируют на контрольных частотах из диапазона рабочих частот экранированного сооружения в соответствии с заданным интервалом времени, после чего измеряют мощность (уровень) просочившегося через элемент конструкции экранированного сооружения сигнала, принятого приемной антенной. Причем обработку просочившегося сигнала выполняют в режиме накопления, либо в режиме фиксации пикового уровня мощности.

Предлагаемый способ основан на передаче и приеме сигнала с помощью передающей и приемной антенн с широкой диаграммой направленности, находящихся по разные стороны элемента конструкции ЭС, как показано на рисунке 1, с последующим измерением уровня принятого сигнала. При этом передаваемый сигнал генерируют источником сигнала, подключенным к передающей антенне, а для измерения уровня принятого сигнала используют измеритель уровня электромагнитной энергии.

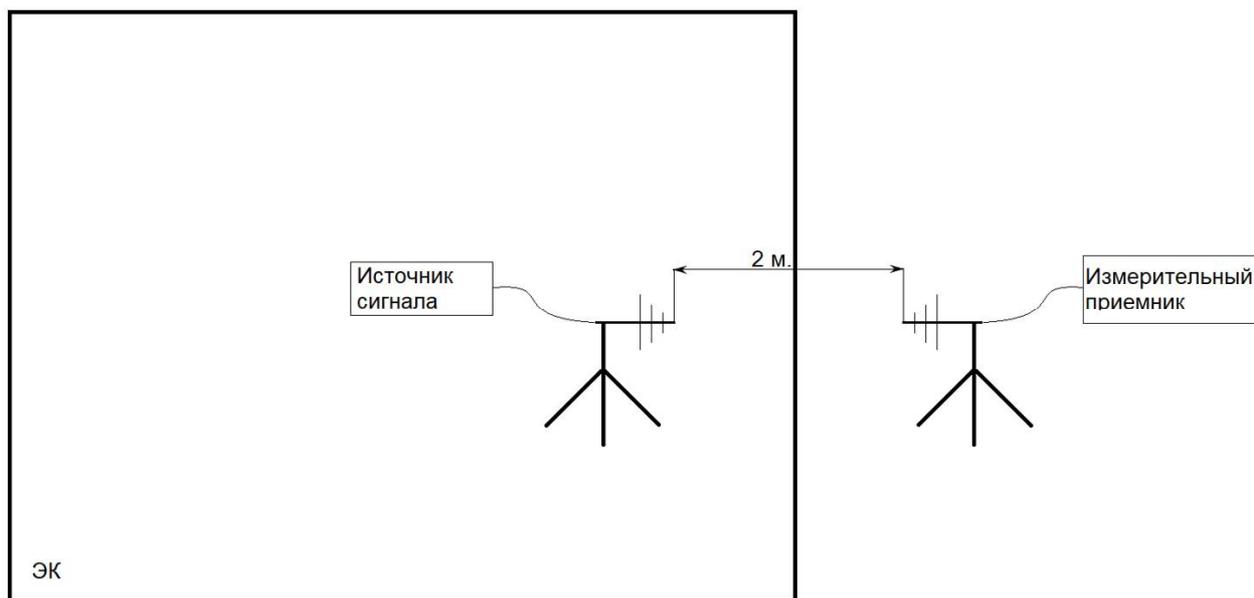


Рис. 1. Способ проведения измерений

Главным преимуществом описанного способа измерений является отсутствие необходимости повторной калибровки оборудования при смене рабочей частоты. Также, в отличие от описанных в работах других авторов, в частности способов определения величины реального затухания электромагнитного поля, реализованных «Автоматизированным комплексом определения коэффициента реального затухания электромагнитного поля при проведении

аттестационных испытаний объектов информатизации «ЗОНД», производства ЗАО НПЦ Фирма «Нелк», преимуществом является то, что в описанном способе отсутствует необходимость подключения измерительной аппаратуры на передающей и приёмной стороне с помощью соединительных кабелей к управляющей ими ПЭВМ, что повышает точность оценки эффективности экранирования ЭС в целом и элементов его конструкций, в частности, и достоверность определения величины реального затухания электромагнитного поля. Положительный эффект достигается за счёт исключения необходимости прокладки соединительного кабеля, связующего передающую и приёмную стороны, в процессе чего может быть нарушена радиогерметичность ЭС, либо, если для прокладки соединительного кабеля предполагалось применение специализированных радиочастотных фильтров, то отпадает необходимость их использования, что ведёт к упрощению конструкции ЭС. Кроме того положительным эффектом является оптимизация состава измерительной техники, применяемой при проверке ЭС, путем исключения из него ПЭВМ.

Заключение

В ходе написания данной работы авторами был проведен анализ нормативно-методической документации, а именно ГОСТ Р 30373-95/50414-92. В ходе анализа были выявлены положения указанного документа, требующие корректировки или доработки, предложены меры по корректировке документа. Основываясь на практическом опыте, авторами также был разработан способ автоматизированного измерения уровней электромагнитных излучений в элементах конструкции экранированного сооружения в требуемой полосе частот, на который впоследствии был получен патент РФ № 2761478.

Корректировка нормативно-методической документации по вопросам проведения оценки эффективности экранирования ЭС с учетом изложенных в данной работе замечаний поможет значительно снизить трудозатраты на выполнение таких работ и позволит минимизировать количество ситуаций, в которых применение нормативно-методической документации будет представляться невозможным ввиду отсутствия четко заданных требований и рекомендаций по проведению работ для различных условий и характеристик ЭС.

Библиографический список

1. ГОСТ Р 30373-95/50414-92 «Совместимость технических средств электромагнитная. Оборудование для испытаний. Камеры экранированные. Классы, основные параметры, технические требования и методы испытаний».
2. ГОСТ РВ 6601-001-2008 «Оборудование бортовое авиационное. Общие требования к восприимчивости при воздействии электромагнитных помех и методики измерения».
3. ГОСТ РВ 6601-002-2008 «Оборудование бортовое авиационное. Общие требования к допустимым уровням создаваемых электромагнитных помех и методики их измерений».
4. РЕКОМЕНДАЦИЯ МСЭ-R P.525-2* «Расчет ослабления в свободном пространстве» Электронный ресурс: https://www.itu.int/dms_pubrec/itu-r/rec/p/R-REC-P.525-2-199408-S!!PDF-R.pdf
5. Конкин А.А., Лынов Е.В., Никифоров М.С. «Способ автоматизированного измерения уровней электромагнитных излучений в элементах конструкции экранированного сооружения в требуемой полосе частот» Патент на изобретение 2761478 С1, 08.12.2021.
6. «Измерения характеристик безэховых камер» [Электронный ресурс] URL: <https://www.vniifri.ru/catalog/services/izmereniya-kharakteristik-bezekhovykh-kamer/>.

УДК 004.8; ГРНТИ 28.23

ОПИСАНИЕ КОНЦЕПЦИИ ПРИМЕНЕНИЯ МАШИННОГО ОБУЧЕНИЯ ПРИ ОБРАБОТКЕ БОЛЬШОГО ОБЪЕМА ГРАФИЧЕСКОЙ ИНФОРМАЦИИ

И.С. Панина, А.А. Выюгина, С.Н. Баранова

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

Аннотация. Целью работы является анализ возможностей применения технологий больших данных для методов машинного обучения. Многие задачи обработки изображений представляют собой их классификацию или определение объектов, которые могут решаться с помощью искусственного интеллекта. Однако с увеличением количества поступающих данных и требованиям ко времени обработки, появляется проблема масштабирования систем для решения поставленных задач. Возможным направлением развития систем является внедрение технологий больших данных для работы с обработкой изображений.

Ключевые слова: большие данные, машинное обучение, MapReduce, нейросети, обработка изображений.

DESCRIPTION OF THE CONCEPT OF USING MACHINE LEARNING IN PROCESSING A LARGE AMOUNT OF GRAPHICAL INFORMATION

I.S. Panina, 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. The purpose of the work is to analyze the possibilities of using big data technologies for machine learning methods. Many image processing tasks are their classification or the definition of objects that can be solved with the help of artificial intelligence. However, with an increase in the amount of incoming data and processing time requirements, there is a problem of scaling systems to solve the tasks. A possible direction for the development of systems is the introduction of big data technologies for working with image processing.

Keywords: big data, machine learning, MapReduce, neural networks, image processing.

Информация, которая воспринимается зрением, называется визуальной. Любое изображение является видом визуальной информации.

Изображение, то есть графические данные, а также связанные с ними функции или ограничивающие рамки и т. д., являются входной информацией в задачах обработки изображений, которая подразумевает работу как с аналоговыми, так и цифровыми изображениями. Такая обработка изображения включает в себя манипулирование цифровым изображением для генерации информации с помощью сложных алгоритмов.

Основными целями, которые ставятся при обработке изображений, являются:

- визуализации изображений, т.е. представления полученных или видоизмененных данных в виде визуального вывода для лучшего восприятия пользователем. Эта задача в основном выполняется для объектов, которые трудно различить на изображении;
- повышения качества изображения;
- поиска изображений;
- выполнения классификации для поиска различных объектов и определения их положения на изображении.

Таким образом, появляется возможность проведения дальнейшего анализа с использованием всей имеющейся и полученной разнородной информации.

Объем, разнородность и скорость получения графических данных дают основания причислять данный вид информации к большим данным, которые имеют свои особенности при работе. Стоит отметить, что процесс обработки больших данных состоит из нескольких этапов:

1. Постановка задачи анализа.
2. Сбор необходимых для исследования данных.

3. Подготовка данных к анализу.
4. Анализ данных.
5. Моделирование данных.
6. Обработка полученных результатов [1].

Наиболее затруднительными этапами считаются подготовка и анализ данных, которые, в зависимости от задачи, требуют большую точность и время обработки. С каждым годом объем информации в мире нарастает, однако ее ценность при этом зависит от скорости получения результатов, поэтому все большее количество систем стремится перейти к работе большими данными в режиме реального времени. Для этого одним из наиболее результативных методов обработки моделей считаются алгоритмы машинного обучения.

Среди наиболее востребованных технологий можно выделить глубокое обучение, обучение с подкреплением и механизмы коллективного принятия решений, коллективного управления.

Обучение с подкреплением дает мощный инструмент для создания систем, способных к самостоятельному поиску наиболее эффективных стратегий действия, самообучению, что позволяет создавать действительно автономные системы.

Несмотря на то, что искусственный интеллект с большим успехом применим фактически ко всем сферам человеческой деятельности, основной задачей создания специализированной программы расчета параметров моделей, является поиск эффективных алгоритмов оптимизации, которые давали бы наилучшие результаты в конкретной области применения.

При выборе алгоритма обработки модели необходимо полагаться на поставленную задачу и учесть ряд основополагающих факторов. Самым важным аспектом является вид исходных данных с учетом его особенностей, качество и полнота имеющихся сведений. Затем, необходимо определить критерии оценки точности, полноты и качества полученного результата.

Автоматическая обработка графической информации является одним из наиболее перспективных направлений в области искусственного интеллекта. Интерес к проблемам компьютерной обработки данных определяется расширением возможностей как самих компьютерных систем, так и разработкой новых технологий обработки, анализа и идентификации различных видов изображений. При этом для создания эффективных технологий разрабатываемые методы и алгоритмы должны удовлетворять ряду требований по быстродействию и точности.

При визуальной обработке данных часто встречаются выбросы или отклонения, которые существенно отличаются от большей части показателей и являются нежелательной характеристикой данных или процессов. Такие данные называются аномалиями. Существуют совершенно различные определения понятия «аномалия», используемые в различных прикладных задачах и зависящие от специфики данных [2].

Обнаружение выбросов или аномалий является одной из основных проблем интеллектуального анализа данных. Экспоненциальный рост информации, а также распространение устройств Интернета Вещей заставляют переосмыслить подход к аномалиям и варианты использования, которые можно построить, рассматривая эти аномалии.

Увеличение объема обучающей выборки и количества параметров сети имеют существенное значение для качества классификации. Собранные сведения полуструктурированы или неструктурированы вовсе, со слабой взаимосвязью, имеют объем от петабайт до эксабайт и децентрализованный способ хранения с горизонтальной моделью обработки. связи с этим следует говорить о применении технологий для работы с большими данными, которые будут реализованы в подсистеме трансформации данных.

Наиболее распространенной технологией является модель распределенных вычислений, называемой MapReduce. Суть этой модели заключается в выполнении двух функций – Map и Reduce (рисунок 1). Map выполняет обработку данных. Входной набор распределяется

между узлами на выполнение главным узлом кластера. Результатом выполнения первого шага является формат «ключ-значение», которое записывается во временное хранилище. Reduce реализует обработку групп данных по ключу и «склеивку» результатов на главный узел кластера [3]. Результат является решением поставленной задачи.



Рис. 1. Функции Map и Reduce

С помощью этой модели приложение делится на фрагменты для обработки на машинах в кластере вычислительной системы, что позволяет обрабатывать почти неограниченные объемы данных, с возможностью быстрого увеличения объема хранилища путем добавления новых машин в кластер. При выходе из строя одного из таких узлов, управление передается другому компьютеру, что обеспечивает высокую отказоустойчивость системы.

Нейросети обычно обучают с помощью градиентных методов. Что касается обучения на одном компьютере, то наиболее эффективным является использование GPU (Graphics Processing Unit). В тоже время, память GPU, как правило, не превышает 2-4 GB, что накладывает ограничение на размер нейросети. Также проблемой является долгое (дни и даже недели) обучение нейросети на одной машине, что замедляет прогресс в разработке новых нейросетевых архитектур [4]. Наиболее простое решение - использование нескольких GPU, соединенных высокопроизводительной шиной на одном узле. Однако пропускная способность шины не позволяет использовать более 4-8 GPU. Поэтому для дальнейшего ускорения необходимо переходить к распределенным архитектурам.

Разработаны два основных способа по распараллеливанию обучения нейросети: по данным (data parallel) и по модели (model parallel). В первом случае производится вычисление градиентов независимо каждым узлом по определенному диапазону выборки, для получения конечного результата эти градиенты суммируются, синхронно или асинхронно. Объем передаваемых между узлами данных пропорционален количеству параметров модели. Вычисление градиентов на GPU происходит достаточно быстро, поэтому параллельные по данным вычисления непрактичны, если передача данных будет занимать больше времени, чем вычисление градиентов. Это имеет место для больших нейросетей при передаче данных по Ethernet. При распараллеливании по модели необходимо передавать между машинами выходы нейронов для выполнения прямого и обратного прохода. В этом случае обмен данными может быть существенно сокращен, если архитектура сети - не полносвязная, что типично для задач компьютерного зрения.

Операция Map позволяет произвести вычисления по пакетам. Драйвер кластера будет отправлять каждый из этих пакетов на разные вычислительные узлы в кластере (в зависимости от ресурсов и конфигурации кластера). Таким образом, каждый компьютер в кластере сможет вычислить прямое и обратное распространение на назначенном ему фрагменте данных [5].

Переход в необходимое состояние представляет собой преобразование карты. Эти операции выполняются для каждого входного изображения параллельно с разных компьютеров в кластере. Это означает, что если необходимо увеличить обучающий набор, то следует масштабировать ресурс в кластере, чтобы иметь возможность вычислять градиенты для всех обучающих изображений.

Схема работы с графическими данными с помощью MapReduce представлена на рисунке 2. Вычисления выполняются функцией Map. Если обработка задействует несколько узлов, результаты будут переданы функции Reduce для «склейки» в единое целое.

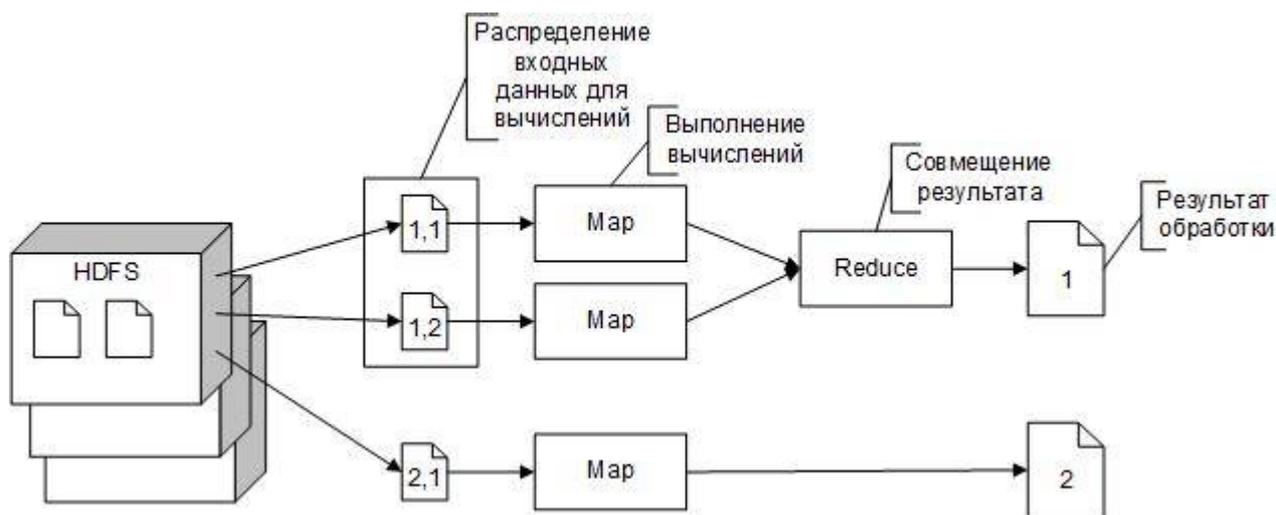


Рис. 2. MapReduce для обучения нейросети

Ускорение методов машинного обучения за счет вычислений на кластере позволяет на этапе исследования пробовать больше вариантов моделей и решать практические задачи, в которых возникают большие обучающие выборки. Все это способствует, в конечном счете, научно-техническому прогрессу.

Библиографический список

1. Davy Cielen Arno D. B. Meysman Mohamed Ali «Introducing Data Science» - Manning Publications Co., 2016 – 322pp.
2. Трофимов И.Е. Распределенные вычислительные системы для машинного обучения //ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ И ВЫЧИСЛИТЕЛЬНЫЕ СИСТЕМЫ 3/ Москва, 2017. - 56-69 с.
3. Баранова С.Н., Вьюгина А.А., Гринченко Н.Н. Использование модели mapreduce в работе с графическими данными //Методы и средства обработки и хранения информации: Межвузовский сборник научных трудов / Под ред. Б.В. Кострова - Рязань: ИП Коняхин А.В. (Book Jet). Рязань, 2020. – 31-35 с.
4. Кондыбаева Алмагуль Бауржановна, Оситис Анастасия Петровна, Калашников Евгений Александрович Использование распределенных вычислительных систем в машинном обучении // T-Comm. 2018. №5. URL: <https://cyberleninka.ru/article/n/ispolzovanie-raspredelennyh-vychislitelnyh-sistem-v-mashinnom-obuchenii>
5. K. R. Jayaram, A. Gandhi, H. Xin and S. Tao, "Adaptively Accelerating Map-Reduce/Spark with GPUs: A Case Study," 2019 IEEE International Conference on Autonomic Computing (ICAC), 2019, pp. 105-114, doi: 10.1109/ICAC.2019.00022.

УДК 004.623; ГРНТИ 20.23.17

ИСПОЛЬЗОВАНИЕ ИНФОГРАФИКИ ВО ВРЕМЯ ПРИЕМНОЙ КАМПАНИИ В ВУЗЕ

Н.Н. Гринченко, А.А. Крошила

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

Аннотация. Система высшего образования России на сегодняшний день построена таким образом, что каждый абитуриент начинает свой путь с дверей приемной комиссии. Для выпускника этот этап, безусловно, является первым шагом к студенческой жизни. А для ВУЗа – результат грамотной организации подготовительного этапа, который начинается задолго до сдачи ЕГЭ. Именно на подготовительном этапе следует рассказать о преимуществах данного ВУЗа перед другими, имеющихся специальностях, помочь выпускнику в выборе профессии. Для более эффективного привлечения абитуриентов следует воспользоваться современными методами передачи информации, одним из которых является инфографика.

Ключевые слова: инфографика, приемная комиссия, приемная кампания, графика, графические инструменты, визуализация данных

THE USAGE OF INFOGRAPHICS DURING THE ADMISSION CAMPAIGN AT THE UNIVERSITY

N.N. Grinchenko, A.A. Kroshilina

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

The summary. The system of higher education in Russia today is built in such a way that every applicant begins his journey from the doors of the admissions committee. For an applicant, this stage is certainly the first step towards student life. And for the University – the result of a competent organization of the preparatory stage, which begins long before the unified state exam (USE). At the preparatory stage it is necessary to tell about the advantages of this university, about specialties. To attract applicants more effectively, you should use modern methods of transmitting information, one of which is infographics.

Keywords: infographics, admissions committee, admissions campaign, graphics, graphical tools, data visualization.

Образование в последние годы всё чаще рассматривается как условие и ресурс развития современной экономики. Работа приемной комиссии связана с предоставлением разной информации: о числе бюджетных мест, стоимости обучения, существующих направлениях подготовки, льготах, дополнительных баллах, наличии общежития и т.д. Несмотря на все многообразие параметров приема, в момент зачисления каждый абитуриент рассматривается отдельно. Основной задачей приемной комиссии как раз-таки и является набор студентов. Однако, чтобы этот набор успешно состоялся, требуются подготовительные этапы, среди них и дни открытых дверей, и встречи с выпускниками в школах, и проведение олимпиад, и т.п. То есть все те мероприятия, которые обеспечивают связь ВУЗа с потенциальными студентами.

Благодаря современным способам передачи данных в настоящее время каналы коммуникации достаточно развиты: приемные комиссии имеют электронные почты и даже группы в социальных сетях. Однако тенденция ведет к тому, что люди все меньше хотят читать и больше смотреть. Визуальная составляющая информации уже не является капризом «для красоты», она выступает полноправным инструментом в борьбе за привлечение внимания. Но что если сделать картинку не только бросающейся в глаза, но и информативной? Тогда абитуриент будет получать ответы на интересующие его вопросы в понятной и легкой форме. Для достижения этого результата можно воспользоваться средствами инфографики.

Инфографика удобна тем, что сфера ее применения обширна: от естественных и гуманитарных наук до маркетинга и статистики, то есть в любой научной или профессиональной сфере, где существует необходимость схематизации материала.

В современных реалиях знаменитое изречение «кто владеет информацией, тот владеет миром» можно дополнить. Уже недостаточно ею просто обладать, необходимо ее анализировать. Для упрощения данного процесса существуют различные технологии визуализации данных, одной из которых выступает инфографика.

Компании должны строить свою деятельность, основываясь на том факте, что активное внедрение инфографики стало причиной появления «экономики внимания». То есть после того, как потребителем была замечена какая-либо графическая информация, он все свое внимание устремляет на тот объект, который ему её предоставил.

Если говорить о рассматриваемой предметной области «ВУЗ», то одной из важных функций, обеспечивающей конкурентоспособность вуза, выступает формирование контингента студентов, а именно приём абитуриентов на основе организованного вузом конкурса. Чтобы принять какие-либо стратегические решения по поводу приемной кампании, необходимо организовать информационные потоки с обратной связью, обработать полученную информацию, проанализировать ее и выработать на ее основе соответствующие рекомендации и проекты, где и незаменимой становится инфографика: от очевидных аналитических отчетов до рекламной кампании. Современный пользователь большое внимание уделяет внешней составляющей, недаром говорится «встречают по одежке». В значительной степени это касается интернет-ресурсов, когда не понятная глазу картинка может стать причиной ухода абитуриента.

Определение степени востребованности направлений подготовки и специальностей, а также мониторинг факторов, которые определяют численность и состав потенциальных абитуриентов – это другая немаловажная задача для вуза. Поскольку на рынке образовательных услуг спрос характеризуется сложной структурой, то инфографика позволяет в наглядной форме получить информацию, которая повлияет на выбор вуза абитуриентом. Например, поскольку существуют конкуренция в сфере образования и определенные ограничения по предоставлению платных образовательных услуг, то это приводит к идентичной стоимости обучения. Поэтому на выбор абитуриента влияет ряд перечисленных ниже факторов [3].

1. Престижность и спрос профессии на рынке труда.
2. Репутация вуза.
3. Наличие общежития, студенческих объединений;
4. Перечень вступительных испытаний и т.д.

Поэтому инфографика должна в значительной мере воздействовать на перечисленные неценовые факторы спроса. Использование актуальных средств визуализации будет говорить о том, что ВУЗ идет в ногу со своими абитуриентами, и этим будет повышаться доверие и формироваться лояльность к учебному заведению.

Также следует отметить, что человек, не погруженный в историю ВУЗа или не определившийся со специальностью, вряд ли будет читать большие объемы текста, а инфографика поможет донести самое главное (рисунок 1).

С другой стороны, абитуриент, уже определившийся, приходит с конкретными вопросами, которые также можно решить с использованием упомянутого средства визуализации, тем самым сохранив полезное время как выпускника, так и сотрудника приемной комиссии (рисунок 1).

В социальных сетях и на сайте инфографика будет служить не только средством привлечения внимания, но и «вирусной рекламой», которой легко поделиться. А значит, число потенциальных абитуриентов может возрасти.

Обмен информацией в интернете, социальных сетях побуждают многие бизнес-проекты соответствовать трендам, и как следствие, сегодня они стремятся максимально визуализировать информацию, упростить ее, чтобы она дошла до пользователя наиболее эффективно. Здесь имеет место сразу два биологических аспекта, во-первых, человеческий мозг устроен таким образом, что основная часть информация из окружающего мира получается с

помощью органов зрения, при этом около 80% из осевшей в памяти информации приходится на графику и только 20% - на текст; во-вторых, выстраивание коммуникации на основе образов и ассоциаций повышает делает информацию эмоциональной и запоминающейся, то есть воздействует напрямую на человека.

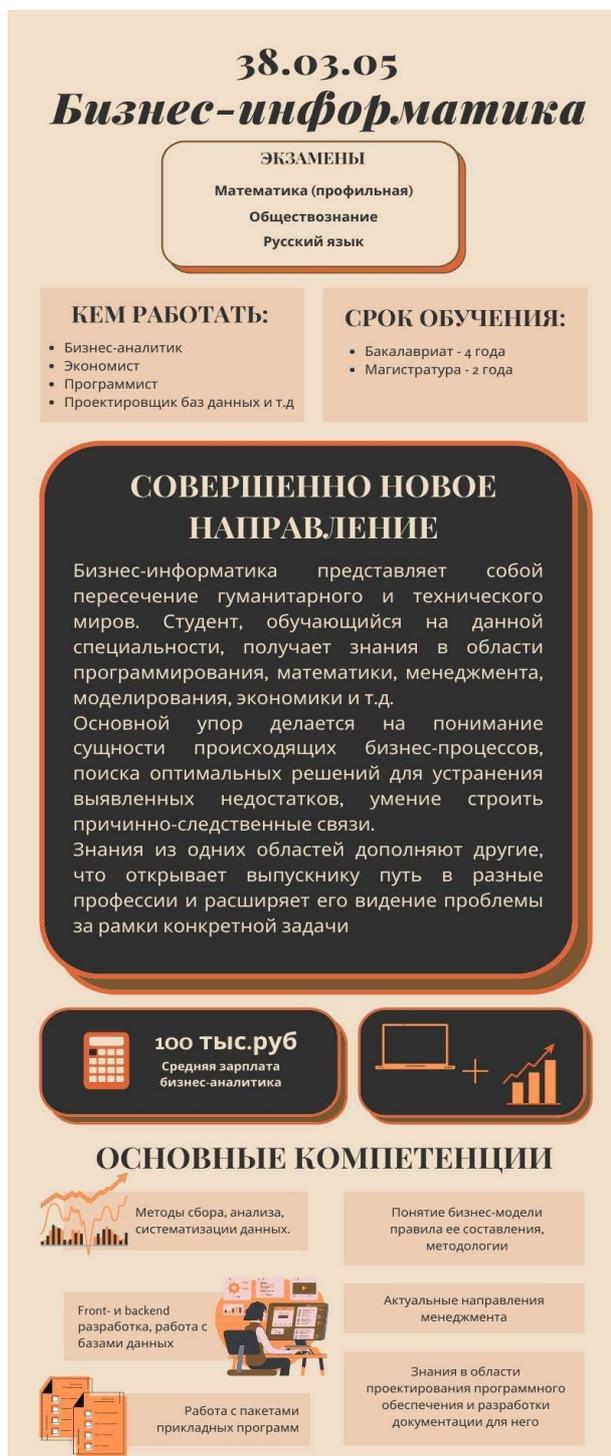


Рис. 1. Пример инфографики во время приемной кампании

Инфографика совершенствует процесс восприятия информации, объясняет сложную информацию с помощью простых образов, передает данные компактным и интересным сообщением, которое выглядит более привлекательно по сравнению с обычным текстом. Вы-

бор формы зависит напрямую от информации, которую необходимо предоставить аудитории. Применяя инфографику, можно визуализировать информацию различной тематики: от серьёзных аналитических данных до обзора настольных игр.

К основным средствам визуализации относят [2]: графики и диаграммы; блок-схемы; матрицы; карты и картограммы.

Инфографику следует выделить отдельно, поскольку её можно отнести к журналистике данных, объясняющей факты по конкретной предметной области, явлению или процессу с помощью графиков и схем. Она, как правило, включает в себя иные способы визуализации (графики, изображения, диаграммы, карты, схемы, таблицы), но при этом имеет текстовые вставки. Инфографику можно охарактеризовать как короткий рассказ с цифрами и фактами, в котором есть завязка и заключение [4].

Среди отличительных особенностей можно выделить следующие [1]:

1. Пертинентность информации. Иллюстрация может воздействовать на эмоции, но не показывать ничего конкретного, а в инфографике всегда есть цифры или факты.
2. Цельность. К инфографике не нужен большой сопроводительный текст, она его как раз заменяет.

На рисунке 2 представлены 5 основных видов инфографики [5]:

Аналитическая, или цифровая инфографика	•представляет статистические данные, в ней много цифр, диаграмм, графиков
Новостная инфографика	•Наглядно рассказывает про свежие события, отражает хронологию и важные этапы происходящего вокруг
Конструкционная инфографика	•Демонстрирует устройство объекта или механизм явления, иногда — хронологию и причины исторического события
Рекламная инфографика	•Создаётся компаниями для продвижения своего продукта. Как правило, это яркая картинка с описанием достоинств товара и выгод от покупки.
Сравнительная инфографика	•Позволяет сравнить характеристики различных объектов

Рис. 2. Основные виды инфографики

Таким образом, инфографика является современным и удобным инструментом визуализации информации, который может помочь эффективно провести приемную кампанию.

Библиографический список

1. Андреев М.Д. Инфографика как способ переработки информации: смысл vs картинки (рекомендации по созданию инфографики) // Актуальные вопросы гуманитарных наук – 2020. Материалы выступлений молодых ученых в рамках Ломоносовских чтений 2020. — Архангельск, 2020. — С. 73-78.
2. Золотухин С.А. Инфографика как информационный пакет. Инфографика в образовании // Медиа. Информация. Коммуникация. — 2018. № 25. С. 1-8.
3. Логачев, А. И. Работа приемной комиссии высшего учебного заведения: проблемы и перспективы / А. И. Логачев. — Текст : непосредственный // Молодой ученый. — 2018. — № 38 (224). — С. 130-131.
4. Остриков С.В. Гибридикация как метод проектирования дизайнерской инфографики // Декоративное искусство и предметно-пространственная среда. Вестник МГХПА. — 2014. № 2. —С. 225-234.
5. Шелудивченко Б.А., Ткаченко О.Н. Виды и типы инфографики // Творчество молодых: дизайн, реклама, информационные технологии. сборник трудов XIV Международной научно-практической конференции студентов и аспирантов. Научный редактор Л. М. Дмитриева. — 2015. С. 125-128.

УДК 004.657; ГРНТИ 20.23.17

МНОГОМЕРНЫЙ АНАЛИЗ ДАННЫХ С ИСПОЛЬЗОВАНИЕМ СРЕДСТВ MS SQL SERVER И MS VISUAL STUDIO

Н.Н. Гринченко, А.А. Крошила

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

Аннотация. Целью работы является исследование возможностей MS SQL Server и MS Visual Studio для многомерного анализа, который может быть полезен в реализации бизнес-целей. В работе рассмотрены основные проблемы, которые могут возникнуть в процессе обработки многомерных данных, описаны преимущества в сравнении с реляционными базами данных, описаны средства анализа, предоставляемые MS SQL Server и MS Visual Studio. Результатом работы является обработка результата развернутого на сервере куба данных по предметной области «Разработка программных продуктов».

Ключевые слова: MS SQL Server, MS Visual Studio, многомерный анализ, куб данных, разработка программных продуктов.

MULTIDIMENSIONAL DATA ANALYSIS USING MS SQL SERVER AND MS VISUAL STUDIO TOOLS

N.N. Grinchenko, A.A. Kroshilina

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

The summary. The aim of the work is to explore the capabilities of MS SQL Server and MS Visual Studio for multidimensional analysis, which can be useful in the implementation of business goals. The study discusses the main problems that may arise during the processing multidimensional data, describes the advantages in comparison with relational databases, and shows the analysis tools provided by MS SQL Server and MS Visual Studio. The result of the work is the result's processing of the data cube deployed on the server in the "Software development" domain.

Keywords: MS SQL Server, MS Visual Studio, multidimensional analysis, data cube, software development.

Одной из важнейших областей применения вычислительной техники (ВТ) остается переработка и хранение больших объемов информации.

Существующие на данном этапе информационные системы (ИС) отличаются существенными объемами хранимых и обрабатываемых данных, сложной организацией, необходимостью удовлетворять разнообразные требования многочисленных пользователей.

В основе любой системы, реализующей автоматизированный сбор, обработку данных, а также манипулирование ими, лежит база данных (БД) [1].

ИС крупных предприятий чаще всего включают в себя приложения, которые предназначены для комплексного анализа данных, их динамики, тенденций и т.п. Конечной целью данного анализа так или иначе является содействие в принятии решений, а значит главным потребителем его результата выступает высший аппарат менеджмента.

Для принятия управленческого решения любого рода нужно обладать релевантной информацией, чаще всего количественной. Для удобной работы данные сначала необходимо собрать из всех ИС предприятия, затем привести к общему формату и только после этого анализировать. С этой целью создаются хранилища данных (Data Warehouses) [2].

Системы поддержки принятия решений (СППР), как правило, включают в себя средства предоставления пользователю агрегатных данных для различных выборок из исходного набора в виде, наиболее удобном для восприятия и анализа (таблицы, диаграммы и т.п.) [2].

Реляционные БД (РБД) были созданы для осуществления большого количества параллельных транзакций, с возможностью сохранения производительности и целостности данных. Однако из-за особенностей своей конструкции РБД не являются эффективными в случае агрегирования и поиска при больших объемах данных. Чтобы выполнить поставленную задачу РБД должна получить запрос, основанный на наборе. Подобные запросы — очень за-

тратные, в силу того, что они основаны на множественных соединениях и агрегатных функциях; еще больше эффективность они теряют при работе с большими массивами данных [1].

Классический подход сегментирования исходных данных заключается в следующем: из исходных данных выделяются один или несколько многомерных наборов данных (они носят название гиперкуб или метакуб), оси которых содержат атрибуты, а ячейки – агрегируемые количественные данные. При этом атрибуты вдоль каждой оси могут быть организованы иерархическим образом, что позволяет представить различные уровни их детализации. При помощи подобной модели данных пользователи могут составлять сложные запросы, создавать отчеты, получать подмножества данных.

Центральным компонентом организации традиционных хранилищ данных выступает описанная в 1993 году Эдгаром Коддом технология комплексного многомерного анализа под названием OLAP (On-Line Analytical Processing) [3].

Для устранения описанного выше недостатка в РБД применяются многомерные сущности (кубы). Используя куб, можно предоставить пользователям структуру данных, обеспечивающую быстрый отклик на запросы с большими объемами агрегации (объединения). Кубы способны выполнять это за счёт предварительного агрегирования данных по нескольким измерениям, которое, как правило, обычно осуществляется во время его обработки.

В аналитической службе MS SQL Server (SSAS) куб выступает в качестве центральной конструкции данных в оперативной системе OLAP. С логической точки зрения куб выступает своеобразным хранилищем данных, который состоит из размерностей (dimensions) и измерений (measures) [4].

Размерности включают в себя описательные признаки и иерархии, а измерения представляют собой факты, описанные в размерностях. Измерения объединяются в логические сочетания, представляющие собой группы измерений. Привязка размерности к группам измерений осуществляется на основе признака, который носит название «степень детализации».

Для файловой системы куб представляет собой последовательность связанных бинарных файлов, именно подобная архитектура обеспечивает быстрое извлечение больших объемов многомерных данных. Кубы строятся из основной РБД, называемой моделью размерностей, которая содержит реляционные таблицы (измерения и размерности), что связывает ее с сущностями куба [1].

В рамках предметной области «Разработка программных продуктов» могут быть существовать разного рода задачи на анализ продуктивности работы команды, соответствия конечного продукта требованиям заказчика и т.п. Все они решаются посредством многомерного анализа [5].

Для использования средств многомерного анализа в MS SQL Management Studio необходимо установить пакет анализа Microsoft Data Tools Analysis Services – расширение для Visual Studio, позволяющее создавать проекты для многомерного анализа. А также SQL Server Data Tools (SSDT) – службу для запуска SQL Server в режиме для подключения к пакетам анализа.

Допустим, необходимо получить количество заказов определенного типа, выполненных определенной командой в заданный период времени. Первым шагом при создании куба является подбор мер и измерений. Из условия задачи видно, что единственным измерением будет выступать количество заказов, а срез будет производиться по трем условиям, которые станут измерением – дата, команда и тип продукта (рисунок 1).

	Тип1	Тип2	Тип3
Дата1	3	15	35
Дата2	10	25	28
Дата3	8	20	18

Рис. 1. Макет реализуемого куба

Перед непосредственным развертыванием куба необходимо выполнить предварительную настройку MS SQL Server [4], чтобы локальный сервер стал доступен для инструментов анализа. В мастере установки MS SQL Server важно не только выбрать Analysis Services, но и изменить режим на многомерный анализ, иначе сервер не сможет развернуть куб. Проверить доступность сервера для анализа можно в MS SQL Server Configuration Manager, где в службах должен отображаться сервер с заданным именем и типом службы «Analysis Services». Для последующей связи службы анализа с сервером и получения доступа к базе данных необходимо настроить параметры входа и указать текущего пользователя на компьютере и придумать пароль.

Для непосредственного анализа и развертывания куба необходимо воспользоваться расширением Microsoft Analysis Services Projects для MS Visual Studio [4]. Особенно важно в настройке проекта указать тип аутентификации по пользователю и паролю, которые были указаны при настройке MS SQL Server, а в настройках проекта в пункте развертывания поменять сервер на тот, который был создан для службы анализа. Если перечисленные действия не будут выполнены, то при обработке куба возникнут ошибки доступа и он не будет развернут.

В литературе чаще всего встречается вариант создания измерений куба на основе специально создаваемых отношений для каждого измерения. Этот способ является и наиболее легким, так как создание кубов в MS Visual Studio не предполагает использования напрямую отношений с простым ключом [1].

Однако часто приходится решать задачи в рамках существующих БД, где создание новых специальных отношений невозможно или неоправданно. Для того чтобы создать измерение на основе таблицы с составным ключом, нужно создать новое вычисляемое поле, которое станет простым ключом для измерения, он же добавится в исходную БД [2].

Создание куба начинается с создания представления, в которое включаются таблицы, содержащие измерения и меры. В рамках решаемой задачи это таблицы Договор, которая содержит дату, ТипПродукта, в которой содержится код типа продукта, таблица Команда, содержащая код команды, а в качестве таблицы мер выступает ПрограммныйПродукт. На основе существующего представления создаются указанные меры и измерения. Схема куба представлена на рисунке 2, на нем видно, что измерения выделяются синим цветом, а меры – желтым.

Если сервер был правильно настроен, то при обработке куба (рис. 3) не возникнут ошибки. По завершении процесса с многомерной сущностью можно работать. Основные средства включают в себя создание срезов по заданным условиям.

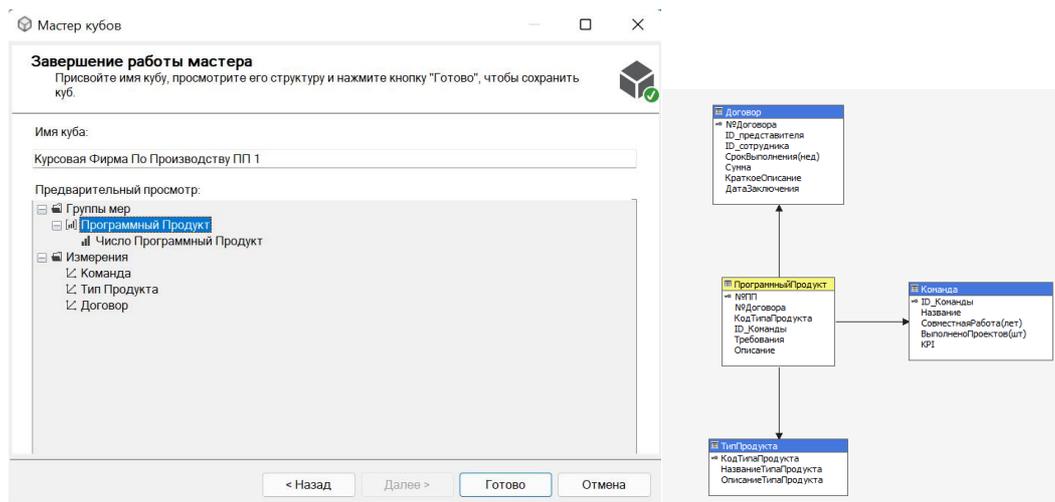


Рис. 2. Представление куба

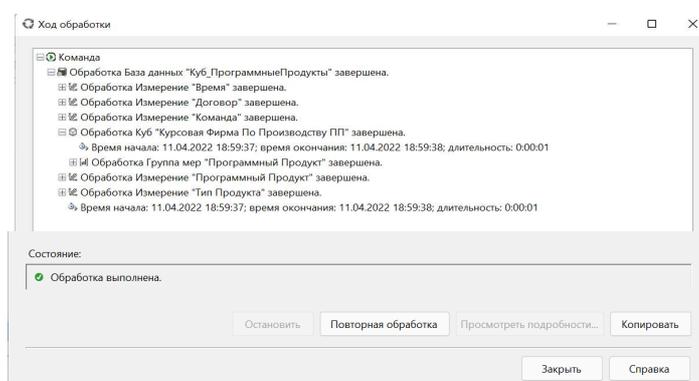


Рис. 3. Обработка куба

Итог решаемой задачи представлен на рисунке 4. На основе созданного куба можно решать и другие задачи, так как измерения строятся не по отдельному атрибуту таблицы, а по нескольким, можно выбирать разные наборы данных. Также существует возможность настройки параметров атрибутов, что еще больше расширяет возможности многомерного анализа.

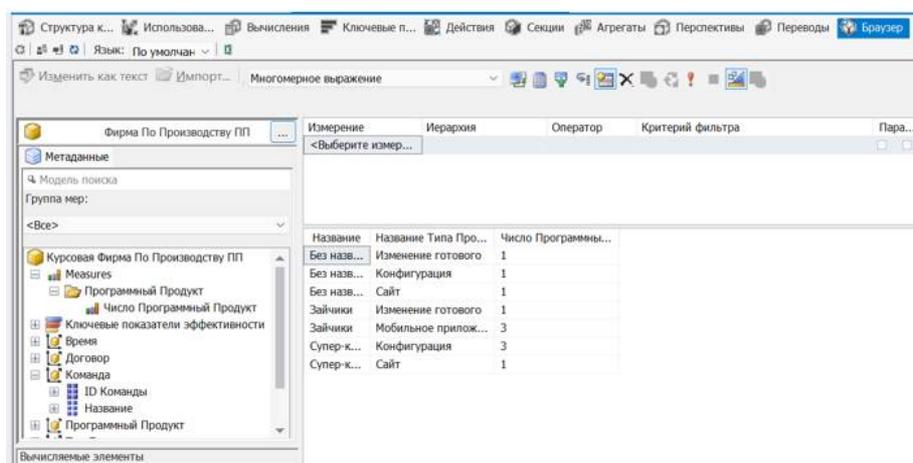


Рис. 4. Развернутый куб

Таким образом, MS SQL Server и MS Visual Studio обладают обширным набором средств для многомерного анализа, который широко используется при решении конкретных бизнес-задач. Основным затруднением при работе с SSAS является настройка сервера, но описанные нюансы способствуют разрешению возникающих ошибок.

Библиографический список

1. Бурывлов В.С., Назаров Р.О. // Анализ и синтез сложных бизнес-систем с помощью методики многомерного анализа данных / Вестник ИНЖЭКОНа. Серия: Экономика. – 2014. – № 2 (69). – С. 156-158.
2. Благодатский Г.А., Докучаев Д.Е., Саетова Л.Г. // Многомерный анализ баз данных: способы и методы / Социально-экономическое управление: теория и практика. – 2020. – № 1 (40). – С. 76-79.
3. Минаева Ю.В., Асунин Г.И. // OLAP-технологии комплексного анализа многомерных данных / В сборнике: Интеллектуальные информационные системы. Труды Международной научно-практической конференции. В 2-х частях. – 2019. – С. 47-48.
4. Шаркова М.А., Сарапулова Т.В. // Возможности MS SQL Server для интеллектуального анализа данных / В сборнике: Материалы Всероссийской молодежной конференции "Информационно-телекоммуникационные системы и технологии (ИТСиТ-2012)". – 2012. – С. 182-183.
5. Тимофеев Д.А. // Программные продукты: от разработки к производству / Прикладная информатика. – 2010. – № 3 (27). – С. 54-61.

УДК 004.023; ГРНТИ 28.17

ПОСТРОЕНИЕ ПРОСТЕЙШЕЙ СИМУЛЯЦИИ ЖИЗНЕДЕЯТЕЛЬНОСТИ ОРГАНИЗМОВ

Д.Ю. Логинов, Н.И. Хизриева

*Рязанский государственный радиотехнический университет имени В.Ф. Уткина,
Российская Федерация, Рязань, dima2013loginov@mail.ru*

Аннотация. В работе рассматриваются возможные принципы построения простейшей симуляции жизнедеятельности организмов на основе генетического алгоритма, описываются основные элементы системы и методы их взаимодействия.

Ключевые слова: генетический алгоритм, естественный отбор.

BUILDING A SIMPLE VITAL ACTIVITY SIMULATION

D.Y. Loginov, N.I. Khizrieva

*Ryazan State Radio Engineering University named after V.F. Utkin,
Russia, Ryazan, dima2013loginov@mail.ru*

The summary. The paper discusses the most essential principles of building a simple simulation of vital activity of organisms based on a genetic algorithm, it describes main elements of the system, and mediums of their interaction.

Keywords: genetic algorithm, natural selection.

Генетический алгоритм — это один из методов поиска и оптимизации, прообразом которого послужил биологический принцип естественного отбора [1]. Следовательно, помимо множества прикладных задач данный алгоритм может эффективно использоваться для симуляции жизни. В таких условиях на реалистичность влияет не только непосредственно реализация генетического алгоритма, но также ряд решений по формированию самой системы и взаимодействий в ней. Рассмотрим принципы, которые были использованы автором при практическом построении такой симуляции.

Основные элементы симуляции и способ ее формирования

Помимо среды обитания в системе присутствуют следующие элементы: пища, которая изначально находится в среде обитания и ей порождается (в дальнейшем для простоты

“растения” или “продуценты”), особи, поглощающие эту пищу, являющиеся аналогами консументов первого порядка (в дальнейшем “травоядные”) и особи, питающиеся травоядными, аналогичные консументам второго порядка.

В начале симуляции в среду с некоторым количеством растений помещаются травоядные особи. В таком случае популяция продуцентов начинает убывать, и вследствие вымирает вся система, поэтому необходимо предусмотреть механизм их восстановления. Так, для простой симуляции подойдет периодическое добавление нескольких особей продуцентов, однако если стоит задача создания более стабильной системы с уменьшенной частотой вымирания (например, для демонстрации конкретных генетических процессов) можно также добавить границу, ниже которой число растений опуститься не может. Спустя некоторое время система придет к балансу - среднее количество погибающих и рождающихся особей равно. Далее в такую систему могут быть добавлены хищники.

Характеристики существ

Неотъемлемой частью как генетических алгоритмов, так и жизнедеятельности организмов является процесс эволюции в условиях отбора. Для его моделирования хищным и травоядным особям был добавлен набор характеристик, таких как голод, скорость передвижения, количество очков здоровья, эффективность в бою и степень защиты. Эти параметры напрямую влияют на состояние системы и баланс в ней, а потому требуют их точной настройки у начальной популяции особей. В соответствии с определением генетического алгоритма[1] в процессе жизнедеятельности характеристики существ из поколения в поколение должны изменяться посредством мутаций, скрещивания и отбора.

Механизм размножения и мутаций

Для нормального функционирования системы особи в ней должны размножаться. При реализации системы был использован аналог механизма полового размножения, использующийся многими видами живых существ. Так в процессе репродукции из генов двух особей формируется набор характеристик из потомка, который с некоторым шансом (а для ускорения эволюционных процессов с вероятностью 100%) может мутировать. Мутация в рассматриваемой симуляции полагается как приращение (как положительное, так и отрицательное) случайных характеристик на случайную величину, которое может быть нулевым, притом сумма этих приращений не может быть больше фиксированного значения суммарной силы мутаций. Стоит также отметить, что в рассматриваемой системе также были сформированы ограничения для особей по частоте размножения и правила, не позволяющие размножаться недостаточно “сытым” особям.

Способы перемещения

Одним из самых важных решений, влияющих на качество симуляции, является реализация механизма перемещения. Слишком примитивная модель может негативно сказаться на реалистичности, а слишком сложная может помешать наблюдению за эволюционными процессами и ограничить систему в масштабе за счет более ресурсоемких вычислений. Так в процессе создания симуляции были опробованы следующие алгоритмы.

1. Алгоритм, основанный на определении ближайшего объекта.

Метод полагается на поиск двух векторов: к ближайшей пище и к ближайшему потенциальному партнеру. После их нахождения особь двигалась в направлении меньшего из них, если она способна к размножению (учитывая критерии, перечисленные в предыдущем пункте), или в сторону пищи, в случае неспособности. Несмотря на всю простоту этого алго-

ритма, стоит отметить, что он все же позволяет построить визуально реалистичную модель поведения организмов: они объединяются в группы, конкурируют за пищу и разделяются.

Сильными сторонами данного алгоритма являются:

- 1) Простота реализации
- 2) Малое количество затрачиваемых ресурсов позволяет увеличить размеры симуляции
- 3) Относительная визуальная реалистичность симуляции

Недостатками алгоритма являются:

- 1) Низкая правдоподобность поведения

2. Алгоритм, основанный на характеристиках существ

Для реализации метода существам добавляются характеристики, отражающие индивидуальные приоритеты особи, а именно коэффициент склонности к пище (α_1), размножения (α_2) и избегания опасности (α_3), которые также подвержены мутациям. Сначала находятся три вектора: к ближайшей пище (A_1), к ближайшему партнеру (A_2) и от ближайшего врага (A_3). Затем вычисляется вектор

$$\gamma = \min(A_i \cdot (\alpha_i + \kappa_i)), i = \overline{1,3},$$

где κ_1, κ_2 - скорость передвижения особи, κ_3 - сумма характеристик атаки и защищенности. Эти коэффициенты позволяют особям пользоваться своими сильными сторонами для выживания. В соответствии с вектором γ и происходит перемещение.

Преимуществом этого алгоритма является его реалистичность: присутствуют элементы группового взаимодействия, как и в случае первого метода, однако особи также способны убежать от опасности и имеют свои индивидуальные предпочтения, что позволяет разнообразить поведение существ и качество симуляции, однако данный способ гораздо более требователен к вычислительным ресурсам.

В процессе разработки системы также рассматривалась схема, основанная на наличии у каждой особи “зрения”, которое охватывало бы лишь небольшую область вокруг себя, однако в таком случае особь все также ищет ближайшие объекты к себе, а значит, при большой плотности существ в симуляции этот алгоритм ничем не отличался бы от предыдущего при существенно большей сложности и затрачиваемым ресурсам.

Пример работы системы

На скриншоте ниже (рис. 1) представлена система, построенная по правилам, описанным в статье с использованием второго алгоритма перемещения. Симуляция создана с помощью среды разработки Unity на языке C#. Также в проекте использовались средства фреймворка Photon для написания сетевого кода, который позволяет нескольким пользователям наблюдать за системой и вносить в нее изменения, например, замедлять особей в области или уничтожать их в ней.

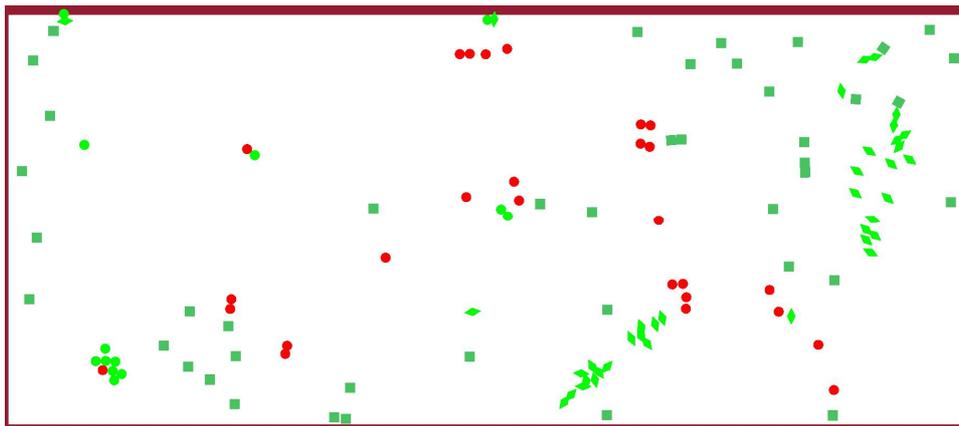


Рис. 1. Симуляция системы живых организмов

В данной системе растения обозначены квадратами темно зеленого цвета, травоядные - кругами, а хищники - ромбами, причем красный цвет означает доминирование характеристики здоровья, зеленый - скорости передвижения, синий - защищенности, черный - эффективности в бою, однако последние два цвета являются довольно редкими ввиду высоких начальных значений, заданных первым двум характеристикам и на рисунке не представлены.

1. Генетический алгоритм [Электронный ресурс]: Википедия. Свободная энциклопедия. – URL: https://ru.wikipedia.org/wiki/Генетический_алгоритм (дата обращения: 14.05.2022).

СЕКЦИЯ «ИНФОРМАЦИОННЫЕ СИСТЕМЫ И ТЕХНОЛОГИИ»

УДК 004.853; ГРНТИ 50.51.19

ОБУЧЕНИЕ ОНТОЛОГИИ ЗНАНИЯМ О СТАНДАРТЕ IEC 61499**Д.С. Журкин*, А.В. Дубинин*, В.Н. Дубинин*, Л.П. Климкина****** Пензенский государственный университет,
Россия, Пенза, dubinin.victor@gmail.com**** Пензенский государственный аграрный университет,
Россия, Пенза, ludmila.klimkina@gmail.com*

Аннотация. Рассматривается методика обучения онтологий для извлечения онтологии стандарта IEC 61499 из его текстового описания. Эта методика основана на подходе к обработке естественного языка с использованием средства NLTK. Также представлена онтология стандарта EC 61499, построенная по этой методике.

Ключевые слова: стандарт IEC 61499, онтология, обучение онтологий, обработка естественного языка, NLTK.

**ONTOLOGY LEARNING FOR THE IEC 61499 STANDARD
BODY OF KNOWLEDGE****D.S. Zhurkin*, A.V. Dubinin*, V.N. Dubinin*, L.P. Klimkina****** Penza State University,
Russia, Penza, dubinin.victor@gmail.com**** Penza State Agricultural University,
Russia, Penza, ludmila.klimkina@gmail.com*

Abstract. An ontology learning technique to extract an ontology of the IEC 61499 standard from its textual description is considered. This technique is based on a natural language processing approach using NLTK. The EC 61499 standard ontology, built using this technique, is also presented.

Keywords: IEC 61499 standard, ontology, ontology learning, natural language processing, NLTK.

Стандарт IEC 61499, принятый в 2005 году, определяет путь для построения систем управления промышленными процессами нового поколения [1]. Это интеллектуальные распределенные компонентно-базированные системы. Архитектура IEC 61499 строится на основе определений IEC 61131-3. Основной элемент архитектуры – это функциональный блок (ФБ), заимствованный из стандарта IEC 61131-3, но расширенный в нескольких направлениях для того, чтобы отразить новые достижения в области проектирования современных программных систем. Одно из основных расширений ФБ – событийный интерфейс, который позволяет явно определить последовательности выполнения ФБ. Другое расширение ФБ связано с объектной и компонентной ориентацией. В ФБ могут быть инкапсулированы алгоритмы. В общем и целом, ФБ IEC 61499 представляет собой независимую программную единицу, которая может быть реализована, протестирована и использована отдельно от других ФБ. Кроме ФБ, элементами архитектуры IEC 61499 являются следующие артефакты: система, устройство, ресурс, приложение, субприложение, адаптер. Следует отметить, что стандарт IEC 61499 является далеко не тривиальным документом, состоит из нескольких частей, и содержит большое количество слабоструктурированной информации.

При автоматизированном проектировании систем управления возникает проблема использования стандарта IEC 61499, обусловленная тем, что стандарт представлен в форме, ориентированной на человека (текстовые, табличные и графические данные), а не на машинную обработку. Актуальной становится задача представления стандарта в формализованном, структурированном и машинно-ориентированном виде (например, в виде базы данных или базы знаний). Такое структурированное представление информации также будет способствовать: а) облегчению и ускорению обучению стандарту IEC 61499 различных категорий

пользователей - от студентов до инженеров и научных работников; б) избавлению от двусмысленности и проблем толкования определений и утверждений стандарта IEC 61499.

В настоящее время большое значение приобрело онтологическое представление знаний [2]. Под *онтологией* понимается формальное представление множества концептов внутри домена и отношений между этими концептами. Использование Web-онтологий в проектировании имеет множество преимуществ, хорошо описанных в литературе. К настоящему времени известны несколько онтологий стандарта IEC 61499, построенных вручную, например [3,4]. К особенностям и недостаткам ручного построения онтологий относятся: 1) онтологии не являются всеобъемлющими и ориентированы на конкретную задачу; 2) не лишены ошибок; 3) большая трудоемкость разработки. Из этого вытекает необходимость автоматизировать процесс построения онтологий. Среди известных методов выделяется метод обучения онтологий. Под обучением онтологии будем понимать процесс извлечения и определения концептуальных знаний предметной области из текстовой информации, который использует обработку естественного языка и методы машинного обучения для устранения узких мест в получении знаний в ручном режиме. Методы обучения онтологий обычно разделяются на три основные группы: 1) методы, основанные на подходах из области искусственного интеллекта; 2) статистические методы; 3) методы, использующие лингвистические подходы. Обзор методов обучения онтологий можно найти в работе [5]. В нашей работе будем базироваться на работе [6], где онтология обучалась системному проектированию. К настоящему времени известно несколько инструментальных средств обучения онтологий: *Text-To-Onto*, *Text2Onto*, *OntoLearn*, *TextOntoEx* и др. Следует отметить, что данные средства не подходят для решения задачи генерации онтологии IEC 61499 из-за их ограниченности и малой мощности.

Предлагаемый подход к построению онтологии стандарта IEC 61499 на основе метода обучения онтологий состоит из трех этапов: 1) предварительная обработка документов; 2) лексический анализ с помощью средств обработки естественного языка; 3) извлечение компонентов онтологии. Исходными данными для первого этапа является Часть 1 стандарта IEC 61499 на английском языке в формате PDF [1]. При преобразовании данного файла в текстовый убираются все «шумы», такие как: пунктуация, рисунки, таблицы, номера страниц, форматирование текста, так как они могут помешать лексическому анализу. Для проведения второго этапа использовалась хорошо зарекомендовавшая себя платформа *Natural Language Tool kit* (NLTK) [7] и язык программирования *Python*. NLTK используется для проведения символического и статистического анализа, включая токенизацию, морфологический поиск, тегирование, лексический и синтаксический анализ. На рисунке 1 представлены наиболее часто встречающиеся в стандарте IEC 61499 существительные и именные группы, определенные на этапе 2. Кроме того, в процессе анализа выделяются глаголы и глагольные словосочетания.

На третьем этапе сегментируются и помечаются объекты, которые являются существительными, именами собственными и именными группами. В то время как существительные и именные словосочетания являются понятиями (концептами), то глаголы и глагольные фразы описывают, какие отношения существуют между понятиями. Для извлечения ключевых терминов в домене стандарта IEC 61499 используется частота распределения различных видов терминов. В зависимости от принятого порога частоты встречаемости слов и выражений, объем онтологии будет различным. В нашем случае порог был принят равным 10. В дальнейшем концепты связываются соответствующими отношениями. В онтологиях есть два вида отношений: таксономические и нетаксономические. Таксономические отношения используются для построения иерархии концептов. Нетаксономические отношения описывают, как понятия связаны друг с другом. Данный вид отношений определяется вручную на основе анализ глаголов и глагольных фраз.

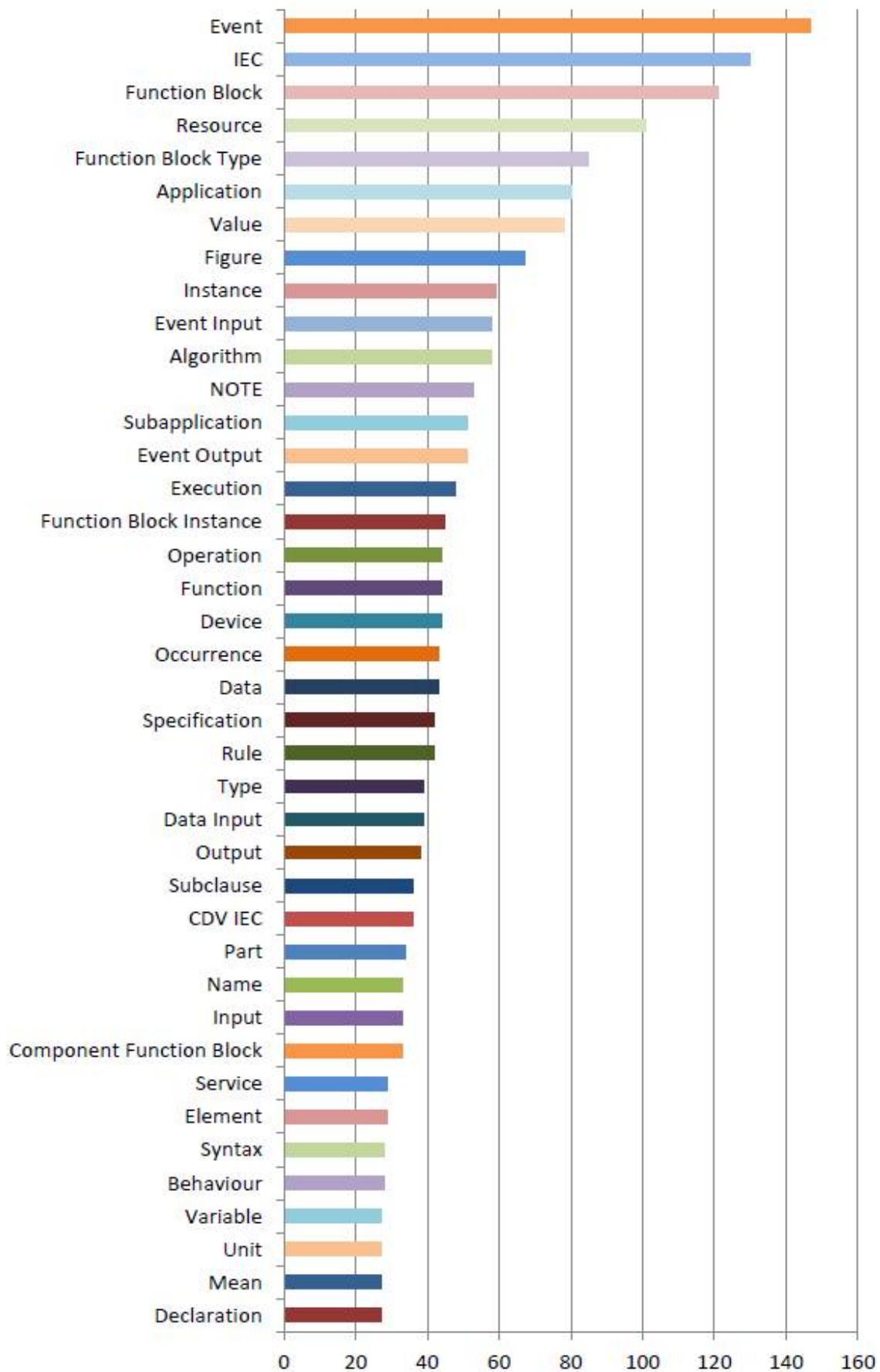


Рис. 1. Наиболее часто встречающиеся существительные и именные группы

Сгенерированная онтология содержит 85 классов, некоторые из которых содержат три уровня вложенности (см. рис. 2 и 3). Среди наиболее важных объектных свойств онтологии можно отметить следующие: *according_to*, *is_defined*, *is_associated*, *is_connected*, *is_used*, *occurs*, *provides*, *represents* и др. При формировании онтологии использовалась система *Protégé* [8], а для ее визуализации – онлайн-средство *WebOWL* [9].

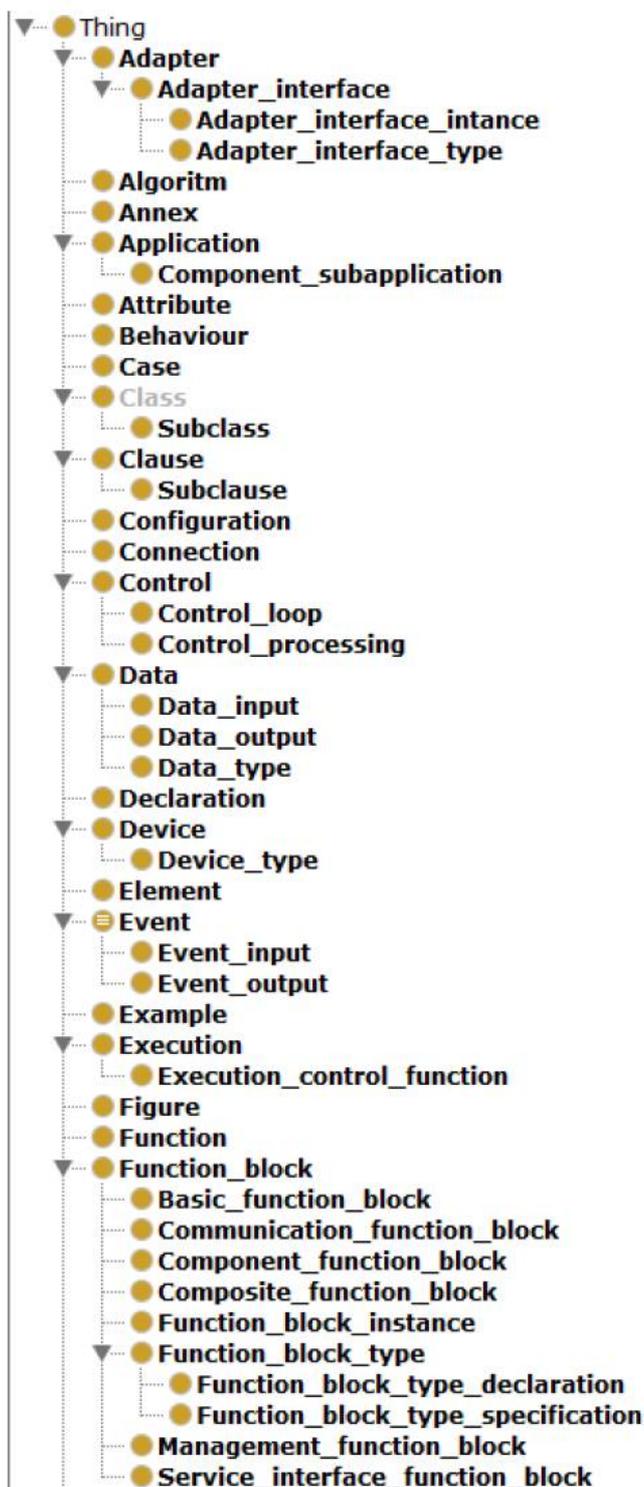


Рис. 2. Классы сгенерированной онтологии стандарта IEC 61499 (фрагмент)

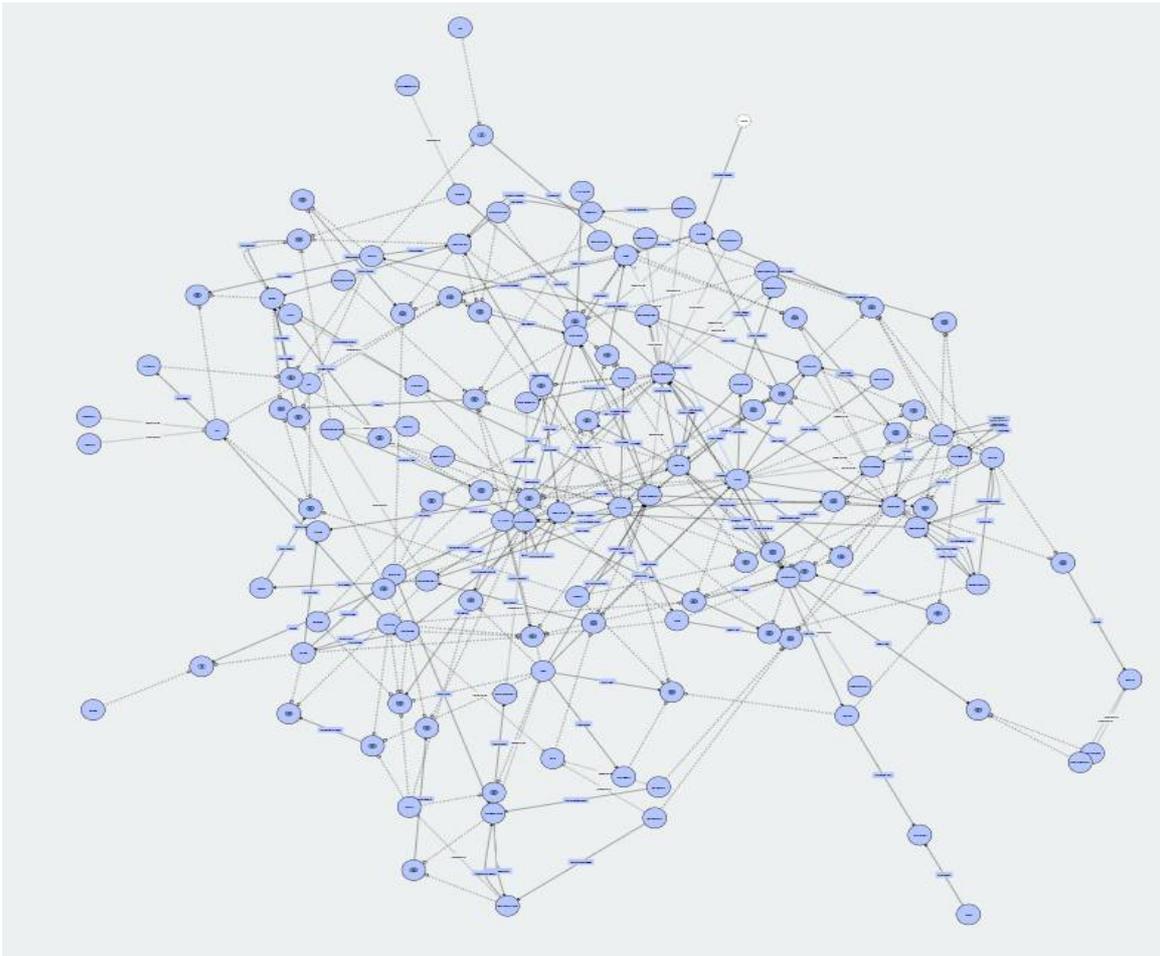


Рис. 3. Визуальное представление сгенерированной онтологии стандарта IEC 61499 в системе WebOWL

Направлением дальнейших исследований является разработка методов автоматического определения нетаксономических отношений и экземпляров классов, а также совершенствование средств поддержки генерации онтологий.

Библиографический список

1. International Standard IEC 61499. Function blocks for industrial-process measurement and control systems (edition 2.0). Part 2: Software tool requirements: ed2.0 / International Electrotechnical Commission. – Geneva, 2012. – 97 p.
2. Гаврилова, Т. А. Базы знаний интеллектуальных систем / Т. А. Гаврилова, В.Ф. Хорошевский. – СПб. : Питер, 2000. – 384 с.
3. Дубинин В.Н., Вяткин В.В. Онтология функциональных блоков стандарта IEC 61499 // Труды Международной научно-технической конференции “Современные информационные технологии”, Пенза, 2010, вып.12. – С. 113-126.
4. Dai W., Dubinin V., Vyatkin V. Automatically Generated Layered Ontological Models for Semantic Analysis of Component-Based Control Systems // IEEE Transactions on Industrial Informatics. . – 2013. Vol. 9, Issue 4. – P. 2124-2136.
5. Asim M. N., Wasim M., Khan M. U. G., Mahmood W., Abbasi H.M. A survey of ontology learning techniques and applications // Database. – 2018. - Vol. 2018. - P. 1–24.
6. Yang L., Cormican K., Yu M. Ontology Learning for Systems Engineering Body of Knowledge // IEEE Transactions on Industrial Informatics. . – 2021. Vol. 17, Issue 2. – P. 1039-1047.
7. NLTK : Natural Language Toolkit [Электронный ресурс]. Режим доступа: <https://www.nltk.org/>.
8. Protégé [Электронный ресурс]. Режим доступа: <http://protege.stanford.edu/>.
9. WebOWL. [Электронный ресурс]. Режим доступа: <http://vowl.visualdataweb.org/webvowl.html>.

УДК 658.7; ГРНТИ 81.88.75

СОВРЕМЕННЫЕ ТЕХНОЛОГИИ В ТРАНСПОРТНОЙ ЛОГИСТИКЕ

Я.Ю. Кремез, М.А. Баталова

*Белорусский национальный технический университет,
Республика Беларусь, Минск, ukremez@gmail.com*

Аннотация. В статье рассматриваются новейшие передовые цифровые технологии, эффективность их использования в управлении цепями поставок. На данный момент существует множество факторов, которые могут препятствовать процессу перевозки, при этом нужно уметь быстро реагировать на любые изменения и трудности, которые возникают в логистической цепочке. Благодаря сложным современным интернет-технологиям участники цепи поставок могут обеспечить достаточную прозрачность в логистических операциях, позволяя повысить производительность процесса доставки груза.

Ключевые слова: цифровизация, транспортно-логистические системы, большие данные, интернет вещей, информационные технологии.

MODERN TECHNOLOGIES IN TRANSPORT LOGISTICS

Y.Y. Kremez, M.A. Batalova

*Belarusian National Technical University,
Republic of Belarus, Minsk, ukremez@gmail.com*

Abstract. The article discusses the latest advanced digital technologies, the effectiveness of their use in supply chain management. At the moment, there are many factors that can hinder the transportation process, while you need to be able to quickly respond to any changes and difficulties that arise in the supply chain. Thanks to sophisticated modern Internet technologies, participants in the supply chain can provide sufficient transparency in logistics operations, allowing to increase the productivity of the cargo delivery process.

Keywords: digitalization, transport and logistics systems, big data, internet of things, information technology.

В настоящее время, когда мир активно прогрессирует и развитие технологий не стоит на месте, в области логистики также внедряются инновации и тенденции для экономического развития, которые основываются на новом технологическом строении. Происходит плавный переход от традиционного подхода управления цепями поставок к виртуальному, что позволяет увеличить эффективность деятельности любого бизнеса, сократить издержки, повысить точность доставки и сохранить время. Сейчас очень важно достигнуть максимальной эффективности управления цепями поставок не за счёт количества задействованных ресурсов, а с помощью грамотной организации процесса перевозок путём внедрения инновационных технологий.

Цифровые технологии играют значительную роль в абсолютно всех секторах экономики, логистика и управление цепочкой поставок также не могут оставаться в стороне. А так как в отрасли логистики активно применяются большие объемы данных, которые хранятся отдельно и в различных программах, и весьма зачастую вводятся вручную, то, возможно, именно логистика больше выиграет от внедрения новейших способов работы [1].

Исследование сведений по тенденциям развития логистики и управления цепями поставок на период до 2030 года крупнейших консалтинговых, аналитических и IT-компаний обнаружил следующие мировые тенденции, доказывающие значимость и актуальность исследования развития цифровизации логистики.

Big Data (Большие данные) совместно с автоматизированными технологиями: Blockchain, ML, IoT, AI, AR/VR будут использоваться для повышения эффективности логистики [2]. Big Data дает возможность «собрать» достаточно информации, которую можно описать так: данные, получаемые в ходе обработки транспортных заявок; данные, которые возникают в процессе управления расписанием на погрузку/разгрузку; вспомогательные опции, а также статусы, которые хочет фиксировать грузоотправитель. Анализ и учет больших данных нужен и в логистике, так как дает возможность обладать наилучшей управляемостью

цепями поставок, грамотно планировать и в любой момент времени иметь реальную, фактическую картину происходящего положения дел [3]. Структура системы обработки больших данных представлена на рисунке 1.

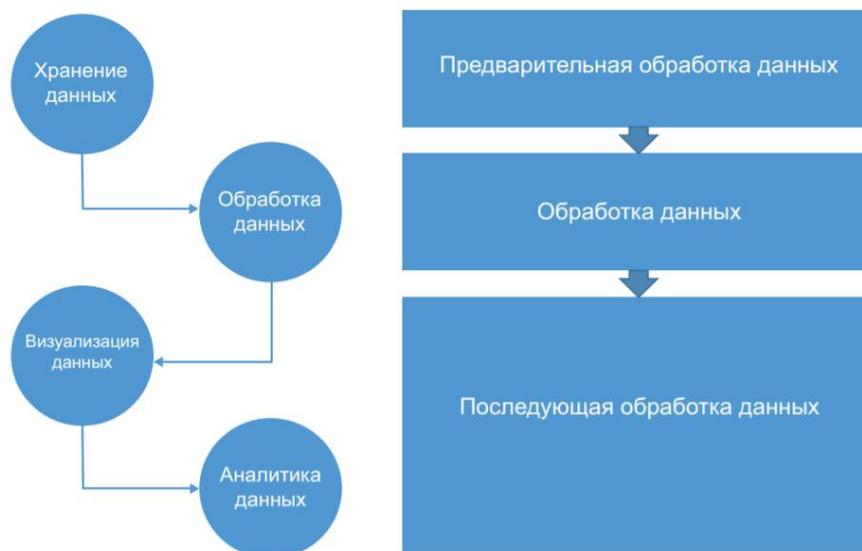


Рис. 1. Структура системы Big Data

Internet of Things, IoT (Интернет вещей) позволит расширить логистический сервис. Это система, которая объединяет различные устройства, взаимодействующие между собой с целью управления, прогнозирования и надзора, помогает руководителям автопарков повышать эффективность контроля трафика, уменьшать затраты с помощью оптимизации ремонта и обслуживания автомобилей [4]. Основные компоненты интернета вещей показаны на рисунке 2.



Рис. 2. Основные компоненты IoT

Last-mile Logistics (доставки на последней миле) – это доставка заказов до конечного покупателя. Цель каждой логистической цепочки состоит в том, чтобы забрать заказ со склада и вовремя доставить его покупателю, в магазин или в конкретный пункт приема. Этот самый последний этап доставки товара и называется “последней милей” [5].

Максимальная автоматизация: идея «умных» складов состоит в том, что всю работу, которую могут выполнить роботы, не должны делать люди. Поэтому применяются на таких

складах не только дроны-доставщики, но и датчики для «умной» инвентаризации, специальные камеры для отслеживания брака и поломок на выдаче, автоматические упаковщики и т.д. Благодаря таким механизмам создается целая экосистема взаимосвязанных устройств, которые собирают информацию о товарах и помогают оптимизировать все логистические издержки [6]. Логистические гиганты, подобные DHL и торговые «киты» вроде Amazon.com и Walmart, давно сделали роботизацию частью стратегии лидерства и активно продвигают свои достижения в этой области, представлены на рисунке 3.



Рис. 3. Робот на складе

Amazon, например, еще в 2012 году приобрела за 775 млн долл. компанию Kiva — производителя промышленных роботов для отбора и упаковки товаров на складе. По данным Deutsche Bank, использование этих роботов позволило Amazon на 20% снизить операционные расходы, что составляет около 22 млн долл. на один распределительный центр. В настоящее время Kiva используются в 13 таких центрах Amazon, но если проект будет масштабирован на все 110 центров компании, то она сможет достигнуть снижения издержек на 800 млн долл. За счет чего достигается такая экономия? В первую очередь за счет повышения эффективности складских процессов: с помощью Kiva цикл складских операций для конкретного товара сократился с 60–75 до 15 минут, а пространство складов оптимизировано на 50% за счет более рационального его использования.

Автономная доставка (автомобили без водителей) уже через пару лет станут очевидными. Технологии в области беспилотных транспортных средств все более нацелены на создание машин с искусственным интеллектом. Искусственный интеллект может полностью контролировать положение и доставлять товары безопасно. Повышаются скорость доставки и оптимальность маршрута. Компьютер сможет анализировать большее количество маршрутов, чем человек. Оценка и применение самых оптимальных маршрутов значительно улучшит трафик и снизит нагрузку на общественные дороги. Это поможет сократить сроки доставки на высоконагруженных участках дорог, а также оптимизировать топливную эффективность [7]. Такого плана автомобиль представлен на рисунке 4.



Рис. 4. Автомобиль с искусственным интеллектом

Новые информационные системы обмена данными и управления процессами в логистике и грузоперевозках призваны обеспечить быструю реакцию на индивидуальные потребности заказчиков, получив при этом экономические преимущества

RFID – это сканер, с помощью которого считывается и распознается идентификационный код единицы хранения склада или артикул. Стационарный считыватель обладает возможностью крепления антенн, которые и отправляют радиосигнал. Зачастую к считывателю подключают от 2 до 8-ми антенн. Обычно такие считыватели крепятся на ворота склада для получения информации со всех меток, которые через них проносятся. Из-за чего стационарные считыватели называют порталными. Общая схема работы на RFID складе может выглядеть, как на рисунке 5.



Рис. 5. Схема работы на складе с RFID

RFID технологии на складе удачно сочетаются с использованием WMS систем и адресным хранением. К тому же RFID сканеры имеют все шансы предоставить возможность создать на складе так называемые "умные полки", когда можно получить в онлайн режиме данные о том, что лежит в конкретной ячейке, благодаря считыванию меток на определенном стеллаже [8].

Цифровой технологический тренд в логистике, связанный с развитием робототехники, автоматизацией логических бизнеспроцессов, использованием систем искусственного интеллекта, направлен на уменьшение количества операций, выполняемых человеком, тем самым снижая влияние человеческого фактора. Технология дополненной реальности позволяет увеличить эффективность таких операций за счет снижения количества ошибок и скорости принятия решений.

Перспективными направлениями применения дополненной реальности в цепях поставок являются:

1. планирование складских помещений. С помощью технологии виртуальной/дополненной реальности склад может быть визуализирован в полном масштабе. Можно моделировать логистические процессы на складе, проводить перепроектирование зон — все это виртуально.
2. погрузка коробок/паллет/контейнеров. Грузчики видят на дисплее устройства дополненной реальности список товаров к погрузке и пошаговые инструкции, как наиболее эффективно загрузить контейнер с учетом размера, формы и веса, входящих в него товаров.

На современных, высококонкурентных рынках эффективное складирование имеет решающее значение, поскольку на него приходится значительная часть логистических затрат. Компании стараются использовать высокоадаптивный и гибкий дизайн складских помещений, который может способствовать интеграции новых технологий, таких как дополненная реальность.

То, что происходит на данный момент с цифровой трансформацией отдельных процессов предприятий, организаций и в целом цепей поставок в промышленности, торговле, сфере услуг нельзя назвать иначе, как «цифровой хаос».

Таким образом, можно заявить, что цифровизация считается преобладающим процессом в транспортной логистике из числа всех проявлений научно-технического развития. К тому же, процессы цифровизации сами по себе происходят стремительнее, нежели предшествующие технологические революции. В результате, окончательный итог данных преобразований предсказать трудно. Однако, возможно отметить два более значимых элемента. На первый взгляд, эффективное использование цифровых технологий в транспортной сфере означает уровень конкурентоспособности предприятия. Те, которые игнорируют современные изменения, рискуют покинуть рынок. А если посмотреть с другой точки зрения, именно процессы цифровизации являются источником повышенных рисков: и в контексте экономического развития, и в контексте общественного прогресса.

Библиографический список

1. ТОП-10 инноваций, которые изменят мир логистики в 2020 году [Электронный ресурс]. – Режим доступа: <https://trademaster.ua/articles/313096>.
2. Цифровые технологии в логистике и управлении цепями поставок: аналитический обзор [Текст] / В. В. Дыбская, В. И. Сергеев, Н. Н. Лычкина и др. ; под общ. и науч. ред. В. И. Сергеева ; Нац. исслед. ун-т «Высшая школа экономики». — М.: Изд. дом Высшей школы экономики, 2020. — 190, [2] с. — 500 экз
3. Big data в логистике – модный тренд, реальность или необходимость? [Электронный ресурс]. – Режим доступа: <https://logistics.ru/automation/news/big-data-v-logistike-modnyu-trend-realnost-ili-neobhodimost>
4. Влияние Интернета вещей (IoT) на управление цепочками поставок [Электронный ресурс]. – Режим доступа: <https://www.digiteum.com/iot-supply-chain/>
5. Эффективная логистика последней мили [Электронный ресурс]. – Режим доступа: <https://www.trans.eu/ru/blog/logistika-posledney-mili/>
6. Складские роботы [Электронный ресурс]. – Режим доступа: <https://robotrends.ru/robopedia/skladskie-roboty>
7. Как беспилотные автомобили изменят логистику [Электронный ресурс]. – Режим доступа: <https://cargotime.ru/analitika/bespilotnye-avtomobili/>
8. Внедрение RFID технологии для автоматизации склада и проведения инвентаризации [Электронный ресурс]. – Режим доступа: <https://infostart.ru/1c/articles/626447/>

УДК 004.6; ГРНТИ 20.23.17

СОВРЕМЕННЫЙ ПОДХОД ОРГАНИЗАЦИИ ПРОЦЕССА ОБРАБОТКИ ИНФОРМАЦИИ В СИСТЕМАХ БОЛЬШИХ ДАННЫХ

А.А. Касымов

*Воронежский государственный технический университет,
Российская Федерация, Воронеж, kasimlele@live.ru*

Аннотация. В статье рассматривается рабочий процесс обработки информации в системах больших данных, который происходит в рамках стандартного потока обработки информации. Проанализированы и представлены базовые операторы. Предложенная методика позволит оптимизировать на уровне рабочего потока, что дает перейти на новый уровень взаимодействия между процессами, обеспечивая создания сквозного рабочего процесса обработки данных.

Ключевые слова: Apache Airflow, организации процесса обработки данных, системах больших данных, оркестрации данных.

A MODERN APPROACH TO ORGANIZING THE INFORMATION PROCESSING PROCESS IN BIG DATA SYSTEMS

A.A. Kasymov

*Voronezh State Technical University,
Russia, Voronezh, kasimlele@live.ru*

The summary. The article discusses the workflow of information processing in big data systems, which occurs within the framework of a standard information processing flow. The basic operators are analyzed and presented. The proposed methodology will optimize at the workflow level, which allows you to move to a new level of interaction between processes, ensuring the creation of an end-to-end workflow for data processing.

Keywords: Apache Airflow, data processing process organization, big data systems, data orchestration.

Рост объема видеоданных и количество операций над ними остается весомым, информация поступает из сотни источников. Процесс по перемещению данных по системам больших баз данных становится сложнее, что затрудняет процесс управление. Но по причине того, что данные стали активом, как капитальное имущество и интеллектуальная собственность.

Внедрение процессов, ориентированных на работу с данными, часто вызывает затруднения это связано с тем, что этот процесс требует, многократное усилия над координацией работы между разными системами и связывать аккуратно в единую экосистему, которая будет своевременной для последующего анализа или развертывания продукта.

Для решение сложившейся проблемы с организации работы по управлению процесса обработки данных в данной работе рассматривается оркестрация данных для оптимизации рутинных задач по обработке информации. Инструмент для оптимизации работы с данными был выбран инструмент Apache Airflow. Пример выполняемых задач по: сбору данных из различных источников, запуск процессов по консолидации данных, а также наблюдение за обслуживанием этих процессов. ETL (Extract, Transform, Load) — это подход к работе с данными, который состоит извлечение, преобразование и загрузка данных. Применение ETL дает :

- объединять различные формы данных в единое унифицированное виде, что упрощает поиск данных, а также выполнение задач по анализу данных;
- повышать продуктивность работы с данными, поскольку ETL систематизирует и повторно использует процессы, перемещающие данные;
- сравнивать образцы данных между исходной и целевой системой.

Этот подход предоставляет множество новых возможностей: анализ исторических записей для оптимизации процесса продаж, корректировка цен и запасов в реальном времени,

использование машинного обучения и искусственного интеллекта для создания прогнозных моделей, разработка новых потоков доходов, переход в облако и т. д.

То есть процесс, с помощью которого данные из нескольких систем собирают, преобразуют и хранят данные из миллионов транзакций в самых разных источниках данных и потоках в единое хранилище данных наглядно представлен на рисунке 1.



Рис. 1. Логика работы подхода ETL

Apache Airflow — это инструмент с открытым исходным кодом для организации сложных рабочих процессов и конвейеров обработки данных. Это платформа для программного планирования и мониторинга рабочих процессов для запланированных заданий.[1]

DAG — это ориентированный ациклический граф, у которого отсутствуют циклы, но могут быть параллельные пути, выходящие из одного и того же узла, а также он является сущностью, которая объединяет задачи в единый цепочку задач, где явно видны зависимости между узлами [2].

Реализация логики работы подхода ETL в виде классической DAG, где задача по анализу данных является конечной в цепочке и зависит от всех задач слева от него представлена на рисунке 2.

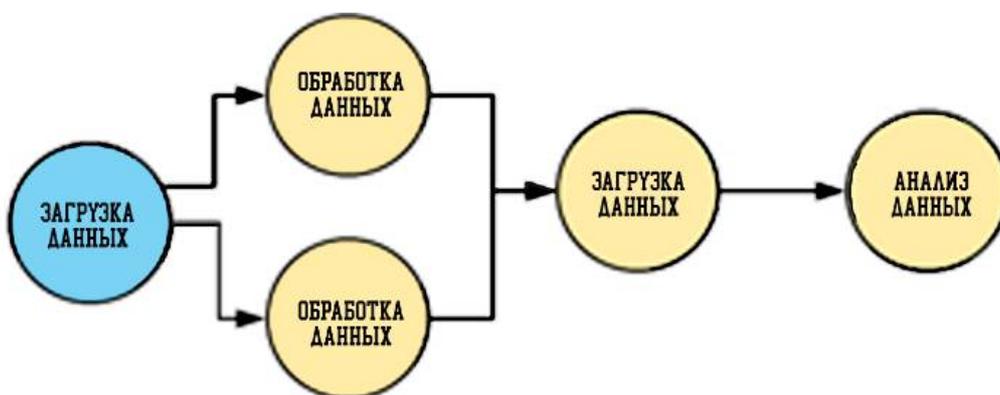


Рис. 2. Логика работы платформы

Цепочку задач состоит из операторов, где каждый оператор — это звено в цепочке задач. В данной работе был выбран PythonOperator так как он является более гибкий для реализации любого рода алгоритмов.

Планировщик, который обрабатывает как запуск запланированных рабочих процессов, так и отправку задач исполнителю для запуска. Исполнитель, который обрабатывает запущенные задачи. В установке Airflow по умолчанию это запускает все внутри планировщика, но большинство исполнителей, подходящих для производства, фактически передают выполнение задач рабочим. Веб-сервер с удобным пользовательским интерфейсом для провер-

ки, запуска и отладки поведения групп обеспечения доступности баз данных и задач. Папка файлов DAG, читаемая планировщиком и исполнителем (и любыми рабочими процессами, имеющимися у исполнителя). База данных метаданных используется планировщиком, исполнителем и веб-сервером для хранения состояния [3].

Реализация оркестрации на рабочем процессе обработки информации

Задачи выбранные для демонстрации работы оркестрации обработки данных:

- получения набор данных из интернета;
- две параллельные задачи по получение части данных, где одна получает первые 5 записей из набора данных, а вторая берет 5 последних записей из набора данных;
- 2 задачи производят отделение только требуемых столбцов по условию и создавали ещё 2 новых набор данных уже с нужными столбцами для последующих вычислений;
- финальная задача занималась объединение данных и получение среднего и максимальное значение из массива чисел в полученном набор данных.

Получения результата работы Apache Airflow: в графовом виде алгоритма работы представлен на рисунке 6.

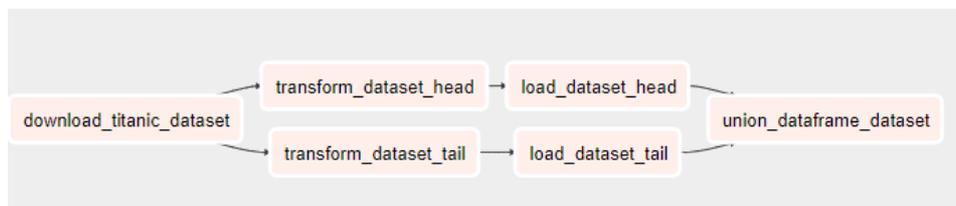


Рис. 6. Представление алгоритма в графовом виде

На рисунке 7 представлен работа алгоритма древовидная виде на платформе Apache Airflow.

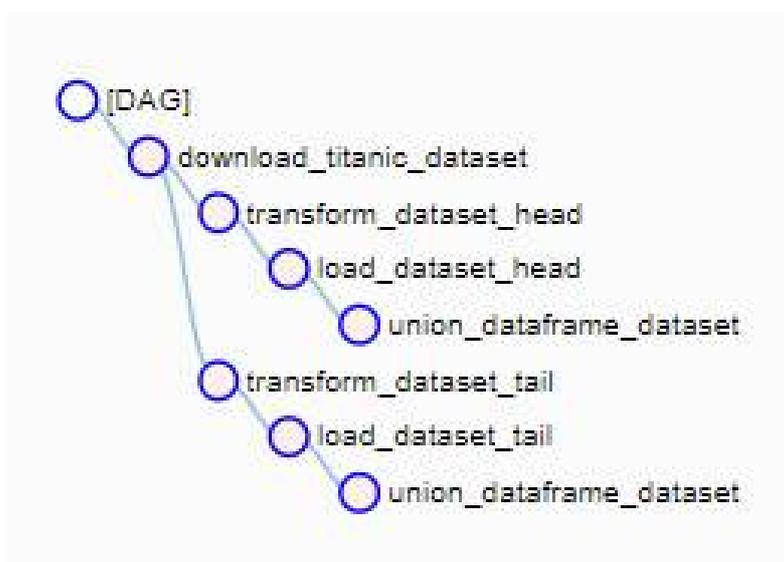


Рис. 7. Представление алгоритм работы в древовидный виде

На рисунке 8 показано отображение работы очереди выполнения задач, где каждая строка отвечает за определённую задачу и отображает статусы задач в цветовом виде.

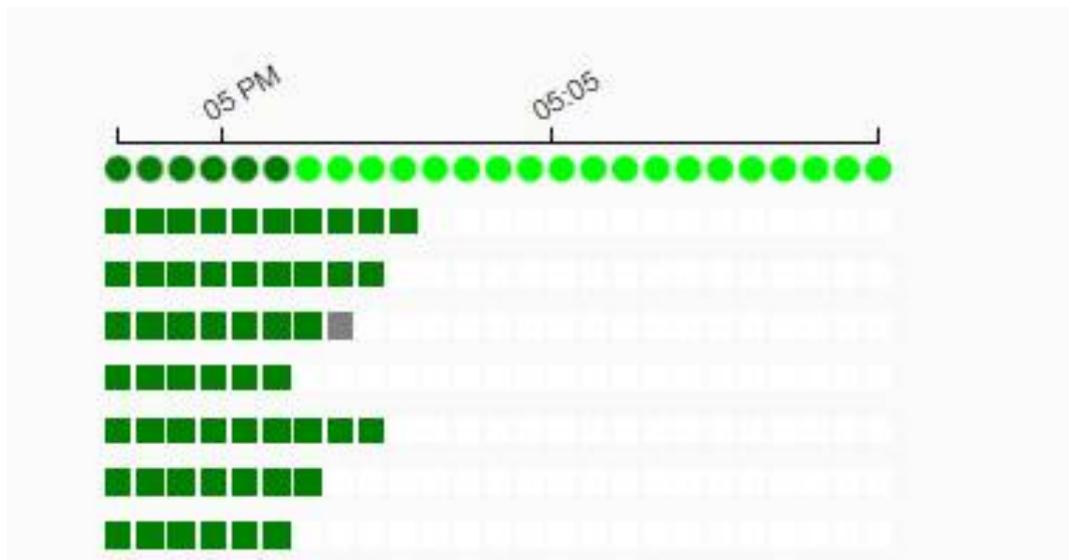


Рис. 8. Очередь выполнение задач

На рисунке 9 представлен график продолжительности работы задач, где видно, что самой продолжительной задачей является задача по загрузки набора данных из интернета.

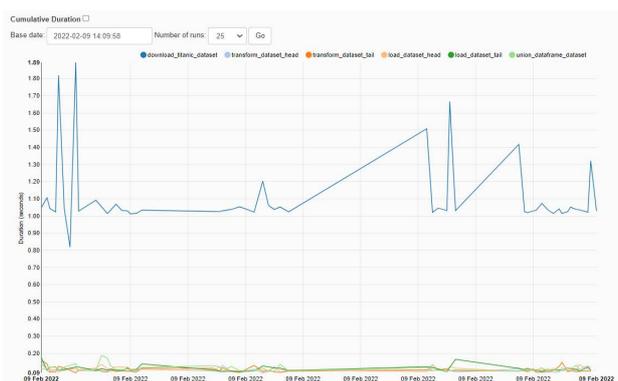


Рис. 9. График время продолжительности

На рисунке 10 представлен график времени загрузки, где видно, что с течение всей загрузки падает затраченное время на неё, а также по графику видно, что были падения во времени загрузки, это было связано с тем, что во время испытания платформы был предприняты действия в изменения логики работ задач.

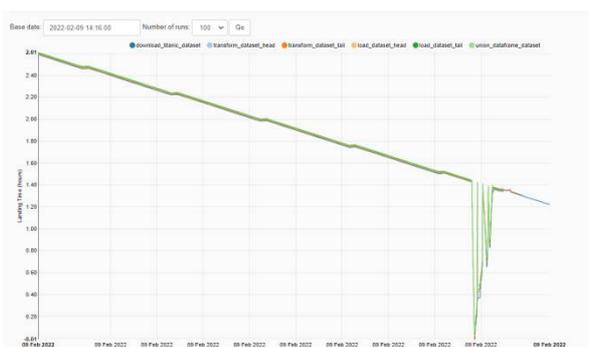


Рис. 10. График времени загрузки

На рисунке 11 продемонстрирован результат математических вычислений — это демонстрирует, что задача по оркестрации была выполнена верно. Также данный результат показывает, что данные и написанные Python операторы отработали, как и было задумано.

```
Dependencies all met for <TaskInstance: research_dag.union_dataframe_dataset 2022-02-09T02:23:00+00:00 [queued]>
Dependencies all met for <TaskInstance: research_dag.union_dataframe_dataset 2022-02-09T02:23:00+00:00 [queued]>

-----
Starting attempt 1 of 1

-----
Executing <Task(PythonOperator): union_dataframe_dataset> on 2022-02-09T02:23:00+00:00
INFO - Started process 18460 to run task
- Running %s on host %s <TaskInstance: research_dag.union_dataframe_dataset 2022-02-09T02:23:00+00:00 [running]> 73c18740a3f1
Среднее29.521650000000004,Максимум 71.2833
0 - Done. Returned value was: None
- Marking task as SUCCESS.dag_id=research_dag, task_id=union_dataframe_dataset, execution_date=20220209T022300, start_date=20220209T141044, end_date=20220209T141044
- [2022-02-09 14:10:54,005] {{local_task_job.py:103}} INFO - Task exited with return code 0
```

Рис. 11. Результат вывода данных Python оператора

Библиографический список

1. Apache Airflow: делаем ETL проще // habr.com URL: <https://habr.com/ru/post/512386/> (дата обращения: 27.01.2022).
2. Apache Airflow in 10 minutes // medium.com URL: <https://medium.com/swlh/apache-airflow-in-5-minutes-c005b4b11b26> (дата обращения: 27.01.2022).
3. Зачем дата-инженеру нужен оркестратор? // otus.ru URL: https://otus.ru/nest/post/722/?_gl=1*hh6f1l*_ga*WEtVdVJuMnFEcWl2QnlzVy1peEhyU2thRk4tcHRneEVfc3JYOTN1eUo3T2ZnY3VCaXk4VHliOXRtQVJHTFhsTw. (дата обращения: 23.01.2021).

УДК 004.415.25; ГРНТИ 50.41.25

ВЗАИМОДЕЙСТВИЕ КЛИЕНТСКОГО И СЕРВЕРНОГО ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ ПРИ ИССЛЕДОВАНИИ 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 и выступает как первичное средство сбора и обработки информации. Каждый прибор перед эксплуатацией проходит ряд проверок, одной из которых является корректное считывание солёности раствора. В работе производится описание процесса взаимодействия клиентского и серверного программного обеспечения на примерах считывания показателя солёности с поверяемого солемера.

Ключевые слова: автоматизированная система, удаленное управление, протокол взаимодействия, диаграмма последовательности.

INTERACTION OF CLIENT AND SERVER SOFTWARE FOR RESEARCH OF METROLOGICAL CHARACTERISTICS OF SALINOMETERS

P. Nikolov

*Utkin Ryazan State Radio Engineering University
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 which is a correct reading of the salinity of the solution. The paper discusses the process of interaction between client and server software using examples of salinity index reading from the checking salinometer.

Keywords: automated system, remote control, interaction protocol, sequence diagram.

Процесс поверки солемеров включает в себя как сбор данных с поверяемого прибора, так и со вспомогательных приборов (лабораторный солемер, термометр и т.д.). Взаимодействие клиентского программного обеспечения для исследования метрологических характеристик солемеров с приборами осуществляется через сервер.

Протокол общения клиентского и серверного программного обеспечения (ПО) состоит из определенных структур сообщений[1] (таблица 1).

Таблица 1. Структуры сообщений взаимодействия клиентского и серверного программного обеспечения

Тип сообщения	Описание	Структура сообщения
CALL	Запрос	[<Номер типа сообщения>, <Идентификатор>, <Действие>, {<Данные>}]
CALLRESULT	Ответ на запрос	[<Номер типа сообщения>, <Идентификатор>, {<Данные>}]
CALLERROR	Ошибка на запрос	[<Номер типа сообщения>, <Идентификатор>, <Код ошибки>, <Описание ошибки>, {<Данные о ошибке>}]

Целью работы является рассмотрение потоков событий взаимодействия частей системы на примере команды считывания солесодержания с поверяемого солемера.

Диаграмма последовательности

Диаграмма последовательности (sequence diagram) – это способ описания поведения системы «на примерах».

Фактически, диаграмма последовательности – это запись протокола конкретного сеанса работы системы (или фрагмента такого протокола). В объектно-ориентированном программировании самым существенным во время выполнения является пересылка сообщений между взаимодействующими объектами. Именно последовательность посылок сообщений отображается на данной диаграмме, отсюда и название.

На диаграмме последовательности применяют один основной тип сущностей – экземпляры взаимодействующих классификаторов (в основном классов, компонентов и действующих лиц), и один тип отношений – связи, по которым происходит обмен сообщениями. Предусмотрено несколько способов посылки сообщений, которые в графической нотации различаются видом стрелки, соответствующей отношению.

Важным аспектом диаграммы последовательности является явное отображение течения времени. В отличие от других типов диаграмм, кроме диаграмм синхронизации, на диаграмме последовательности имеет значение не только наличие графических связей между элементами, но и взаимное расположение элементов на диаграмме. А именно, считается, что имеется (невидимая) ось времени, по умолчанию направленная сверху вниз, и то сообщение, которое отправлено позже, нарисовано ниже [2].

Считывание солесодержания с солемера

Данный сценарий описывает наиболее часто используемый случай – единоразовое считывание показателя солёности с поверяемого солемера в благоприятной ситуации (сообщения отправляются без потерь, солемер подключен к серверу и отвечает на команды) (таблица 2, рисунок 1).

Таблица 2. Описание сценария считывания солесодержания с солемера

Тип	Описание
Название	Считывание солесодержания с солемера
Действующие лица	Клиентское ПО, серверное ПО, поверяемый солемер
Краткое описание	Получение показателя солёности раствора на клиентском ПО с поверяемого солемера
Предусловия	Клиентское и серверное ПО работают корректно и подключены к локальной сети, поверяемый солемер подключен к серверу
Постусловия	Клиентское ПО получает показатель солёности раствора
Основной поток (нормальное течение)	<ol style="list-style-type: none"> 1. Клиентское ПО отправляет запрос на сервер на получение солёности раствора. 2. Серверное ПО преобразует запрос от клиента в запрос солемеру. 3. Серверное ПО отправляет запрос солемеру. 4. Солемер получает запрос на считывание солёности раствора. 5. Солемер вычисляет солёность раствора. 6. Солемер отправляет ответ серверу на запрос считывания солёности раствора. 7. Серверное ПО преобразует ответ от солемера в ответ клиентскому ПО. 8. Серверное ПО отправляет ответ клиентскому ПО. 9. Клиентское ПО получает показатель солёности раствора.
Альтернативный поток (альтернативные течения)	1. Считывание солесодержания с солемера, который не отвечает или не подключен к серверу
Приоритет (Критично Важно Желательно):	Критично
Частота использования (Всегда Часто Иногда Редко Один раз):	Часто

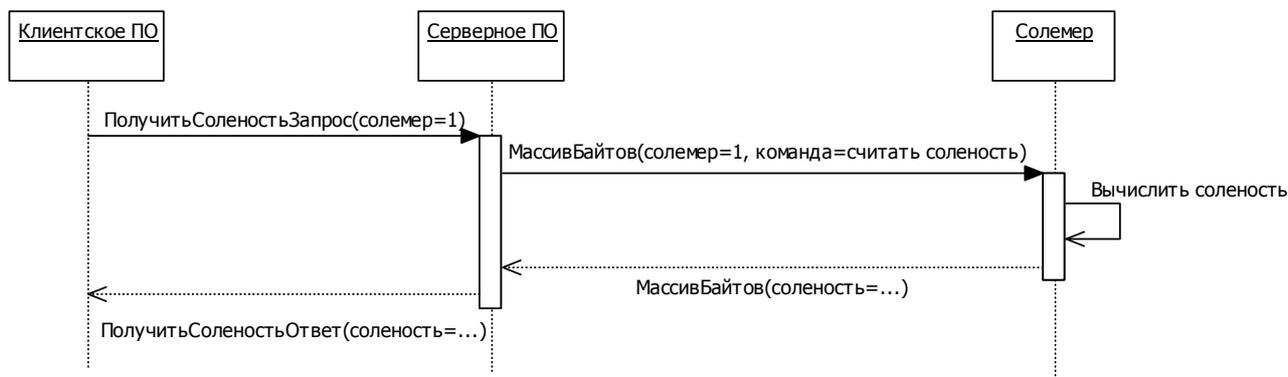


Рис. 1. Диаграмма последовательности считывания солесодержания с солемера

Считывание солесодержания с солемера, который не отвечает или не подключен к серверу

Данный сценарий описывает ситуацию при которой невозможно считать показатель солёности с поверяемого солемера (таблица 3, рисунок 2).

Наиболее часто встречаемые причины при которых невозможно считать данные с солемера:

1. Солемер не подключен к серверу.
2. Солемер не отвечает на команды от сервера.

Со стороны сервера, в такой ситуации, невозможно исключить одну из причин по причине взаимодействия сервера и солемера по интерфейсу RS-485.

Для избежания ситуации, когда серверу приходится бесконечно ожидать ответа от солемера, реализуется таймер ожидания ответа. При таймауте прекращается ожидание ответа от солемера и отправляется сообщение клиенту, информирующее о инциденте.

Таблица 3. Описание сценария считывания солесодержания с солемера, который не отвечает

или не подключен к серверу

Тип	Описание
Название	Считывание соледержания с солемера, который не отвечает или не подключен к серверу
Действующие лица	Клиентское ПО, серверное ПО, солемер
Краткое описание	Получение ошибки на клиентском ПО с сервера, информирующее, что солемер не ответил за допустимое время
Предусловия	Клиентское и серверное ПО работают корректно и подключены к локальной сети, поверяемый солемер отключен от сервера или не отвечает
Постусловия	Клиентское ПО получает уведомление о недостижимости или о некорректной работе солемера
Основной поток (нормальное течение)	<ol style="list-style-type: none"> 1. Клиентское ПО отправляет запрос на сервер на получение солености раствора. 2. Серверное ПО преобразует запрос от клиента в запрос солемеру. 3. Серверное ПО отправляет запрос солемеру. 4. Серверное ПО запускает таймер на ожидание ответа от поверяемого солемера. 5. На сервере срабатывает таймаут ожидания ответа. 6. Серверное ПО отправляет уведомление клиентскому ПО: солемер не ответил.
Альтернативный поток (альтернативные течения)	1. Считывание соледержания с подключенного солемера
Приоритет (Критично Важно Желательно):	Важно
Частота использования (Всегда Часто Иногда Редко Один раз):	Иногда



Рис. 2. Диаграмма последовательности считывания соледержания с солемера, который не отвечает или не подключен к серверу

Выполнение сценария на считывание соледержания

Основной частью процесса проверки солемера является многократное считывание показателя солености с определенным интервалом, после каждого считывания показатель сравнивается с эталонными данными. Автоматизация данной части достигается за счет сценариев проверки, например, необходимо считать 10 показаний с поверяемого солемера с интервалом 1 мин. Сценарий отправляется на сервер для создания задачи на считывание показателя солености. Далее сервер выполняет задачу и, после каждого считывания показателя солености, отправляет информацию на клиент (таблица 4, рисунок 3).

Таблица 4. Описание случая выполнения сценария на считывание солесодержания

Тип	Описание
Название	Выполнение сценария на считывание солесодержания
Действующие лица	Клиентское ПО, серверное ПО, солемер
Краткое описание	Получение набора показателей солёности раствора на клиентском ПО с поверяемого солемера
Предусловия	Клиентское и серверное ПО работают корректно и подключены к локальной сети, поверяемый солемер подключен к серверу
Постусловия	Клиентское ПО получает набор показателей солёности раствора
Основной поток (нормальное течение)	1. Клиентское ПО отправляет сценарий считывания солёности на сервер. 2. Серверное ПО создает задачу. 3. Серверное ПО уведомляет клиента о создании задачи. 4. Серверное ПО выполняет сценарий: отправляет команду солемеру и, после получения ответа от солемера, отправляет информацию клиенту.
Альтернативный поток (альтернативные течения)	
Приоритет (Критично Важно Желательно):	Критично
Частота использования (Всегда Часто Иногда Редко Один раз):	Часто

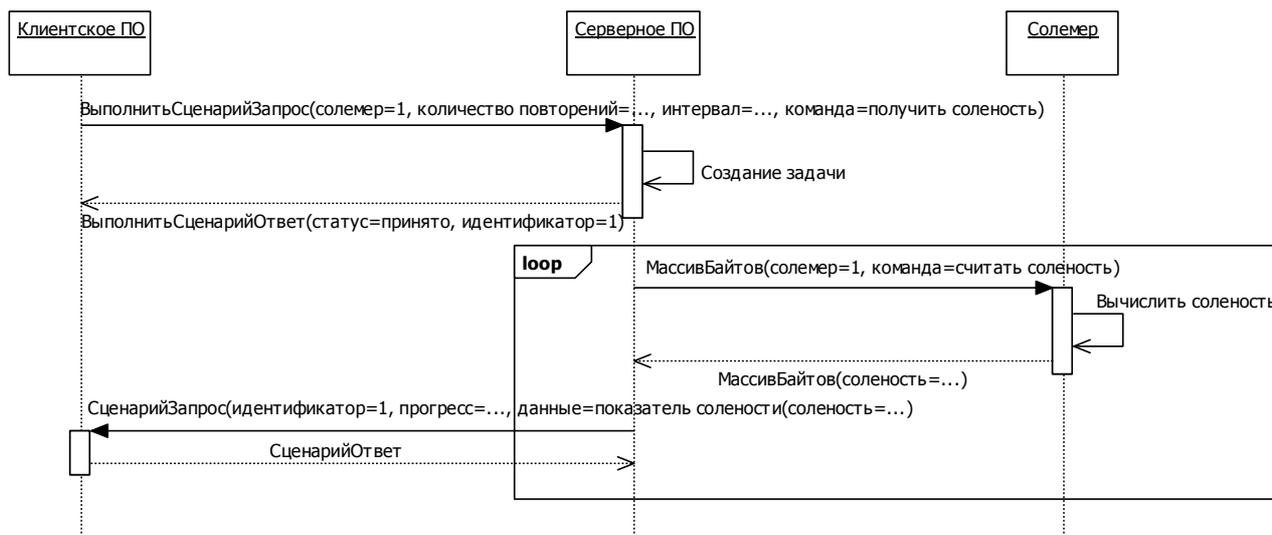


Рис. 3. Диаграмма последовательности выполнения сценария на считывание солесодержания

Заключение

В статье рассмотрен процесс взаимодействия клиентского и серверного программного обеспечения на примерах считывания солесодержания раствора в различных ситуациях. Для данных сценариев были описаны диаграммы последовательности на унифицированном языке моделирования (UML).

Библиографический список

1. Николов П. Разработка программного обеспечения для исследования метрологических характеристик солемеров // Материалы VI Всероссийской научно-технической конференции «Актуальные проблемы современной науки и производства». – Рязань: ИП Коняхин А.В. (Book Jet), 2021 – С. 209-215.
2. Иванов Д. Ю., Новиков Ф. А. Основы моделирования на UML: Учеб. пособие. – СПб.: Изд-во Политехн. ун-та, 2010. – 249с.

УДК 621.396; ГРНТИ 27.03.33

ОБЗОР СОВРЕМЕННЫХ ПОДХОДОВ К ОЦЕНКЕ РИСКОВ НАУЧНО-ОБРАЗОВАТЕЛЬНОГО КОМПЛЕКСА

Е.Н. Евдокимова*, И.В. Устинов**

*Рязанский государственный радиотехнический университет имени В.Ф. Уткина,
Российская Федерация, Рязань, e008en@mail.ru

**Рязанский государственный радиотехнический университет имени В.Ф. Уткина,
Российская Федерация, Рязань, ustinov.ilya93@gmail.com

Аннотация. В работе рассматриваются обзор современных подходов к управлению рисками. Рассмотрен различный математический аппарат и представлены имеющиеся наработки по данной теме.

Ключевые слова: риск-менеджмент, управление рисками, нечеткая логика, нечеткие множества.

REVIEW OF MODERN APPROACHES TO RISK ASSESSMENT OF SCIENTIFIC AND EDUCATIONAL COMPLEX

E.N. Evdokimova*, I.V. Ustinov**

*Ryazan State Radio Engineering University named after V.F. Utkin,
Russian Federation, Ryazan, e008en@mail.ru

**Ryazan State Radio Engineering University named after V.F. Utkin,
Russian Federation, Ryazan, ustinov.ilya93@gmail.com

Annotation. The paper considers an overview of modern approaches to risk management. Various mathematical apparatus is considered and the available developments on this topic are presented.

Key words: Risk management, risk management, fuzzy logic, fuzzy sets.

Актуальность проблем обеспечения комплексной безопасности в организациях в Российской Федерации начала активно обсуждаться с 2010х годов. Это связано с обострением экономического кризиса в США, который в свою очередь пошатнул валютное миростройство и тем самым заставил правительства различных стран, пересмотреть вопросы обеспечения безопасности организаций в целом. В России после ряда внешне политических событий и наложения санкций также обострились требования к обеспечению комплексной безопасности. Ориентация на Евросоюз способствовала частичному принятию международных стандартов финансовой отчетности, вместе с которой и началась новая глава использования такого инструмента обеспечения комплексной безопасности как риск-менеджмент. Рассматривая инструменты, которые предлагаются к использованию, обнаруживается ряд проблем, к которым можно отнести существенные различиями в развитии бизнеса и государственного устройства, частоту использования таких инструментов и прочие. Одним из важных нововведений является то, что на уровне законодательства [2] организации обязали осуществлять риск-менеджмент. Однако, существующие механизмы в целом не подходят для образовательных и научных организаций в силу их специфики и особенностей.

Анализируя различные подходы к оценке и построению риск-менеджмента, следует акцентировать внимание на одной схожая для всех проблеме - отсутствие какого-либо внятного механизма управления рисками для образовательных и научных учреждений во всем мире. Судя по различным источникам [2-8] внимание к этим вопросам уделяется в США, Евросоюзе, Австралии и Израиле. Однако, как показывает практика до сих пор не существует четко отработанного механизма. На сегодняшний день, в мире доминирует западный подход. Как правило, в открытом доступе нет готовых кейсов по управлению рисками на предприятии. Обучающие материалы в свою очередь обходятся общими фразами, например, определить риски, рассчитать их, застраховать или принять. Однако, если определить риски еще представляется возможным, то с оценкой могут возникнуть определенные трудности.

В частности, Американская компания CHAS помогает клиентам и поставщикам обес-

печить соблюдение требований в различных областях управления рисками по всей цепочке поставок. Благодаря большому опыту работы сформирована общая стандартная система оценки рисков. В разделе помощь и советы CHAS предлагает полное руководство по управлению рисками [9]. В разделе «какие бывают оценки рисков» подчеркивается, что оценка рисков имеет решающее значение. Компания рекомендует производить следующие виды оценок риска:

1. Качественная система оценки риска. Является самой большой категорией по мнению CHAS. Как правило, в данном разделе используется личный опыт и суждение эксперта при выявлении опасностей.
2. Количественная система оценки рисков предполагает использование матриц. Матрица помогает присваивать значение вероятности и серьезности рисков. Самые простая матрица, которая может быть использована для анализа – 3x3. Для оценки вероятности принята шкала со значениями маловероятно, вероятно, высокая вероятность. Оценка серьезности со значениями небольшая серьезность, серьезный и опасный. Для расчета уровня риска предлагается умножить серьезность на вероятность.
3. Общая система оценки рисков связана с рабочими задачами и действиями. Тут подразумевается заимствование имеющихся методик для уменьшения дублирования процессов.
4. Динамическая оценка рисков проводится на месте при непредвиденных обстоятельствах.

Отечественные компании также ориентируются на западный подход. Например, ООО «Газпромнефть-Снабжение» выпустило методические рекомендации по управлению рисками, которая очень тесно коррелирует с методиками CHAS. В методических рекомендациях «Пять шагов» анализ опасностей и оценка рисков перед началом работы» [10] приведены критические факторы и причины происшествий за период с 2014 года, программа «пять шагов – оцени риск», анализ опасностей и др.

Описание тяжести последствий		Вероятность					
		A	B	C	D	E	
		Событие не имело места в Компании за последние 10 лет	Событие имело место в Компании 1 раз за последние 10 лет	Событие имело место 1 раз в ДО или более одного раза в Компании за последние 10 лет	Событие имело место 1 раз в филиале или более 1 раза в ДО за последние 10 лет	Событие имело место более 1 раза в филиале за последние 10 лет	
Тяжесть	1	Незначительные травмы или случаи ухудшения здоровья, не оказывающие влияние на производительность труда и на жизнедеятельность	Blue	Blue	Blue	Blue	Blue
	2	Травмы или обратимое ухудшение здоровья с потерей трудоспособности до 15 дней	Blue	Blue	Blue	Yellow	Yellow
	3	Тяжелая травма или ухудшение здоровья с потерей трудоспособности более 15 дней, включая необратимый ущерб для здоровья	Blue	Blue	Yellow	Yellow	Red
	4	От 1 до 3 случаев постоянной полной нетрудоспособности или несчастных случаев с летальным исходом	Blue	Yellow	Yellow	Red	Red
	5	Более чем 3 летальных исхода в результате травмирования или профессионального заболевания	Yellow	Yellow	Red	Red	Red

Рис. 1. Матрица определения уровня риска

В практическом пособии «Оценка рисков на рабочем месте» [11] описан опыт Фин-

ляндии по охране труда, в частности показана поэтапная модель оценки и управления рисками, даны признаки «хорошей» оценки рисков и предложен метод выбора объектов оценки. Авторы указывают, что разделение и обособление рисков в процессе оценки с больших на малые значительно упрощает весь процесс и позволяет анализировать каждый объект отдельно. На рисунке 2 приведен пример разделения и обособления оценки рисков.

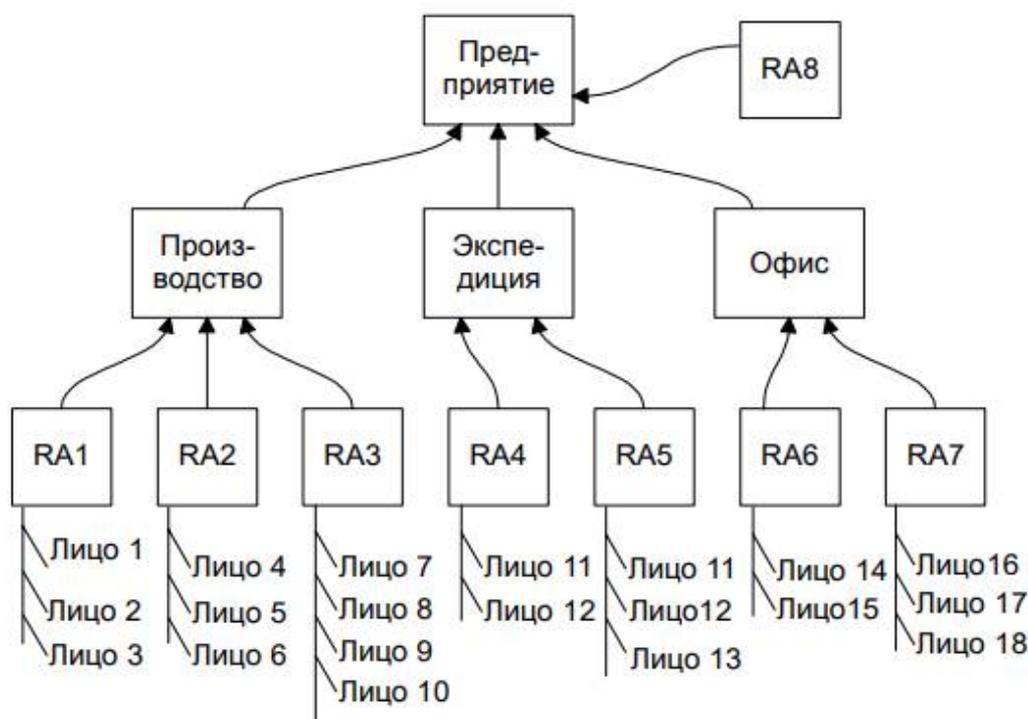


Рис. 2. Пример разделения и обособления оценки рисков

В остальном предлагается использовать общепризнанную методику, которая предполагает выявление опасностей, определение величины риска при помощи экспертных методов, принятие решения о значимости риска, выбор мероприятий для устранения, предотвращения или страхования риска. В качестве инструментов используются анкеты с перечнем вопросов. Для принятия решения также рекомендуют использовать матрицу, таблицу уровня угроз, которая очень похожа на методику CHAS. Однако стоит отметить, что рекомендуется создавать профиль риска, который учитывает количество выявленных опасностей и определяет весовой коэффициент в процентном соотношении.

Е.И Тымуль в своей работе [12] отмечает важность качественного анализа рисков для энергетических предприятий. Количественная оценка риска – продолжение качественного анализа для количественного определения ущерба.

Компания IBM использует автоматизированную систему управления рисками встроенную платформу IBM Engineering Lifecycle Management предназначенную для разработки современных продуктов и программного обеспечения с высоким уровнем сложности. Принятые в платформе алгоритм вычисления количественной оценки рисков базируется на расчете средней арифметической значении вероятности, влияния и текущего влияния. Показатель важности в расчет не входит. Показатели вероятность, влияния и текущего влияния предлагается оценивать по трёхбалльной шкале. Также отмечается, что общая количественная оценка риска – осуществляется по пятибалльной шкале. Предлагается формула перевода трёхбалльной в пятибалльную шкалу. Кроме того, присутствует система динамической оценки риска, при котором, оценка производится исходя из predetermined параметров.

Инструментом расчета служит взвешенная средняя арифметическая. Оценка производится по каждой строке по значению «важность» [13].

Чаще всего при анализе рисков применяются методы сценарного анализа, математической статистики, методы аналогий, аналитики и др.

Таблица 1. Обзор анализа рисков [14-16]

№ п/п	Тип анализа	Описание	Преимущества	Недостатки
1	Сценарный метод Монте-Карло	Метод «дерева решений» и диаграмма «торнадо»	Обеспечивает наглядность Просты в использовании	При увеличении переменных, возрастает объем расчетов, увеличивая вероятность ошибок
2	Математическая статистика	Основан на расчетах дисперсии, среднеквадратического отклонения, вариации, корреляции и математического ожидания.	Объективность Точность Простота	Требования в наличии большого объема точных исходных данных Не учитываются внутренние и внешние факторы
3	Аналитические методы	Использование абсолютных и относительных показателей в статистической обработке	Проведение всесторонней оценки надежности и эффективности. Простота Надежность эффективность	Необходимость принять эталонные значения Требования в наличии большого объема точных исходных данных
4	Метод оценки потерь	Использование экспертных оценок, ранжирование, непосредственная оценка, последовательное сравнение, парное сравнение	Эффективны при решении сложных, нестандартных и неформальных проблем.	Трудности проведения экспертного опроса, вероятность ошибок при отборе экспертов, гарантированность знаний и наличие субъективизма.
5	Метод аналогов	Поиск сходных ситуаций, проектов и событий.	Простой	Подвержен высокой степени субъективизма. Ошибки в решениях других проектов
6	Методы игр	Критерии Вальда, Сэвиджа, Гурвинца, Байеса, Лапласа и др. Применяется в условиях неопределенности.	Возможность учитывать различные факторы.	Сложность в определении воздействия неконтролируемых факторов, в особенности внешней среды.

Особый интерес представляет работа Доан Дык Ня [17], который проводил оценку рисков в проектах подводных трубопроводов в условиях шельфа Вьетнама. Автор представил

вил элементы нечеткой логики и нечетких множеств в контексте анализа рисков. Использована пятерка $\{x, T(x), X, G, M\}$ где x - имя переменной; $T(x)$ — множество имен лингвистических значений переменной x , каждое из которых является нечеткой переменной на множестве X ; G - синтаксическое правило для образования имен значений x ; M - семантическое правило для ассоциирования каждой величины значения с ее понятием. Также проведена классификация факторов риска по фазам жизненного цикла проекта, что позволило определять риски на каждом этапе и в каждом подфазах проекта. По итогам исследования разработан метод оценки риска проектов строительства и эксплуатации подводных трубопроводов с использованием системы нечеткого вывода. Данная система похожа на сценарные методы. Относительную оценку рисков следует определять формулой:

$$OOP = \frac{IS}{LIF} \quad (1)$$

где IS – индекс суммы;

LIF – индекс утечки

В свою очередь индексы определяются формулами 2 и 3

$$IS = TDP + C + D + IO, \quad (2)$$

где: TDP – сторонние повреждения;

C – коррозия; D – проектирование;

IO – ошибки при эксплуатации.

$$LIF = PH * LV * DI * RE, \quad (3)$$

где PH – опасность транспортируемого продукта

LV – объем утечки

DI – дисперсия

RE – рецепторы.

Структура модели оценки рисков с использованием системы нечеткого вывода показана на рисунке 3.

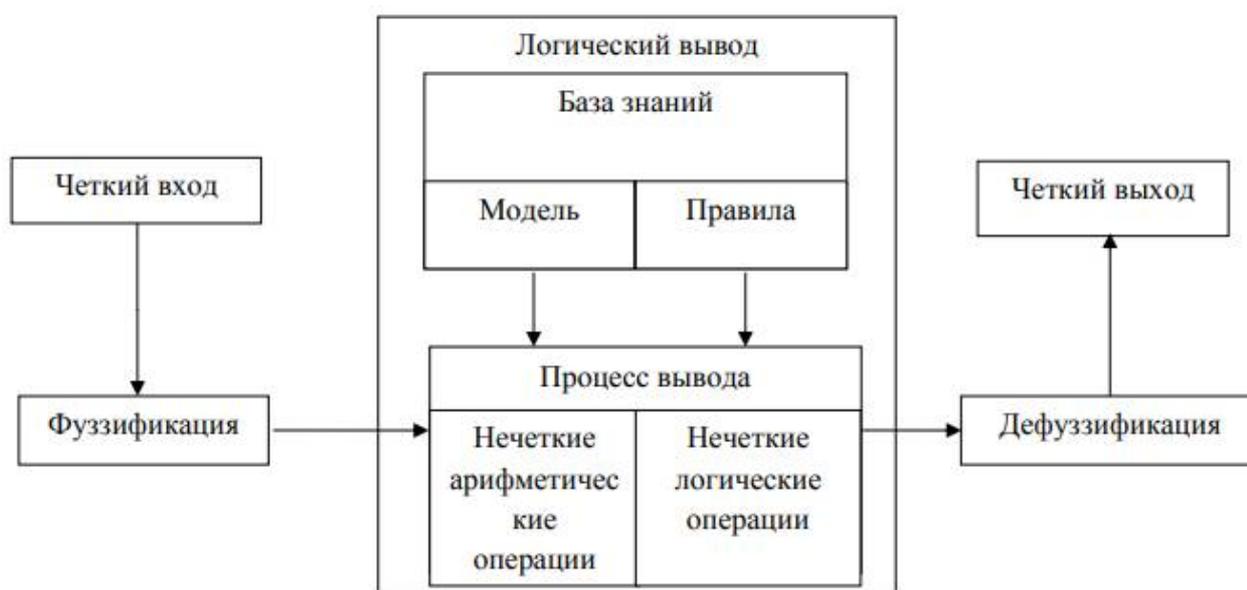


Рис. 3. Структура системы нечеткого вывода [17]

Данная методика используется в построении модели нечеткой оценки рисков и апробирована в среде MATLAB, для каждой величины и индекса сформированы правила.

На основании вышеизложенного можно сделать вывод, что западный инструментарий прост в использовании, не предполагает каких-либо специальных знаний в области управления рисками. Однако важно понимать область, в которой осуществляется процедуры риск-менеджмента. Это в первую очередь связано с определением внешних и внутренних рисков, которые так или иначе могут влиять на объект исследования.

Нами проведены определенные наработки по управлению рисками научно-образовательного комплекса. За основу взят математический аппарат, построенный на нечеткой логике и нечетких множествах.

Нами составлена иерархическая модель управления рисками, которая включает в себя разделение по видам деятельности, функциональные сферы и рискам. Аппарат нечетких множеств позволяет определять вес критериев, которые в свою очередь могут быть использованы для оценки весомости риска. В совокупности с экспертными оценками, данный метод на наш взгляд может дать более точную оценку риска. Возможность оценки риска в динамике на наш взгляд перспективнее, так как способна в режиме реального времени давать рекомендации для принятия решения.



Рис. 4. Примерная модель управления рисками научно-образовательного комплекса

Построенная нами модель является статической и отражает ситуацию состояния НОК на момент проведения исследования. Сложность и неоднозначность модели выражается в том, что нами обнаружены риски, которые имеют смешанный характер. Аппарат нечетких множеств имеет возможность учета таких рисков. Метод экспертных оценок в свою очередь, может помочь определить к какой из категорий данный риск относится больше.

Библиографический список

1. ГОСТ Р ИСО/МЭК 31010-2011. «Менеджмент риска. Методы оценки риска» [Электронный ресурс]. URL - <https://docs.cntd.ru/document/1200090083>
2. YOSSI RAANAN. Risk Management in Higher Education - Do We Need it? [Электронный ресурс]. <https://www.sijm.it/uploads/2018/04>
3. Centk J.D. A report card for risk management in higher education for two year colleges in Minnesota, Wisconsin and Michigan. // A dissertation Submitted to the Graduate Faculty of the North Dakota State University of Agriculture and Applied Science. 2017.
4. The State of Enterprise Risk Management at Colleges and Universities Today [Электронный ресурс]. URL – <https://www.edurisksolutions.org/WorkArea/DownloadAsset.aspx?id=1944>
5. Meeting the challenges of enterprise risk management in higher education. [Электронный ресурс]. URL – <https://files.eric.ed.gov/fulltext/ED524480.pdf> (дата обращения 31.03.2021)
6. Mullin C. M. Evolving practices and emerging innovations in community college finance // New Direction for Community Colleges. 2014. №168. p. 115–125.
7. Australian Capital Territory Insurance Authority. Guide to risk management: Insurance and risk management strategies. [Электронный ресурс]. URL – http://www.riesgooperacional.com/docs/docs/Estandar_ANZ.pdf
8. Ministry of the Premier and Cabinet. Guidelines for managing risk in the Western Australian public sector: Better management. [Электронный ресурс]. URL – www.rdec.gov.tw/DO/DownloadControllerNDO.asp?CuAttachID=17523
9. The 5 steps to risk assessment explained. CHAS. [Электронный ресурс]. URL – <https://www.chas.co.uk/blog/5-steps-to-risk-assessment/>
10. ООО «Газпромнефть-Снабжение». «Пять шагов» Анализ опасностей и оценка рисков перед началом работы. Управление ПЭБ, ОТ и ГЗ. [Электронный ресурс]. URL – https://supply.gazpromneft.ru/upload/iblock/136/d0_9f_d1_80_d0_b5_d0_b7_d0_b5_d0_bd_d1_82_d0_b0_d1_86_d0_b8_d1_8f_20_d0_bf_d1_80_d0_be_d0_b3_d1_80_d0_b0_d0_bc_d0_bc_d1_8b_20_205_20_d1_88_d0_b0_d0_b3_d0_be_d0_b2.pdf
11. Муртонен. М. Оценка рисков на рабочем месте – практическое пособие. Технический исследовательский центр Финляндии. Министерство социального обеспечения и здравоохранения, отдел охраны труда. Тампере, Финляндия. 2007 год.
12. Тымуль Е.И. Выбор метода качественного и количественного анализа рисков для предприятий энергетики. БНТУ. 2021 год.
13. Алгоритм вычисления количественной оценки рисков. [Электронный ресурс]. URL – <https://www.ibm.com/docs/ru/elm/6.0.6?topic=risk-how-assessment-scores-are-calculated>
14. Буянов, В. П. Рискология: управление рисками / В. П. Буянов, К. А. Кирсанов, Л. А. Михайлов. М.: Экзамен, 2002. 382 с
15. Гранатуров, В. М. Экономический риск: сущность, методы измерения, пути снижения / В. М. Гранатуров. М.: Дело и сервис, 1999. 112 с.
16. Хохлов, Н. В. Управление риском / Н. В. Хохлов. М.: ЮНИТИ, 1999. 239 с.
17. ДОАН ДЫК НЯ. Оценка рисков в проектах подводных трубопроводов в условиях шельфа Вьетнама. Автореферат. Москва – 2015.

УДК 004.9; ГРНТИ 82.33.13

**ФОРМИРОВАНИЕ РЕГЛАМЕНТОВ ОБЕСПЕЧЕНИЯ БЕЗОПАСНОСТИ
ДЛЯ АВТОМАТИЗИРОВАННОГО КПП ОБЩЕОБРАЗОВАТЕЛЬНЫХ
УЧРЕЖДЕНИЙ**

Д.В. Пономаренко, В.В. Коваленко

*Сочинский государственный университет,
Россия, Сочи, vlvvas@mail.ru*

Аннотация. В работе представлена разработка автоматизированного контрольно-пропускного пункта (КПП) для общеобразовательного учреждения с расширенным набором регламентов обеспечения безопасности в соответствии с разработанным бизнес-процессом управления деятельностью КПП. Регламентная документация сформирована для обслуживающего персонала в среде пакета “Business Studio” на основе процессного подхода в управлении.

Ключевые слова: бизнес-процесс, регламент, системы управления доступом, процессный подход.

FORMATION OF SAFETY REGULATIONS FOR AUTOMATED CHECKPOINT OF GENERAL EDUCATIONAL INSTITUTIONS

D.V. Ponomarenko, V.V. Kovalenko

Sochi State University, Russia, Sochi, vlvas@mail.ru

The summary. The paper presents the development of an automated checkpoint (checkpoint) for a general education institution with an expanded set of security regulations in accordance with developed business process for managing the checkpoint. Regulatory documentation for service personnel generated in the environment of the “Business Studio” package based on the process approach to management.

Keywords: business process, regulations, access control system, process approach.

Последние десять лет в современном обществе наблюдается пугающая тенденция. Спокойную и размеренную жизнь регулярно будоражат новости об очередной чрезвычайной ситуации. Особенно заметен рост случаев нападения на обучающихся в общеобразовательных учреждениях, так называемый «скулшутинг».

При этом данные события объединяет практически одинаковый план совершения преступлений. Каждый раз преступник, не встречая ни малейшего препятствия, пронес оружие, взрывчатку и все необходимое, через контрольно-пропускные пункты. Анализ ситуации показывает, что в большинстве школ, гимназий, колледжей и даже институтов, пропуск на территорию осуществляется непосредственно охранником без применения каких-либо технических устройств, а журнал посещений и другая отчетность ведется вручную, если вообще ведется.

Сложившееся положение дел прямо указывает на устаревшие регламенты обеспечения безопасности в образовательных учреждениях и необходимости применения систем контроля и управления доступом (СКУД или PACS). Однако сравнительный анализ ценовой политики наиболее известных СКУД в нашей стране (Классический Gate, Octagram, Болид, Smartec, Sigur Rusguard, Бастион, Hikvision, Perco, КОДОС и Parsec) свидетельствует о том, что обычные школы и колледжи не всегда в состоянии их приобрести и эксплуатировать в соответствии с требуемыми регламентами обеспечения безопасности (рис.1).

Для устранения проблем существующих систем контроля и управления доступом в работе разработан доступный по стоимости СКУД, который способен обеспечить более эффективный и совершенный регламент обеспечения безопасности за счёт фундаментального переосмысления существующих регламентных документов с учётом современных вызовов. Регламенты сгенерированы на основе процессного подхода в управлении в среде российского пакета бизнес-моделирования “Business Studio”.

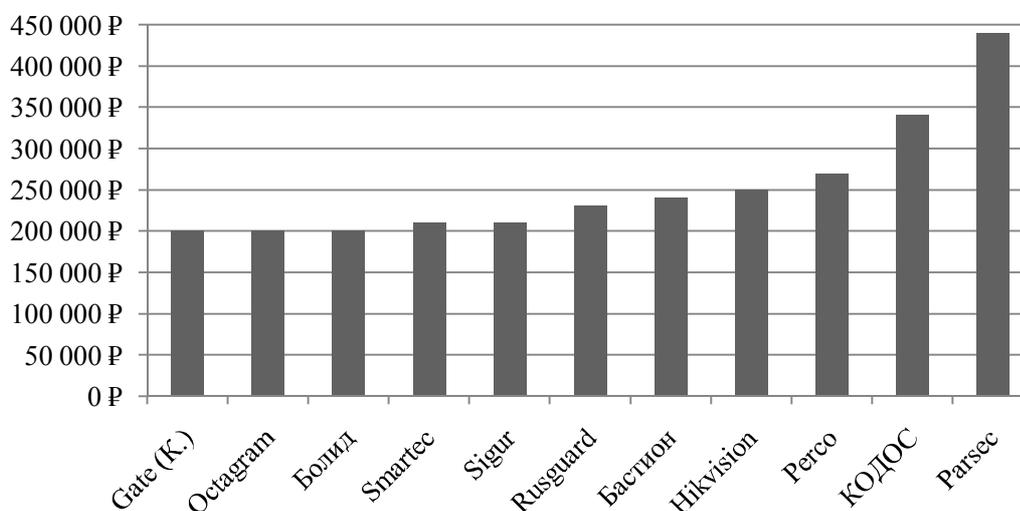


Рис. 1. Сравнительный анализ ценовой политики СКУД

Основная проблема заключалась в том, чтобы максимально расширить регламент обеспечения безопасности. Для этого разработан бизнес-процесс, в который заложили необходимые требования к регламенту, используя процессный подход в управлении (рис. 2) [1, 2].

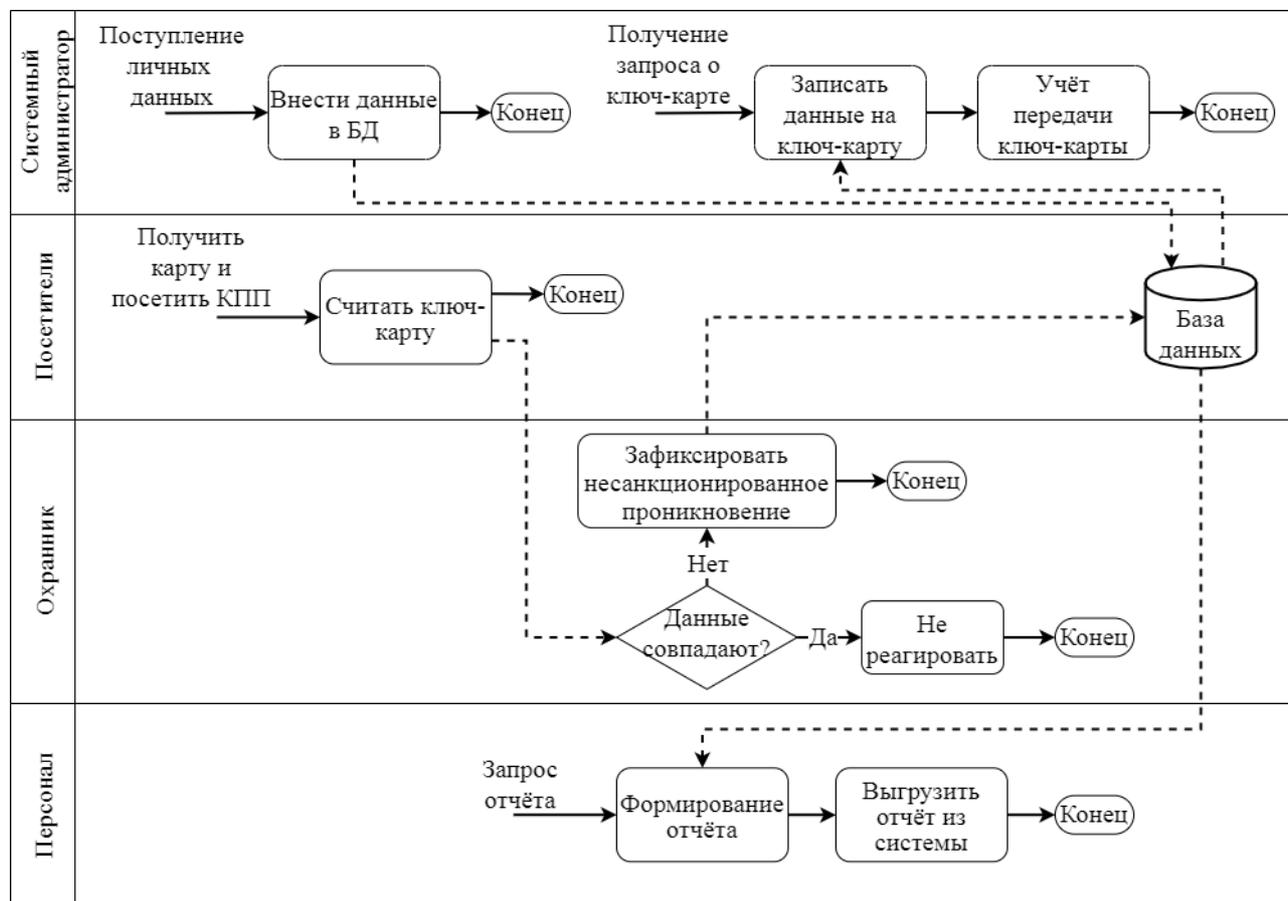


Рис. 2. Бизнес-процесс управления автоматизированным КПП

На карту-идентификатор заносится информация о пользователе из сформированной базы данных и выдаётся посетителю. Каждый раз, проходя КПП, посетителю необходимо поднести карту к считывателю. При совпадении идентификаторов исполнительное устройство, например, турникет, разблокирует проход. Сведения о санкционированном посещении записываются в базу данных и выводятся на экран на охранном посту с полной информацией о посетителе, включая его фото идентификацию. В случае несовпадения идентификатора карты исполнительные устройства блокируются, а на охранный пост передается оповещение о попытке несанкционированного проникновения. Для корректной работы системы, в СУБД формируется база данных посетителей общеобразовательного учреждения.

В соответствии с процессным подходом в управлении на основе сформированных показателей эффективности, исполнителей и владельца сквозного бизнес-процесса в среде российского пакета бизнес-моделирования "Business Studio" сгенерирована регламентная документация в составе: регламент бизнес-процесса, положение о подразделении, должностные инструкции для исполнителей и матрица ответственности [3]. Наличие регламентной документации обеспечивает четкое выполнение регламентов безопасности общеобразовательного учреждения.

В аппаратной реализации системы RFID-модуль RC522 и платформа Arduino Uno R3 соединяются через шлейфы с разъемами «Вилка – розетка», чтобы контроллер распознал считыватель как аппарат (рис.3). Далее микроконтроллер подключается к персональному компьютеру через USB-кабель типа «А – В», так как всё последующее взаимодействие будет протекать через персональный компьютер или в связке с ним.

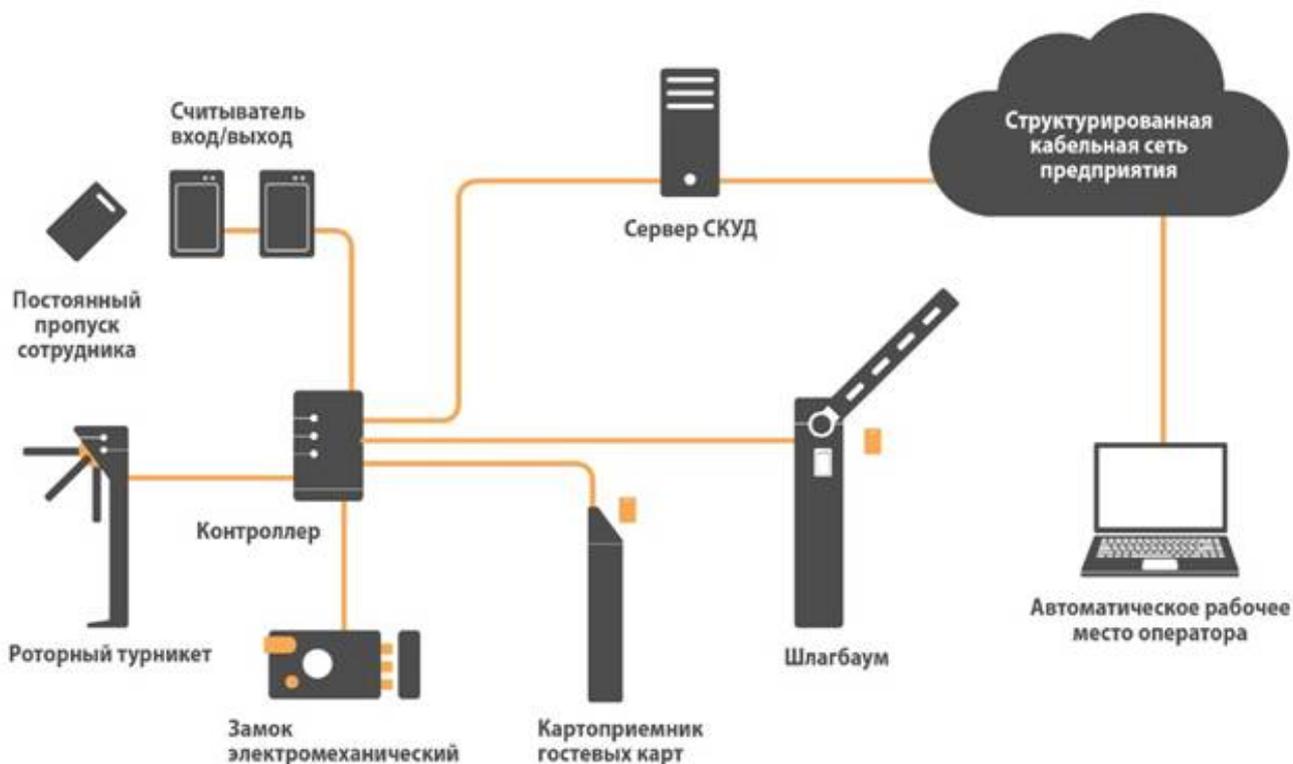


Рис. 3. Классическая структура СКУД

Для перепрограммирования микроконтроллера и задания ему управляющих команд в среде Arduino IDE на языке Arduino C разработан скетч (прошивка), что обеспечивает платформе программно обнаруживать считыватель, исполнительные устройства и возможность взаимодействовать с ними. Скетч необходим для операций записи данных на ключ-карты, последующее считывание этих данных, их обработку и вывод на монитор, а также передачи команд исполнительным устройствам.

В среде Microsoft Visual Studio на языке C# создаётся пользовательское приложение с подключением базы данных через DataGridView. Это облегчит работу с системой и СУБД благодаря простому, интуитивно понятному и наглядному интерфейсу.

Для разработки системы использовались следующие комплектующие:

- контроллер: платформа Arduino Uno R3 на базе микроконтроллера ATmega328P;
- считыватель: RFID-модуль RC522;
- идентификатор: магнитная бесконтактная карта Mifare Standard 1K;
- бесплатные среды разработки Microsoft Visual Studio и Arduino IDE;
- СУБД Microsoft Access.

При разработке системы для общеобразовательного учреждения с потоком в 1000 человек и один КПП затраты составят приблизительно 17 000 рублей.

Библиографический список

1. Kovalenko V.V. Automation of quality management system on the basis of modern technologies / V.V. Kovalenko, V.V. Kovalenko // European Journal of Computer Science. - 2015. № 1 (1). С. 41-47.
2. Коваленко В.В. Проектирование информационных систем / В.В. Коваленко. - М.: ИНФРА-М, 2021. (2-е изд., испр. и доп) – 356 с.
3. Коваленко В.В. Регламентация и автоматизация бизнес-процесса проведения диспансеризации населения // В.В. Коваленко, С.С. Пискунова, В.А. Миносян // В сборнике: Актуальные задачи математического моделирования и информационных технологий Материалы Международной научно-практической конференции. - 2018. - С. 83-85

УДК 004.051

УСКОРЕНИЕ ВЫПОЛНЕНИЯ ПРОГРАММНОГО КОДА С ИСПОЛЬЗОВАНИЕМ JUST IN TIME КОМПИЛЯТОРА В ЯЗЫКЕ PHP

О.Н. Усик

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

Аннотация. В работе рассматривается процесс преобразования кода, написанного на языке PHP, в машинный язык с вариантами ускорения этого процесса.

Ключевые слова: PHP, JIT, программирование, компиляторы.

ACCELERATE CODE EXECUTION USING THE JUST IN TIME COMPILER IN PHP

O.N. Usik

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

The summary. The paper considers the process of converting code written in PHP into machine language with options for accelerating this process.

Keywords: PHP, JIT, programming, compilers.

В современном мире существует огромное количество информационных интернет-сервисов, помогающих пользователям автоматически решать стоящие перед ними проблемы. Каждая система должна удовлетворять пользователя теми или иными качествами, необходимыми для удовлетворения его собственных потребностей.

Современному пользователю важны два основных качественных показателя сайта: дизайн и производительность [1]. Как правило, человек не задерживается на интернет-ресурсе, если его не устраивают цвета, шрифты, расположение элементов, или необходимые ему данные долго извлекаются или обрабатываются. Если первую относительно легко устранить, то с последней могут возникнуть некоторые проблемы.

Хотя производительность и зависит от многих факторов, начиная от мощности серверов и заканчивая грамотным написанием кода, рано или поздно можно достичь некоторого предела в аппаратных или программных решениях стоящей проблемы, поэтому его ускорение может дать некоторое преимущество.

Выполнение PHP-кода

PHP – это интерпретируемый язык программирования. Каждый раз, когда выполняется код PHP, будь то фрагмент или целое веб-приложение, он проходит через интерпретатор PHP. Его работа проста: интерпретировать код и выдать обработанный результат.

По подобной схеме работает каждый интерпретируемый язык. Хотя некоторые шаги могут и варьироваться, их общий принцип остается тем же.

В PHP это происходит следующим образом [2]:

1. Код PHP преобразуется в наборы ключевых слов, которые называются токенами (Tokens). Этот процесс позволяет интерпретатору понять, в какой части программы написан каждый фрагмент кода. Этот этап называется лексированием (Lexing) или токенизацией (Tokenizing).

2. Далее интерпретатор PHP анализирует эту коллекцию токенов с целью построения смыслового или абстрактного синтаксического дерева (Abstract Syntax Tree — AST). Оно генерируется путем синтаксического анализа или парсинга (Parsing). Это дерево представляет собой набор узлов, указывающих, какие операции должны быть выполнены. Например, «echo 1 + 2» означает «вывести результат операции сложения — 1 + 2».

3. Затем происходит генерация промежуточного представления (Intermediate Representation IR), которое в PHP называется операционным кодом (Opcode). Этот процесс называется компиляцией.

4. Далее происходит само выполнение кода. PHP имеет движок под названием Zend VM, который получает список опкодов и выполняет их, в результате чего программа завершается.

Возникает вопрос: какой смысл проходить все этапы каждый раз, когда код выполняется, если он может не меняться так часто? Для этого существует расширение OpCache.

Оно добавляет слой оперативного общего кэша для опкодов. Его основная задача — извлекать опкоды, недавно сгенерированные из AST, и кэшировать их, чтобы последующих выполнениях программы пропускать некоторые этапы интерпретации.

Упрощенная схема процесса интерпретации PHP и того же процесса с учетом расширения OpCache выглядит следующим образом (рис. 1):

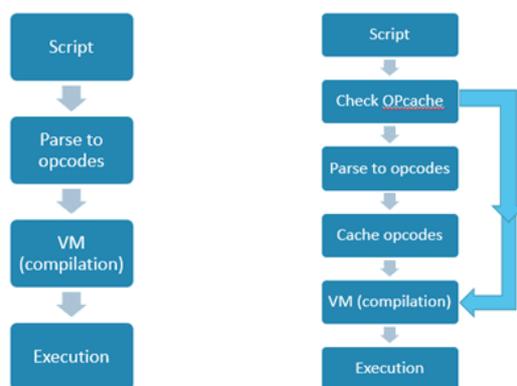


Рис. 1. Классический поток интерпретации PHP (слева) и с OpCache (справа)

Если OpCache позволяет быстрее получать операционный код, то JIT предназначен заставить его работать вообще без Zend VM.

JIT PHP компилирует только несколько опкодов, которые, по его мнению, того стоят. Для этого он анализирует коды операций, и определяет, какие из них имеет смысл компилировать.

Схему интерпретации PHP-кода с использованием JIT-компилятора можно представить следующим образом (рис. 2):

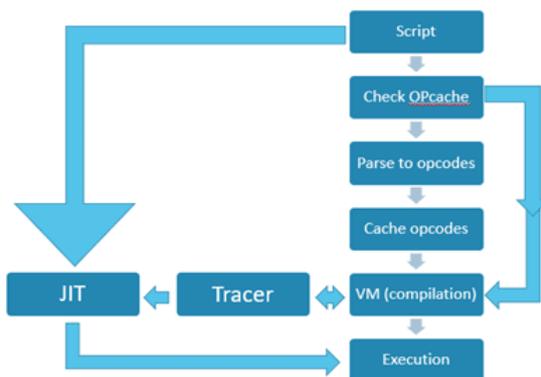


Рис. 2. Поток интерпретации PHP с JIT

Прирост скорости выполнения кода

В качестве примера были протестированы некоторые вычислительные функции, такие как заполнение матрицы, подсчет длины строки и т.п. Тестирование проходило в 4 этапа для каждой функции: без Opсache, с Opсache, с JIT в конфигурации function и с JIT в конфигурации tracing. Результат тестирования представлен на рисунке 3.

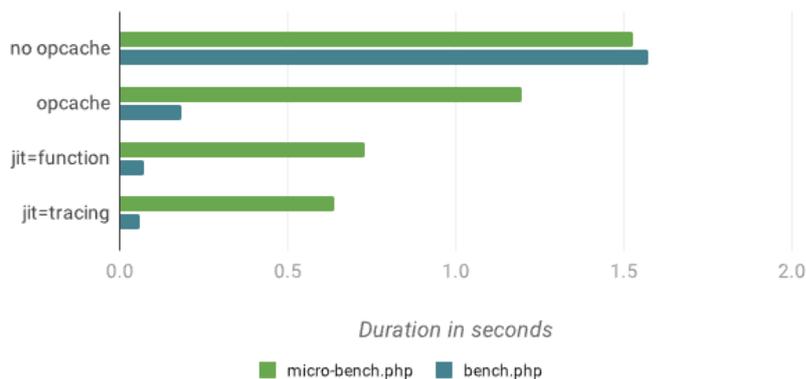


Рис. 3. Время выполнения функций на различных этапах

Второй тест был проведен с использованием функции Фибоначчи. Его результат представлен на рисунке 4.

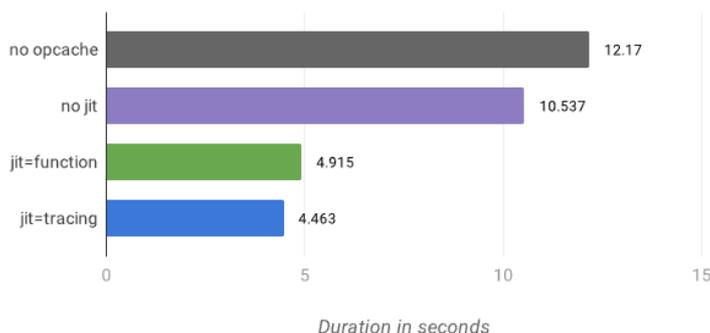


Рис. 4. Результат тестирования функции Фибоначчи

Результаты тестов позволяют утверждать, что Орсасхе действительно ускоряет выполнение кода, а использование JIT компилятора дополнительно ускоряет этот процесс. В результате чего скорость выполнения кода увеличивается в 2-15 раз в различных случаях.

Но стоит понимать, что использование JIT компилятора не сильно увеличит скорость работы многих php-приложений. Как правило, язык используется для операций ввода/вывода вида:

1. Получить данные HTTP-запроса;
2. Выполнить операции с базой данных;
3. Вернуть ответ пользователю.

В этом случае, основная часть задержек для получения итогового результата будет заключаться в операциях чтения или записи данных с диска или сетевыми задержками, в то время как JIT компилятор ускоряет работу со стороны процессора, например, при вычислительных операциях.

Пример увеличения производительности для различных PHP-приложений представлено на рисунке 5 [2].

PHP Benchmark Suite	3x Faster
WordPress	3-5% Faster
Framework-based Apps	No Difference

Рис. 5. Увеличение производительности приложений

Также, JIT дает преимущество, когда он разгружает собственные регистры и инструкции ЦП. PHP — это язык со слабой типизацией. Это затрудняет определение типа переменной и требует дополнительного анализа жизненного цикла переменной, поскольку тип переменной может измениться позднее в той же структуре кода. Строго типизированный код и функции со скалярными типами могут помочь использовать регистры ЦП и специальные инструкции, где это возможно. Например, включенная строгая типизация и чистая функция со строгими типами параметров и возвращаемых значений является лучшим вариантом:

```
declare(strict_types=1);
function sum(float $a, float $b): float {
    return $a + $b;
}
```

Таким образом, JIT-компиляция открывает возможности для переноса кода с различных языков на PHP благодаря увеличению производительности его отдельных аспектов, а также открывает новые возможности для языка в таких дисциплинах, как, например, машинное обучение.

Библиографический список

1. Федоркевич Е.В., Ветошев В.О. Отбор критериев оценки качества сайтов образовательных организаций. 2017, Том 5, №2
2. Exploring the New PHP JIT Compiler
URL: <https://www.zend.com/blog/exploring-new-php-jit-compiler> (дата обращения 12.02.2022)

УДК 004.67; ГРНТИ 50.41.25

РАЗРАБОТКА КОМПЛЕКСНОГО ПОДХОДА РЕАБИЛИТАЦИИ ПОСТКОВИДНЫХ СОСТОЯНИЙ СРЕДСТВАМИ АДАПТИВНОЙ ФИЗИЧЕСКОЙ КУЛЬТУРЫ И ИНТЕРНЕТ-КОММУНИКАЦИЙ

А.И. Козлов

*Рязанский государственный радиотехнический университет имени В.Ф. Уткина,
Российская Федерация, Рязань, contact@kozlov.tech*

Аннотация. В работе рассматривается перспективный вариант реализации комплексного подхода к реабилитации постковидных состояний используя информационную систему. Приводится пример архитектуры такой системы, ее достоинства и перспективы. Описываются технологии, позволяющие эффективно решить задачу.

Ключевые слова: клиент-серверная архитектура, реабилитация, интернет технологии, информационная система.

DEVELOPMENT OF AN INTEGRATED APPROACH TO THE REHABILITATION OF POST-COVID STATES BY MEANS OF ADAPTIVE PHYSICAL CULTURE AND INTERNET COMMUNICATIONS

A.I. Kozlov

*Ryazan State Radio Engineering University named after V.F. Utkin,
Russia, Ryazan, contact@kozlov.tech*

The summary. The paper considers a promising option for the implementation of an integrated approach to the rehabilitation of post-covid states using an information system. An example of the architecture of such a system, its advantages and prospects is given. The technologies allowing to solve the problem effectively are described.

Keywords: client-server architecture, rehabilitation, internet technologies, information system.

Мировая история не раз сталкивалась с пандемиями, вызванными вирусными заболеваниями. Эти пандемии наносят значительный ущерб экономической и общественной жизни как в мире, так и в нашей стране. Они замедляют экономический рост, оставляют людей без работы, приводят к нарушению производственных цепочек, дефициту товаров, заставляют людей обособливаться, сокращают социальные связи. Исследователи всего мира объединены проблемами, связанными с возникшей пандемией COVID-19. Поэтому актуальной задачей является борьба с этими эпидемиями. Долгое время борьба с такими пандемиями велась только силами человеческих ресурсов. Мировое сообщество находится в стадии поисков новых средств борьбы с пандемией. В 21-м веке появляется возможность вовлечь средства научно-технического прогресса в решение таких задач, что позволит снизить риск дополнительного распространения вирусов.

С вирусом необходимо бороться по профилактическому или лечебному направлениям. В рамках лечебного направления, после перенесённого вирусного заболевания пациента беспокоит слабость, утомляемость, нарушения сна, какие-либо другие состояния, не свойственные прежде [3]. В таком случае он нуждается в реабилитации для быстрого восстановления и возвращения к активной социальной жизни. Этот процесс является не быстрым и требует внимательности, сосредоточенности и дисциплинированности в первую очередь со стороны самого пациента. Для решения этой проблемы, помимо медицинских методов, возможно применение современных информационных технологий, таких как сетевые информационные системы.

Восстановление постковидных состояний приобретает сегодня особую актуальность, так как от этого зависит уровень качества жизни переболевших. Анализ литературных источников показал, что данные состояния имеют разнообразные осложнения, что требует определенных мер реабилитации. Оптимальным вариантов восстановления организма постковидных больных, по мнению многих врачей, является санаторно-курортное лечение и иные

рекреационные меры. Не все категории населения могут воспользоваться данными вариантами лечения. Сегодня быстрыми темпами развивается новое направление в медицине – телемедицина. Например, телемедицинские консультации, теле-реабилитации, чат-бот для организации системы дистанционного мониторинга и другие. Это направление позволяет использовать информационные технологии в сферах здравоохранении, в том числе в психологии здоровья, адаптивной физической культуре. Важным аспектом восстановления здоровья и постковидных состояний является реабилитация средствами адаптивной физической культуры с использованием информационных технологий. Использование IT-технологий позволит не только отслеживать состояние постковидных пациентов дистанционно, но и получать своевременную консультацию специалистов в области медицины и здравоохранения.

Для решения этой задачи предлагается разработать информационную систему для контроля пациентов, перенесших заболевание COVID-19, врачами. Такая система способна увеличить эффективность реабилитации постковидных состояний за счет повышения доступности взаимодействия врача и пациентов с помощью интернет-коммуникаций и использования рекомендательной системы на основе закономерностей предметной области, статистического анализа и искусственного интеллекта. Этот подход позволит обеспечить должный дистанционный контроль над состоянием конкретного пациента. Также снизится нагрузка конкретно на врачей и медицинский персонал в целом, что положительно скажется на том объеме пациентов, которые смогут получить квалифицированную медицинскую рекомендацию.

Архитектура

Предлагаемая система будет состоять из 5 основных компонентов: Telegram бота, web-сайта, блока логики взаимодействия, СУБД и рекомендательной системы.

Разрабатываемая информационная система основана на клиент-серверной архитектуре. В основе такой архитектуры лежит два компонента. Первый компонент – «сервер данных» (в дальнейшем сервер), а второй – «клиентское приложение» (в дальнейшем клиент). Архитектуру «клиент-сервер» делят на три класса: одно-, двух- и трёхуровневую [2]. Для реализации информационной системы выбрана трехуровневая архитектура (рис.1). В ней сервер баз данных представляет собой отдельный уровень, результаты работы которого использует сервер приложений, в котором реализуется логика данных и бизнес-логика приложения. Все обращения клиентов к базе данных происходят через промежуточное программное обеспечение (middleware). Вследствие этого, повышается гибкость работы и производительность. Кроме этого, преимуществами такого подхода являются: целостность данных, более высокая безопасность (по сравнению с двухуровневой архитектурой), защищенность базы данных от несанкционированного проникновения [1].

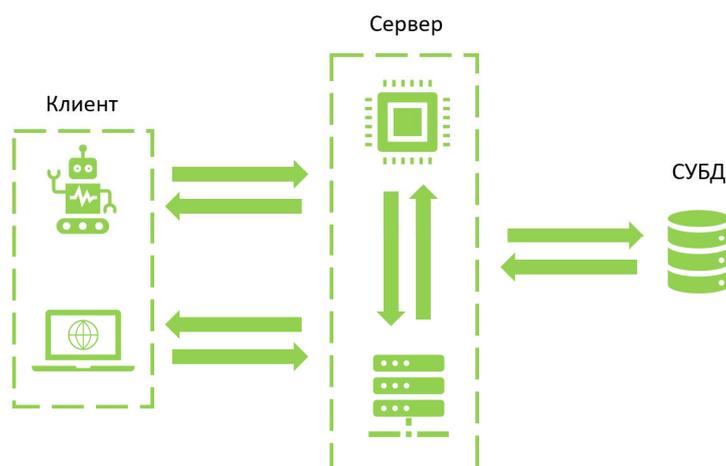


Рис. 1. Структура программного обеспечения

Основным языком создания информационной системы предлагается использовать Java. Такой выбор обусловлен рядом преимуществ: быстрота, высокий уровень защиты и надежность. Java позволяет создавать высоконагруженные системы Enterprise-уровня [4].

Для реализации серверной части предполагается использовать связку технологий с открытым исходным кодом Spring Framework + PostgreSQL. Spring это универсальный фреймворк, который будет отвечать за сервер приложений. Он позволит ускорить разработку и увеличит гибкость нашей системы. На его базе будет построен основной блок логики взаимодействия и реализована возможность добавления дополнительных модулей приложения. PostgreSQL это объектно-реляционная система управления базами данных, в нашем случае выступающая сервером данных, которая будет отвечать за хранение всех данных разрабатываемой информационной системы.

В информационной системе рекомендуется использовать рекомендательную систему. Такая система, на основе алгоритма взаимодействия, способна облегчить труд медицинскому персоналу, замечая различные закономерности. Генерация рекомендаций должна происходить на основе закономерностей предметной области, статического анализа и искусственного интеллекта. Для облегчения создания системы искусственного интеллекта может использоваться библиотека программ с открытым исходным кодом Deeplearning4j.

Разрабатываемая информационная система предусматривает наличие как локальной, так и глобальной сети, состоящей из вышеназванных узлов. Клиенты подключаются к серверу для решения различных задач. Предусматривается несколько типов клиентов. Первый клиент предназначен для медицинского персонала и представляет собой web-сайт. В основе его создания лежит технология Thymeleaf - современный серверный механизм Java-шаблонов для веб- и автономных сред, способный обрабатывать HTML, XML, JavaScript, CSS и даже простой текст. Вторым клиентом является Telegram бот, предназначенный для предоставления доступа к системе пациентам. Общение бота с сервером основано на REST (REpresentational State Transfer) API. Отличительной особенностью сервисов REST является то, что они позволяют наилучшим образом использовать протокол HTTP. В качестве пакета выбран JSON массив, который будет передаваться на конкретный URL, обрабатываемый сервером.

Главным преимуществом предлагаемой информационной системы можно считать удаленное хранение данных [5]. Это означает, что клиент сможет получить доступ к информации дистанционно из любой точки мира, где есть сеть Интернет. Кроме этого, появляется возможность создания ограниченного доступа к информации, так как клиенты не владеют

прямым доступом к базе данных. Еще одним преимуществом является независимость платформы клиента, так как общение с сервером происходит по заранее выбранному протоколу.

Основным потоком информации в системе является общение врача с пациентом (рис.2). Ключевой структурой этого общения является «сценарий». Он содержит в себе (все сразу или только часть) определенный набор вопросов и сообщение от врача. Кроме этого, врач указывает периодичность проведения сценария (на конкретную дату или же каждый день/неделю/месяц). Предполагается использование различных типов «вопросов». Для каждого из них предусмотрен определенный тип ответа: «да/нет» (логический тип данных), текст, целое число, число с плавающей запятой, дата. В сообщении врача могут содержаться информационные материалы, напоминания о необходимости проведения тренировки, информация о текущем состоянии пациента.

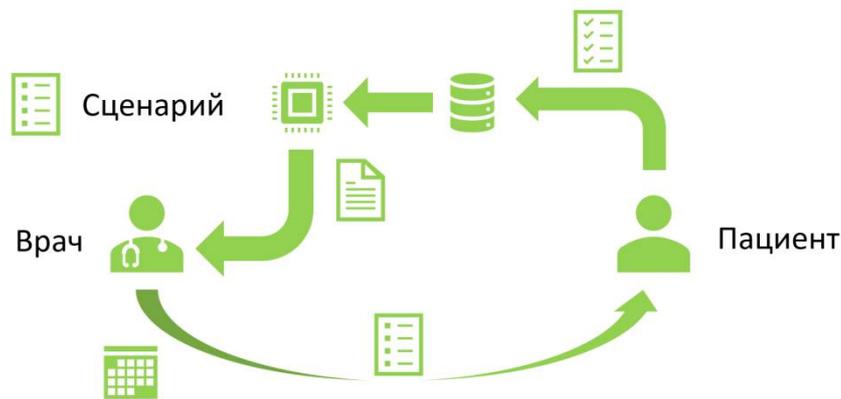


Рис. 2. Алгоритм взаимодействия врача с пациентом

В информационной системе реализуется разделение пользователей на различные роли: администратор, регистратор, врач, пациент. Роль администратора предоставляет возможность управления системы в целом: задание различных параметров системы, добавление пользователей различных ролей и изменение их учетных данных. Регистратор добавляет пациентов в систему и прикрепляет их к конкретному врачу. Врач обладает всеми правами регистратора и имеет возможность работы с прикрепленными к нему пациентами: создание сценария, просмотр состояния. Пациент является простым пользователем системы и не обладает правами изменения ее состояния.

Для каждой в роли в системе предусмотрен собственный интерфейс. Для администратора, регистратора и врача – различные панели управления на web-сайте, для пациента – Telegram бот.

Взаимодействие врача и пациента происходит посредством электронного диалога в форме «вопрос-ответ». Пациент получает сценарий через Telegram бота и в удобное для него время отвечает на вопросы. Также он может задать интересующий его вопрос врачу. Врачу, в свою очередь, поступает уведомление об изменении информации состояния пациента на web-сайте.

Заключение

Предлагаемая информационная система имеет высокую социальную значимость. Ее можно использовать в действующих клиниках для повышения эффективности работы врачей с пациентами. Другим направлением использования может стать сбор различной статистики. Такая статистика позволит решать различные прикладные задачи. Также эта статистика поможет в создании модели для научных исследований.

Предложенное решение является эффективным средством для борьбы с последствиями пандемии. Оно позволяет решить ряд актуальных проблем посредством использования

современных средств цифровизации. Развитие интернет-коммуникаций в сфере медицины это шаг к более высокому уровню жизни современного общества.

Библиографический список

1. Молчанов А.Ю. Системное программное обеспечение: Учебник для вузов. Санкт-Петербург, издательский дом «Питер», 2005.
2. Стасьшин В.М. Проектирование информационных систем и баз данных. Новосибирск, НГТУ, 2021.
3. Гайдаров Л.Ф., Леонкин В.В., Ситкалиева Е.В., Соколова М.В., Лазарева Г.Ю., Муллаярова Э.А. Справочник по реабилитации после заболеваний. Москва, ЭКСМО, 2007.
4. Машнин Т.С. Технология Web-сервисов платформы Java. Санкт-Петербург, БХВ-Петербург, 2012.
5. Бройдо В.Л. Вычислительные системы, сети и телекоммуникации. 3-е изд. Санкт-Петербург, издательский дом «Питер», 2008.

УДК 004.9; ГРНТИ 82.33.13

РАЗРАБОТКА BSC-СИСТЕМЫ НА ОСНОВЕ МЕТОДОЛОГИИ LEAN И ПРОЦЕССНОГО ПОДХОДА ДЛЯ ПРЕДПРИЯТИЯ ПО РАЗРАБОТКЕ ПРОГРАММНЫХ ПРОДУКТОВ

М.К. Бойко, В.В. Коваленко

*Сочинский государственный университет,
Россия, Сочи, vlvvas@mail.ru*

Аннотация. В статье представлена разработка BSC-системы на основе методологии Lean и процессного подхода для предприятий по созданию программного обеспечения, которая обеспечивает устранение “узких” мест в управлении. Качество управления оценивается на основе мониторинга и анализа значений показателей эффективности с использованием цикла Деминга.

Ключевые слова: бизнес-процесс, методология Lean, BSC-система, процессный подход, виды потерь, визуализация показателей, программное обеспечение.

BSC-SYSTEM DEVELOPMENT BASED ON LEAN-METHODOLOGY AND PROCESS APPROACH FOR SOFTWARE DEVELOPMENT ENTERPRISE

M.K. Boyko, V.V. Kovalenko

*Sochi State University,
Russia, Sochi, vlvvas@mail.ru*

The summary. The article presents the development of a BSC-system based on the Lean methodology and a process approach for software development enterprises, which ensures the elimination of bottlenecks in management. The quality of management assessed by monitoring and analyzing the values of performance indicators using the Deming cycle.

Keywords: business process, Lean-methodology, BSC-system, process approach, types of losses, visualization of indicators, software

В последнее время широко используется процессный подход в управлении предприятиями, который обеспечивает максимально производительное и результативное исполнение бизнес-процессов. При этом полезна его интеграция с методологией сбалансированной системы показателей - ССП (Balance Score Card – BSC), позволяющей реализовать схему перевода общей стратегии компании в термины операционного процесса, трансформируя таким образом стратегию компании в конкретные, вполне осязаемые задачи и показатели.

На определенном этапе развитие BSC-методологии стало сопровождаться появлением информационных систем с функциями информационной поддержки процессов управления на базе сбалансированных систем показателей. Эти системы представляют собой инструмен-

тальные средства для реализации методологии Balanced Score Card на основе процессного подхода и называются BSC-системами [1].

BSC-системы позволяют разрешить неожиданные проблемы, когда за пределами информационных систем оказываются бизнес-процессы, которые не определены однозначно и обычно в неявном виде присутствуют на стыках взаимодействия подразделений: кто-то вовремя не выполнил свою часть работ, где-то обнаружилась “бесхозная” функция, за которую никто не хочет брать ответственность, где-то сотрудник допустил ошибку, которая привела к серьёзным финансовым потерям. Разрешение этих “узких” мест в управлении выходят на уровень топ-менеджеров компании, который вынуждены тратить свое драгоценное время на “стыковку” разных мнений руководителей функциональных подразделений.

Обнаружение “узких” мест эффективно выполняется с помощью методологии “Бережливое производство (Lean)”, которая используется как инструмент сокращения снижения потерь времени и других ресурсов в рамках процессного подхода. Подобные проблемы авторами были обнаружены в компании, основной деятельностью которой является разработка программного обеспечения для информационной системы 1С, консультация и обучение пользователей по работе с ней, а также распространение лицензий на ее использование для предприятий. Учитывая специфику компании, использовалась версия методологии “Lean Software Development”, ориентированная именно на “узкие” места подобных предприятий [2].

В результате анализа деятельности компании на основе методологии “Lean Software Development” были выявлены следующие виды потерь: “Дефекты и брак”, “Излишние перемещения” и “Профессиональное выгорание сотрудников”. Для установления функциональности этих “узких” мест разработана стратегическая карта (рис. 1), которая кроме конкретизации целей определяет показатели и место расположения проблем в виде наименования перспектив [3, 4].

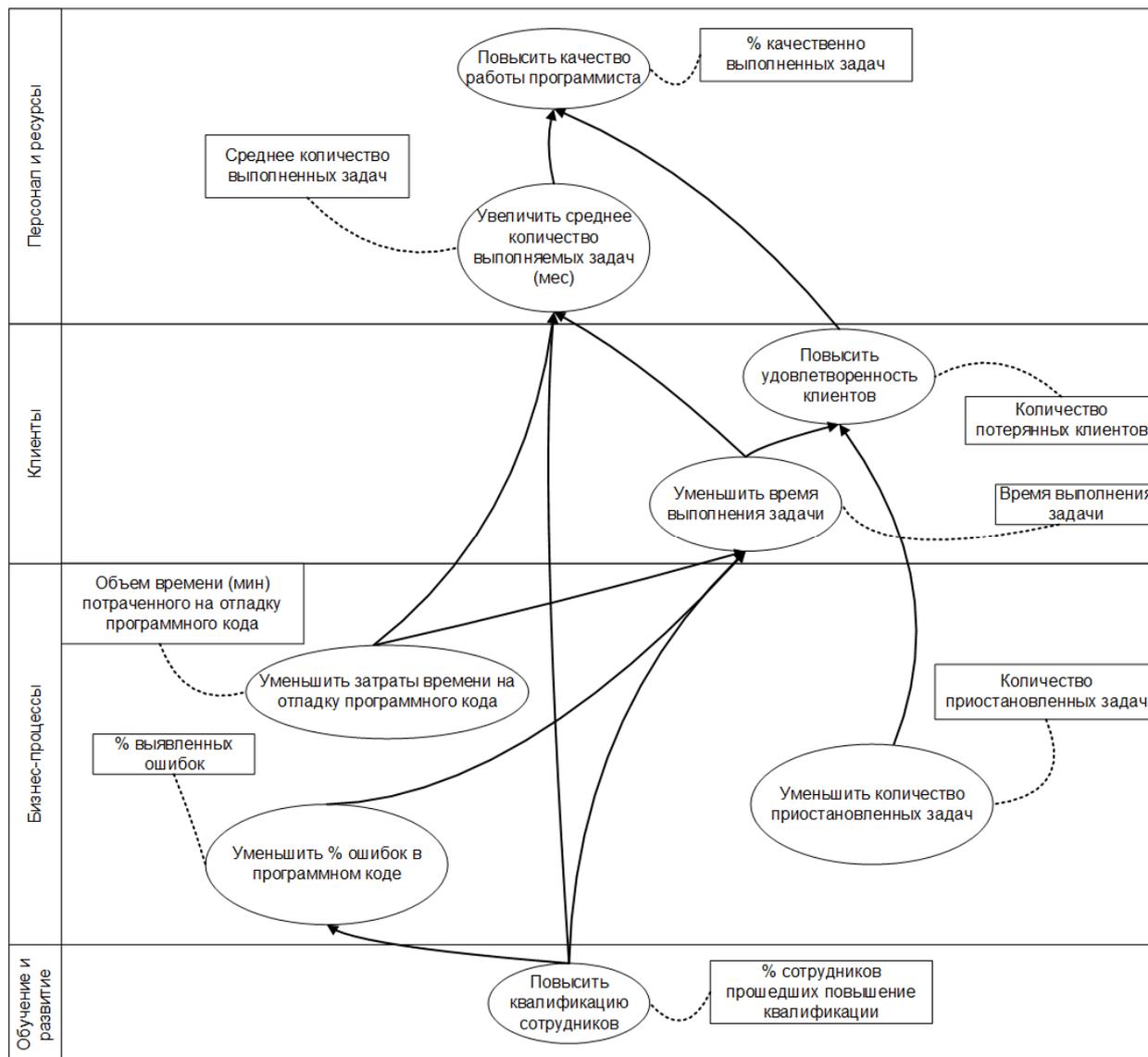


Рис. 1. Стратегическая карта для устранения “узких” мест в управлении

В качестве примера устранение этих видов потерь рассмотрены для бизнес-процесса “Поступление задачи программисту” (рис. 2).

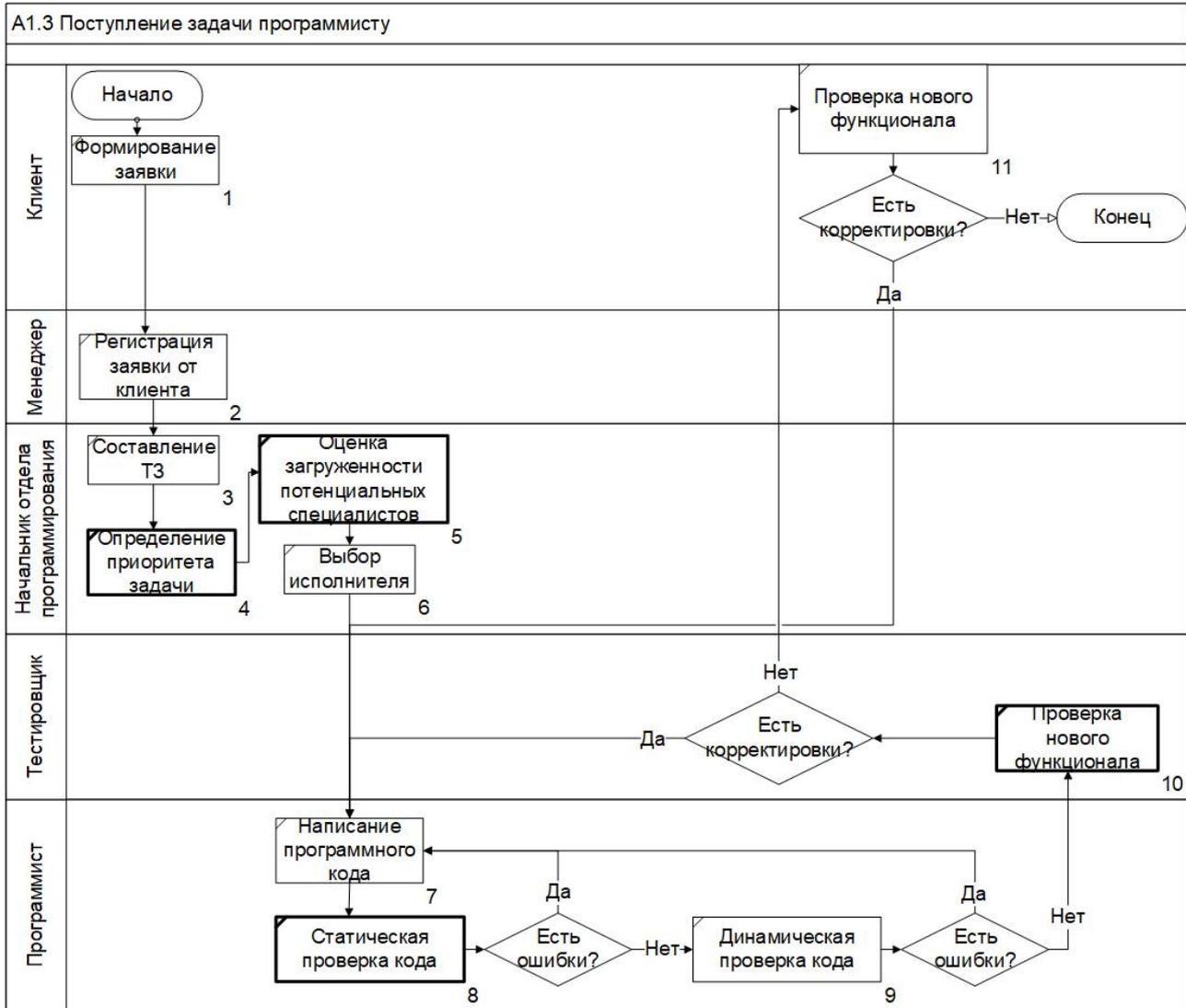


Рис. 2. Бизнес-процесс «Поступление задачи программисту».

Владельцем данного бизнес-процесса является директор компании, исполнителями – начальник отдела программирования и программист. Для устранения перечисленных видов потерь данный бизнес-процесс был модифицирован следующим образом.

Наиболее распространенными видами потерь являются “Излишние перемещения” и “Профессиональное выгорание сотрудников”. Как правило выбор исполнителя для выполнения задачи основывается на опыте специалиста в реализации данного функционала и на факте предыдущей работы программиста с данным клиентом. Если же задача является срочной, то у данного специалиста смещаются приоритеты задач. Он вынужден переходить на выполнение наиболее приоритетной задачи, временно останавливая выполнение текущей. Это приводит к увеличению интенсивности работы программиста и недовольству клиентов, поскольку выполнение задач для них затягивается.

Для исключения этих видов потерь для должности “Начальник отдела программирования” были введены дополнительные функции определения приоритетности задач и оценка загруженности потенциального программиста. (блоки 4 и 5 на рис. 2). Таким образом, перед постановкой задачи требуется оценка не только ее трудоемкости, но и уровень загруженности потенциальных исполнителей.

Виды потерь “Дефекты и брак” и “Неиспользуемый человеческий потенциал” появляются по той причине, что проверка работы нового функционала и его взаимодействия с ос-

тальной частью системы осуществляется самим программистом. Это приводит к тому, что программист тратит свой профессиональный потенциал на работу, не связанную с его прямыми обязанностями (написание программного кода). Для избавления от выделенных видов потерь предлагается ввести в организационную структуру предприятия новую должность “Тестировщик”.

Сотрудник на данной должности будет осуществлять проверку нового функционала, а также его взаимодействие с остальной частью информационной системы (стыки). Данные нововведения позволят снизить нагрузку программиста и уменьшить количество ошибок при тестировании нового функционала на стороне пользователя.

Вид потерь “Дефекты и брак” также присутствует и при написании самого программного кода. Развитие информационных технологий привело к тому, что почти каждая современная среда программирования имеет функционал по проверке синтаксиса написанного программного кода. Поэтому программисты, при выполнении поставленной задачи, целенаправленно не используют статическую проверку кода, отдавая предпочтение данному инструменту. Но даже высокие технологии не идеальны и могут не обнаруживать ошибки.

В результате программист тратит время на ожидание компиляции проекта и дальнейшую отработку нового функционала в скомпилированном виде. Такие ситуации особенно критичны, когда разработка происходит на стороне клиента с устаревшим техническим обеспечением и процесс компиляции может занимать до 10 минут. Для решения данной проблемы предлагается ввести обязательную статическую проверку кода (функциональный блок 8), это позволит более качественно производить отладку нового функционала и уменьшить количество ошибок при его тестировании на стороне пользователя, а также сократить время выполнения задачи.

Реализовав данный бизнес-процесс в российском пакете бизнес-моделирования “Business Studio” и добавив сведения о закреплении его функций и показателей за исполнителями, появляется возможность автоматически сгенерировать регламентные документы и отслеживать динамику изменения значений показателей [5]. Визуализация плановых и фактических значений показателей выполняется в виде индикаторных линеек, одна из которых реализована для показателя “Процент выявленных ошибок” (рис. 3).

Статус	Период	План	Факт	Индикатор
	Май 2021	10	40	
	Июнь 2021	10	30	
	Июль 2021	10	20	
	Август 2021	10	10	

Рис. 3. Индикаторная линейка для показателя “Процент выявленных ошибок”

Индикаторные линейки позволяют определить текущую оценку достижения цели. Линейка разделена на три зоны: зеленая, желтая, красная. Фактическое значение показателя в зеленой зоне означает его близость к плановому, в желтой зоне требует внимания ответственных лиц, а фактическое значение показателя из красной зоны предполагает проведение немедленных корректирующих действий.

Для поддержания значений показателей бизнес-процесса, указанных в стратегической карте, на желаемом уровне удобно использовать цикл Деминга, который реализует непрерывный процесс совершенствования управления деятельностью предприятием.

Библиографический список

1. Коваленко В.В. Разработка BSC-системы для фонда социального страхования на базе процессного подхода // В.В. Коваленко, А.О. Холодов // В сборнике: Актуальные проблемы современной науки и производства. Материалы V Всероссийской научно-технической конференции. Рязанский государственный радиотехнический университет имени В.Ф. Уткина. Рязань, ноябрь, 2020. - С. 209-214.
2. Поппендик Т. Бережливое производство программного обеспечения / Т. Поппендик, М. Поппендик / - М.: Вильямс – 2010. – 257 с.
3. Kovalenko V.V. Automation of quality management system on basis of modern technologies / V.V. Kovalenko, V.V. Kovalenko // European Journal of Computer Science. - 2015. № 1 (1). С. 41-47.
4. Коваленко В.В. Проектирование информационных систем / В.В. Коваленко. - М.: ИНФРА-М, 2021. (2-е изд., испр. и доп) – 356 с.
5. Коваленко В.В. Регламентация и автоматизация бизнес-процесса проведения диспансеризации населения // В.В. Коваленко, С.С. Пискунова, В.А. Миносян // В сборнике: Актуальные задачи математического моделирования и информационных технологий. Материалы Международной научно-практической конференции. Сочинский государственный университет. Сочи, 2018. - С. 83-85.

УДК: 004.041; ГРНТИ 20.23.17

АСПЕКТЫ ИНТЕЛЛЕКТУАЛЬНОГО АНАЛИЗА ДАННЫХ В БИЗНЕСЕ

Аюб Себах

*Рязанский государственный радиотехнический университет имени В.Ф. Уткина,
Российская Федерация, Рязань, ayoubsebah5@gmail.com*

Аннотация. В работе анализируется специфика машинного обучения как одного из средств интеллектуального анализа данных для разных направлений в бизнесе. приведены основные особенности, преимущества использования и возможности
Ключевые слова: Data-Mining, бизнес-решений.

ASPECTS OF DATA MINING IN BUSINESS

Ayoub Sebah

*Ryazan State Radio Engineering University named after V.F. Utkin,
Russia, Ryazan, ayoubsebah5@gmail.com*

The summary. The paper analyzes the specifics of machine learning as one of the data mining tools for different areas of business. the main features, advantages of use and possibilities are given

Key words: Data-Mining, business solutions.

В настоящее время в различных сферах науки и бизнеса наблюдается активная тенденция внедрения интеллектуальных систем, технологий, методов и средств для повышения эффективности трудовой деятельности людей.

Во многом это обусловлено тем, что в информационной среде Internet наблюдается стремительный рост объемов разнородных данных, обусловленный развитием и распространением социальных сетей, интернет-магазинов, тематических блогов и информационных веб-систем [1]. Даная динамика существенно отражается на активности развития разных направлений электронной коммерции, сферы услуг и торговли различной электронной продукцией [2].

Анализируя все перечисленные процессы, следует отметить актуальность использования современных интеллектуальных методов и технологий анализа больших данных (ИАД), в частности, машинного обучения (МО) для решения задач бизнеса.

МО акцентируется на процессах построения и оценки качества математических моделей, предназначенных для задач комплексного исследования данных различных типов и структур. Задачи обучения выражаются в подборе и настройке параметров созданных моделей, которые могут меняться динамически для адаптации качества отражения и визуализации данных. Фактически, создаваемая структура получает возможность приобретения элементов искусственного интеллекта, обучаясь на подаваемых ей данных. По завершению процессов обучения созданных моделей на имеющихся данных результирующие модели можно применять для прогнозирования, классификации и формализации различных аспектов новых данных [3].

МО оперирует объектами, представляющими собой элементарные единицы данных в предметных задачах, характеризуемые наблюдаемыми (явными) переменными x , а также не явными (скрытыми) переменными t , которые могут принимать возможные значения из ряда известных множеств.

Основная задача применения МО в различных бизнес-процессах:

- Данные. Собираются всевозможными способами. Чем больше данных, тем эффективней машинное обучение и точнее будущий результат.
- Признаки. Определяют, на каких параметрах строится машинное обучение.
- Алгоритм. Выбор метода машинного обучения (при условии наличия хороших данных) будет влиять на точность, скорость работы и размер готовой модели.
- Инновационность. возможности ML открывают новые перспективы развития и роста практически всех отраслей экономики.
- Специфичность. машинное обучение применяется для внедрения и разработки новых продуктов исключительно людьми, которые разбираются в IT-технологиях.
- Простота. продукты реализуемые с использованием технологий машинного обучения, становятся понятны даже школьникам и людям преклонного возраста. [4].

Типы машинного обучения принято разделять на три категории (рис. 1):

- обучение с учителем (supervised learning);
- обучение без учителя (unsupervised learning);
- обучение с подкреплением (reinforcement learning).



Рис. 1. Типы машинного обучения

Одной из главных задач применения МО бизнеса является автоматическая идентификация активных взаимозависимостей между явными и скрытыми переменными объекта исследования.

Это требуется для осуществления оценки или прогнозирования возможных значений скрытых компонент заданного объекта по его явным компонентам [5].

Целью данной работы является анализ ключевых возможностей и аспектов интеллектуального анализа данных в различных сферах бизнеса.

В настоящее время разработано большое количество алгоритмов МО, используемых для проведения ИАД в различных сферах бизнеса, к наиболее популярным из которых относят: регрессионные модели, алгоритм к-ближайших соседей, решающие деревья, машины опорных векторов, искусственные нейронные сети [6].

Результаты сравнительного анализа обозначенных выше методов МО приведены в таблице 1.

Таблица 1. Результаты анализа методов МО

Метод	Сравнение методов			
	Длительность обучения	Наглядность	Эффективность обработки данных	Масштабируемость
К-ближайших соседей	низкая	высокая	низкая	средняя
Машины опорных векторов	низкая	средняя	средняя	низкая
Решающие деревья	средняя	высокая	низкая	низкая
Регрессионные модели	средняя	высокая	средняя	низкая
Искусственные нейросети	средняя	средняя	высокая	средняя

Рассмотренные методы МО могут эффективно применяться в различных сферах бизнеса для ИАД, в частности [7]: розничная торговля (с помощью собранной информации и данных о продажах выполняется прогноз спроса и поставки товара конечным пользователям), финансовые услуги (анализа кредитоспособность заемщика, задачи кредитного скоринга и андеррайтинга), телекоммуникации (удержание существующих клиентов мобильной связи и привлечение новых путем их сегментации и выявления социальной принадлежности абонентов), сфера медицинских услуг (диагностика и прогнозирование заболеваний, подбор эффективных средств лечения и диет).

Выводы

В результате проведенного исследования ключевых аспектов использования интеллектуального анализа данных в бизнесе установлены наиболее популярные и актуальные алгоритмы машинного обучения, определены сферы их практического использования.

Библиографический список

1. Тулеубаева М. К. Особенности и перспективы развития электронного бизнеса / Большая Евразия: Развитие, безопасность, сотрудничество. – 2019. – №2. – С. 305-307.
2. Афанасьева Т.В. Введение в проектирование систем интеллектуального анализа данных. - Ульяновск: Ульяновский государственный технический университет (УлГТУ), 2017. — 64 с.

3. Замятин А.В. Интеллектуальный анализ данных. – Томск: Издательский Дом Томского государственного университета, 2016. – 120 с.
4. Наместников А.М. Интеллектуальные репозитории технической документации в проектировании автоматизированных систем. – 2018 – С.67-74.
5. Рудниченко Н.Д. Разработка программного приложения решения задач классификации на основе использования методов машинного обучения / Материалы IX Международной научно-практической конференции «Информационные управляющие системы и технологии». - 2020. - С.307-310.
6. Храмов А.Г. Методы и алгоритмы интеллектуального анализа данных. – Самара: СНИУ им. академика С.П. Королева, 2019. – 176 с.
7. Пальмов С.В. Интеллектуальный анализ данных. – Самара: Изд-во ПГУТИ, 2017. – 128 с.

УДК 004.032.26; ГРНТИ 50.47.02

РАЗРАБОТКА НЕЙРОСЕТЕВОГО АППРОКСИМАТОРА С ИСПОЛЬЗОВАНИЕМ ЯЗЫКА PYTHON

В.С. Анашкин, С.В. Челебаев

*Рязанский государственный радиотехнический университет имени В.Ф. Уткина,
Российская Федерация, Рязань*

Аннотация. В работе рассматривается решение задачи линеаризации тарировочной характеристики различных датчиков. Для решения задачи было разработано программное средство для обучения нейронных сетей произвольной топологии, воспроизводящих любую заданную непрерывную математическую функцию любого числа переменных.

Ключевые слова: нейронные сети, аппроксимация, линеаризация.

DEVELOPMENT OF A NEURAL APPROXIMATOR WITH PYTHON

V.S. Anashkin, S.V. Chelebaev

*Ryazan State Radio Engineering University named after V.F. Utkin,
Russian Federation, Ryazan*

Abstract. The paper considers the solution of the problem of linearization of the calibration characteristics of various sensors. To solve the problem, a software tool was developed for training neural networks of arbitrary topology reproducing any given continuous mathematical function of any number of variables.

Keywords: neural networks, approximation, linearization.

Нелинейность тарировочной характеристики

Исходными данными для информационных систем, обеспечивающих автоматизированное управление технологическими процессами, являются измерения множества датчиков, отслеживающих различные показатели объектов, так или иначе участвующих в техпроцессах. Точность измерений датчиков является ключом к эффективному управлению.

Датчики преобразуют входную (обычно неэлектрическую) величину в выходную (обычно электрическую). Такая зависимость между входом и выходом датчика называется тарировочной (градуировочной) характеристикой. У идеального датчика она линейна. У реальных же датчиков эта зависимость отклоняется от линейной, что снижает их точность. Для нивелирования этого недостатка возможна линеаризация тарировочной характеристики датчика.

Пусть имеется некоторая нелинейная зависимость

$$F(x) = x^2, x \in [0; 1]. \quad (1)$$

Тогда для её линеаризации необходимо подобрать обратную функцию

$$G(x) = F^{-1}(x) = \sqrt{x}. \quad (2)$$

После применения обратной функции к исходной будет получена искомая линейная зависимость

$$G(F(x)) = \sqrt{F(x)} = \sqrt{x^2} = x. \quad (3)$$

Таким образом, для линеаризации тарифовочной характеристики необходимо создать устройство, воспроизводящее необходимую зависимость (обратная для тарифовочной характеристики).

Нейросетевая аппроксимация

Получение аналитического выражения обратной функции для реальных тарифовочных характеристик может быть непростой задачей. Возможным решением может быть аппроксимация (восстановление) обратной функции некоторому конечному набору точек. Такой набор может быть получен по отсчётам известной тарифовочной характеристике и нужной линейной зависимости. Точность восстановления обычно оценивается как максимальная абсолютная ошибка (англ. maximum absolute error, MAE), то есть наибольшее отклонение аппроксимированного значения от истинного.

Существует множество «классических» способов восстановления зависимости по набору точек: полиномиальная регрессия, сплайновая интерполяция, кусочно-линейная аппроксимация и другие. Каждый из них имеет как свои преимущества, так и недостатки – увеличение требований к точности воспроизведения зависимости или увеличение числа входных переменных приводит к значительному изменению модели, усложнению вычислений или аппаратной реализации.

Указанных недостатков лишена аппроксимация при помощи нейронных сетей. Нейронная сеть прямого распространения (без обратных связей) даже с одним скрытым слоем может воспроизводить любую непрерывной зависимости любого числа переменных. Количество входов сети соответствует числу переменных, выход сети – один. Количество и размеры скрытых слоёв могут быть произвольными – оптимальная конфигурация может быть получена только эмпирическим путём [1].

В контексте решаемой задачи обучение нейросети является обучением с учителем, где контрольными данными являются наборы значений входных переменных и соответствующие им значения функции, обратной тарифовочной характеристике. Ввиду вариативности топологии сети наиболее гибким и эффективным способом её обучения является метод обратного распространения ошибки.

Нейросетевая аппроксимация обладает рядом преимуществ перед другими методами:

- **гибкость** – нейронные сети могут менять структуру для воспроизведения различных зависимостей без изменения общего алгоритма их работы и обучения;
- **масштабирование** – нейронные сети позволяют наращивать точность приближения за счёт увеличения времени обучения (т.е. увеличения количества итеративных вычислений);
- **простота реализации** – сложность аппаратной реализации нейросетевого устройства-аппроксиматора прямо пропорционально не зависит от требуемой точности приближения.

Однако, нейросетевая аппроксимация не лишена недостатков, как общих при обучении выбранным методом, так и специфических, связанных с решаемой задачей:

- **случайность** – исходные параметры сети инициализируются случайным образом, поэтому для получения наилучших результатов необходимо проведение серии экспериментов;
- **проблемы градиентного спуска** – обучение методом обратного распространения приобретает недостатки градиентного спуска, который лежит в основе метода (например, «застывание» в локальных минимумах), для борьбы с которыми необходимо подстраивать скорость обучения;

● **снижение средней ошибки** – цель обучения состоит в минимизации среднеквадратичной ошибки (англ. mean squared error, MSE), а при оценке точности аппроксимации используется максимальная ошибка.

Цель работы, постановка задачи

Целью работы является создание нейросетевого аппроксиматора, необходимого для линеаризации сигналов с различных датчиков. Для достижения цели необходимо программное средство, позволяющее через интерактивный графический интерфейс:

- задавать исходные данные в виде набора точек или математической зависимости от любого количества переменных;
- задавать настройки обучения (продолжительность, скорость и другие);
- задавать топологию сети (количество слоёв, их размеры и функции активации);
- получать результаты обучения (достигнутая точность, параметры модели).

Данные требования обусловлены спецификой решаемой задачи. Существующие продукты не удовлетворяют всем требованиям – в некоторых из них затруднен ввод исходных данных, а в других невозможно получение параметров модели.

Таким образом, для достижения цели необходимо разработать программное средство, удовлетворяющее определённым выше требованиям. Для проверки работы средства необходимо выполнить серию экспериментов по обучению нейросетей для воспроизведения некоторых зависимостей.

Разработка архитектуры программного средства

На основе анализа требований было выявлено два независимых слоя разрабатываемого приложения: слой обучения нейронных сетей (сервис), слой графического пользовательского интерфейса (клиент). Каждый слой решает разные задачи независимо друг от друга, может быть реализован на разных языках и исполняться на разных машинах. Характер взаимодействия между двумя частями приложения представлен на рисунке 1 в виде диаграммы последовательностей UML.

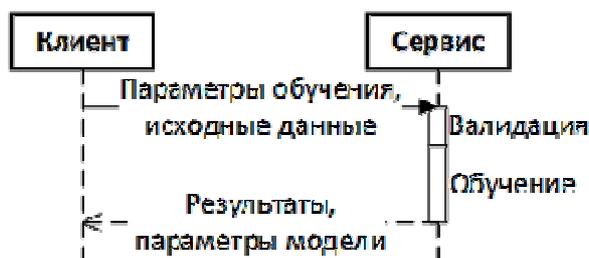


Рис. 3. Последовательность взаимодействия частей разрабатываемого программного средства

Параметры обучения представляют собой множество вложенных пар «ключ: значение». Их структурированное представление возможно с помощью текстового формата JSON – он поддерживает все базовые типы данных, обладает иерархичностью, сообщения в этом формате могут быть прочитаны человеком и переданы по сети.

Графический интерфейс должен поддерживать функции, представленные в виде требований при описании цели работы. Для этого пользовательский интерфейс должен содержать компоненты для ввода или выбора значений различных параметров, а также просмотра текстовых и графических результатов обучения.

Для организации эффективного и гибкого обучения нейронных сетей необходимо разработать объектную модель для представления взаимосвязанных сущностей, участвующих в обучении. В ходе анализа процесса обучения нейронных сетей были выявлены сущности, представленные в таблице 1.

Таблица 1 – Основные сущности части приложения, ответственной за обучение нейронных сетей

Сущность	Предок	Назначение
Операция		Объявление общих методов (прямой и обратный проход по сети)
Операция с параметром	Операция	Объявление методов для операций с параметром
Взвешивание	Операция с параметром	Реализация перемножения входов на весовые коэффициенты
Смещение	Операция с параметром	Добавление смещения к взвешенной сумме
Функции активации (сигмоида, ReLU и другие)	Операция	Различные функции активации
Слой		Набор операций, которые выполняются в рамках слоя
Нейросеть		Набор слоёв, через которые проходят исходные данные
Потеря		Оценка точности нейросети (MSE)
Оптимизатор		Задание правил корректировки параметров модели (стохастический градиентный спуск)
Тренер		Запуск обучения, агрегирование других сущностей с целью предоставления единой точки для начала обучения

Выявленные сущности обладают определёнными атрибутами и взаимодействуют друг с другом посредством вызовов некоторых методов. Диаграмма классов UML, отражающая взаимосвязи выявленных сущностей, представлена на рисунке 2. Во избежание перегрузки диаграммы, на ней показаны не все атрибуты и методы, а самые основные.

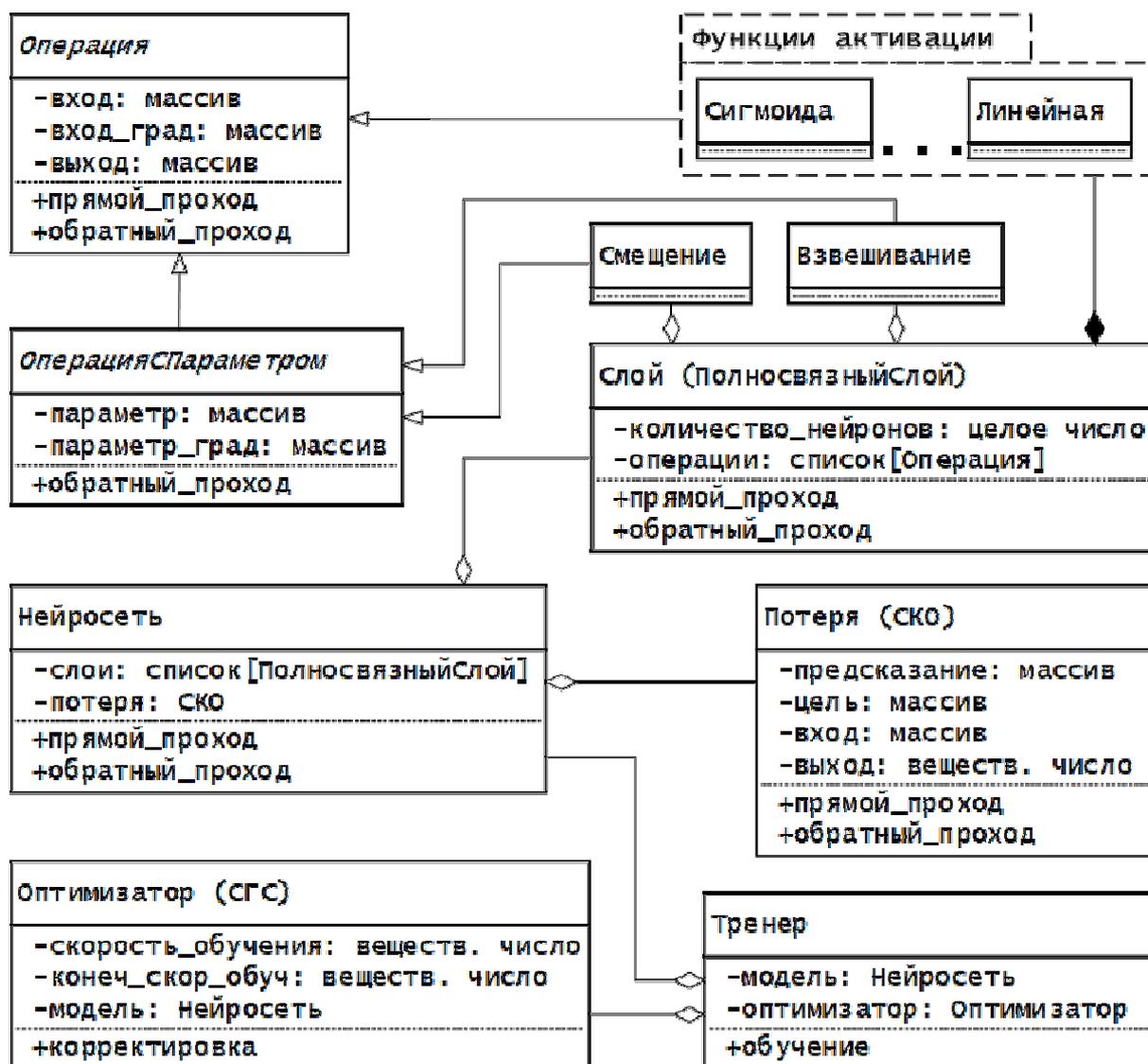


Рис. 4. Объектная модель части приложения, ответственной за обучение нейронных сетей

Разработка алгоритма обучения

Представление результатов разработки всех алгоритмов работы каждой части приложения выходит за рамки данной работы. На рисунке 3 представлена укрупнённая блок-схема алгоритма обучения нейронных сетей в рамках разрабатываемого программного средства.

При разработке алгоритма были реализованы некоторые приёмы повышения эффективности обучения [2]:

- данные были разделены на данные для обучения и данные для тестов – это позволяет объективно оценивать результаты работы сети, т.к. при тестировании нейросеть обрабатывает данные, не встречаемые при обучении;
- при обучении данные разбиваются на пакеты фиксированного размера, что позволяет выполнять эффективное по времени и объёму используемой памяти обучение на больших выборках;
- перед обучением пакеты перемешиваются, что снижает риск переобучения нейронной сети.



Рис. 5. Алгоритм обучения

4. Детализация блока «1. Формирование обучающей выборки» представлена на рисунке 4.

Данный алгоритм учитывает размерность заданной функции. Например, при диапазоне изменения $[0; 1]$ и количестве отсчётов, равном 3, для одной переменной вектор отсчётов будет иметь вид:

$$(0 \quad 0.5 \quad 1).$$

Если же количество входных переменных будет равно 2, то при тех же условиях массив входных переменных будет иметь вид:

$$\begin{pmatrix} 0 & 0 & 0 & 0.5 & 0.5 & 0.5 & 1 & 1 & 1 \\ 0 & 0.5 & 1 & 0 & 0.5 & 1 & 0 & 0.5 & 1 \end{pmatrix}.$$

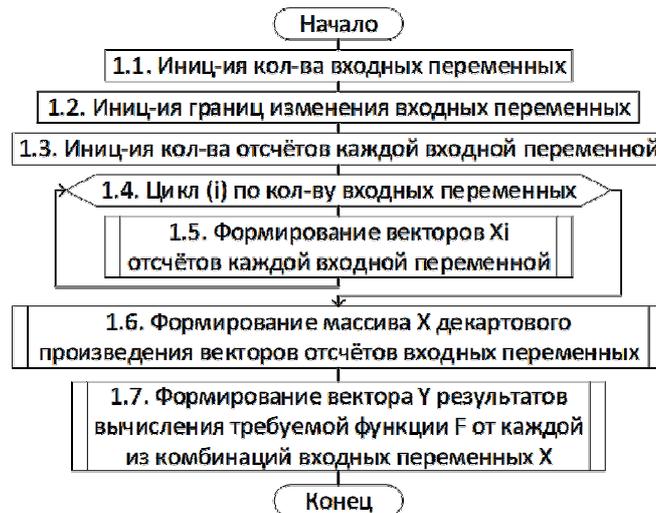


Рис. 6. Алгоритм формирования обучающей выборки

Блок «2. Разбитие обучающей выборки» осуществляет случайное разделение обучающей выборки на данные для обучения и данные для тестов в некотором задаваемом соотношении.

Блок «5. Формирование пакетов» осуществляет разбиение всех данных для обучения на набор пакетов фиксированного размера.

Блок «7. Перемешивание пакета» осуществляет случайное перемешивание данных в пакете, что повышает обучаемость нейросети.

Блоки 8-11 представляют собой суть обучения. «8. Прямой проход» формирует выход нейронной сети. «9. Оценка потерь» сравнивает полученный выход с эталонным и формирует ошибку. «10. Обратный проход» проносит полученную ошибку через последовательность слоёв нейронной сети в обратном направлении, формируя градиенты для коррекции параметров. «11. Коррекция весов и смещений» корректирует параметры нейросети в соответствии с вычисленными на предыдущем шаге градиентами.

Блоки 12-13 выполняют оценку точности работы нейронной сети на основе тестовых данных.

Разработка программного средства

Для разработки был выбран язык программирования Python. При выборе ключевыми оказались следующие преимущества данного языка: поддержка со стороны сообщества (множество библиотек для эффективного решения многих типовых задач), удобное сочетание различных парадигм программирования (функциональное, структурное, объектно-ориентированное), читабельность и низкий порог входа.

Для реализации отдельных функций в рамках программного средства были использованы следующие библиотеки:

- **NumPy** – библиотека, написанные на языке C++, что обеспечивает высокую производительность всех операции, связанных с матрицами (создание, выполнение вычислений, изменение размеров и другое);

- **PyQt5** – библиотека, основанная на фреймворке для C++ Qt, для создания кросс-платформенных оконных GUI-приложений, поддерживающих систему сигналов и слотов (аналог событий и функций-обработчиков);

- **SymPy** – библиотека для символьных вычислений, позволяющая преобразовать написанные текстом функции в лямбда-выражения языка Python;

- **matplotlib** – библиотека для построения различного рода графиков с интерфейсом, схожим с пакетом MatLab.

Рассмотрение подробностей программной реализации выходит за рамки данной работы. Ключевые особенности разработанного программного средства:

- интерактивный GUI для задания параметров сети и настроек обучения, просмотра графических и текстовых результатов обучения;

- применение «best practices» как в области обучения нейросетей (матрицы, перешивание выборки, выделение тестовых данных и др.), так и в области программирования (ООП, отделение изменяемых параметров от неизменяемой логики, модульность);

- каждая попытка обучения запускается в отдельном потоке, что позволяет ускорить обучение и повысить эффективность использования аппаратного обеспечения за счёт включения в работу всех доступных ядер CPU.

Интерфейс разработанного приложения состоит из одного окна и трёх вкладок для задания настроек обучения, просмотра текстовых результатов обучения, просмотра графиков. Вкладка для задания настроек обучения разделена на несколько панелей. Панель «Выборка» позволяет задать: размерность воспроизводимой функции, саму функцию, границы измене-

ния входных переменных, долю данных, которые будут зарезервированы для тестов. Панель «Обучение» позволять задать: начальную скорость обучения, конечную скорость обучения, способ снижения скорости обучения, количество эпох обучения, размер пакета, необходимость вывода промежуточных результатов, необходимость построения графиков, количество попыток обучения (в результате будет сохранена лучшая попытка). Панель «Сеть» позволяет задать количество слоёв, а для каждого слоя задать: количество нейронов, функцию активации.

После завершения обучения программное средство предоставляет возможность экспорта параметров последней обученной модели в текстовом виде. При этом выводится следующая информация: воспроизводимая зависимость, границы изменения входных, точность воспроизведения (максимальная абсолютная ошибка, максимальная относительная ошибка, средняя абсолютная ошибка, среднеквадратичная ошибка), параметры слоёв (размер, функция активации, веса, смещения).

Тестирование разработанного программного средства

Для проверки работы разработанного средства было выполнено несколько попыток обучения при различных комбинациях исходных параметрах. Результаты представлены в таблице 2.

Таблица 2 – Результаты обучения

	1	2	3	4
Зависимость	$\sqrt{x_1}$	$x_1 * \sin(\pi * x_2)$	$\exp(x_1)$	$\log[3](x_1)$
Входные переменные	$0 < x_1 < 1$	$1 < x_1 < 2, 0 < x_2 < 1$	$-1 < x_1 < 1$	$0.1 < x_1 < 3$
Размер выборки	4096	$64 * 64$	2048	1024
Количество эпох обучения	5000	15000	10000	7500
Размер слоя	10	14	8	12
Функция активации	гиперб. танг.	сигмоид	сигмоид	гиперб. танг.
MAE	7.204730e-03	5.775101e-03	2.143905e-03	2.511307e-02
MSE	3.822011e-06	6.573901e-06	4.843823e-07	2.400190e-05
Макс. относ. ошибка, %	7.204730e-01	2.887550e-01	9.121438e-02	8.111719e-01

В качестве примера на рисунке 5 приведены графики, формируемые в после обучения нейронной сети на воспроизведение функции одной переменной.

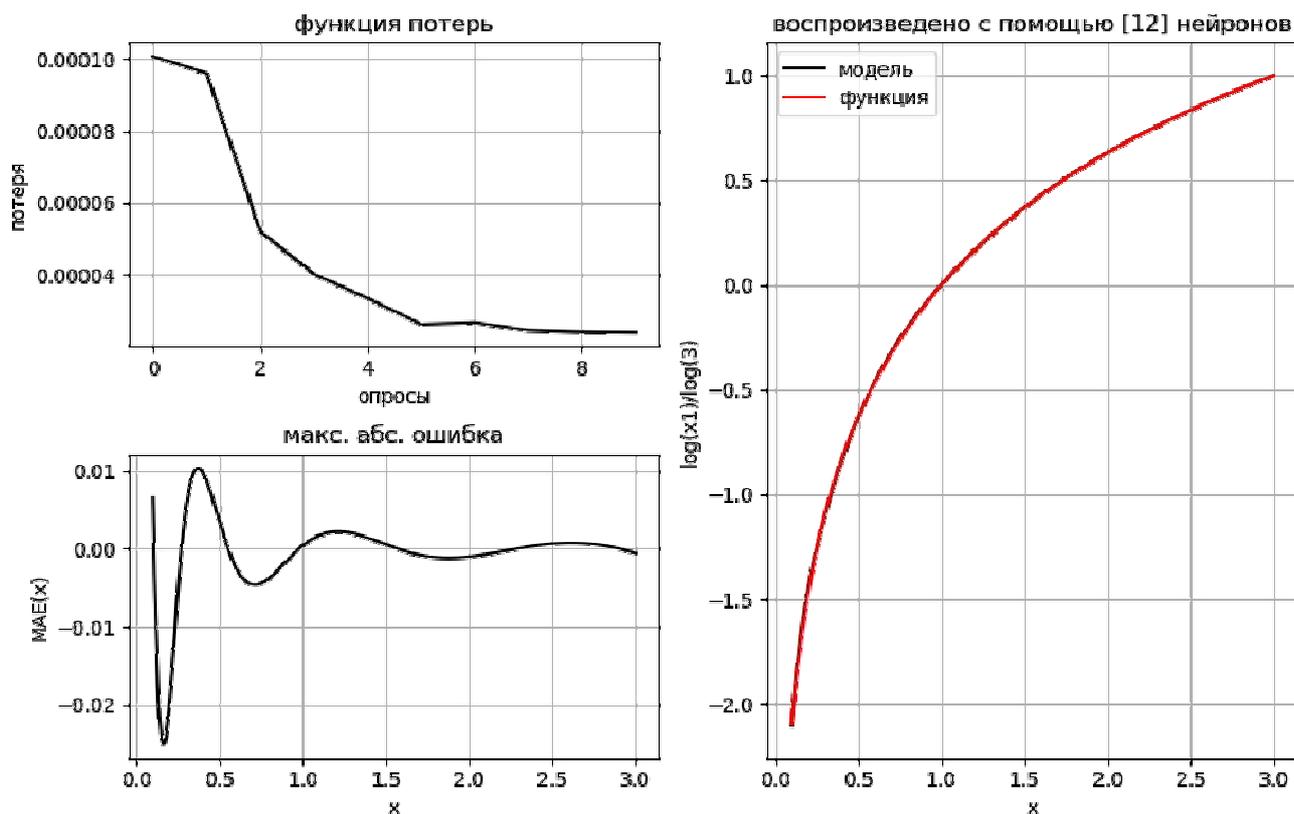


Рис. 7. Графики, формируемые разработанным программным средством

В результате выполнения работы было разработано и протестировано программное средство для обучения нейронных сетей произвольной топологии для воспроизведения произвольных непрерывных функций любого числа переменных.

Библиографический список

1. Локтюхин В.Н., Челебаев С.В., Антоненко А.В. Нейросетевые аналого-цифровые преобразователи. — М.: Горячая линия–Телеком. — 2010. — 128 с.
2. Комарцова Л.Г. Нейрокомпьютеры: учеб. Пособие для вузов. — М.: изд-во МГТУ им. Н.Э. Баумана, 2002. 320 с.

УДК 004.413; ГРНТИ 20.51.17

РАЗРАБОТКА ПОДСИСТЕМЫ ПОДДЕРЖКИ ПОЛЬЗОВАТЕЛЕЙ

А.В. Маркин, А.Ю. Баранов

*Рязанский государственный радиотехнический университет имени В.Ф. Уткина,
Российская Федерация, Рязань*

Аннотация. В работе рассматривается решение задачи автоматизации процессов поддержки пользователей, обращающихся в службу технической поддержки. Осуществляется разработка подсистемы, которая смогла бы решить поставленную задачу.

Ключевые слова: подсистема, техническая поддержка, чат-бот.

DEVELOPMENT OF USER SUPPORT SUBSYSTEM

A.V. Markin, A.I. Baranov

*Ryazan State Radio Engineering University named after V.F. Utkin,
Russian Federation, Ryazan*

Annotation. The work solves solution of automating processes users who contact the technical support service. Being development a subsystem that could solve this problem.

Keywords: subsystem, technical support service, chat-bot.

Цель работы

Целью работы является решение задачи автоматизации процесса поддержки пользователей компонентов расчётно-платёжного комплекса «Абонент+». С этой целью ставится задача на разработку подсистемы, которая могла бы решить поставленную задачу автоматизации.

Поддержка пользователей

Под поддержкой пользователей понимаются ответы отдела технической поддержки на вопросы пользователей. На данный момент в технической поддержке пользователей РПК «Абонент+» осуществляется в личном порядке сотрудниками технической поддержки. Даже самый типовой вопрос требует личного внимания специалиста, что существенно сказывается на общей производительности. Часть задач, которыми на данный момент занимаются специалисты отдела технической поддержки может быть переложено на подсистему, которая сама смогла бы общаться с пользователями. Пользователь задаёт вопрос системе, система пытается на него ответить, если же ответ системы пользователя не устраивает, тогда он может обратиться к специалисту-человеку. Помимо простой попытки ответить на поставленный вопрос, подсистема может помочь пользователю грамотнее сформулировать свой вопрос. Бывают ситуации, когда пользователь использует не точные формулировки, из-за чего оказать поддержку пользователю становится намного сложнее.

Существующие решения

Проблема поддержки пользователей не нова, поэтому на данный момент уже существуют различные способы её решения. Все они не универсальны, имеют свои сильные и слабые стороны, поэтому перед разработкой такой системы необходимо тщательно проанализировать поставленную задачу, оценить материальные и временные рамки.

Среди систем поддержки пользователей можно выделить три наиболее популярных подхода: система интерактивного обучения, виртуальный голосовой помощник и чат-бот.

Говоря про систему интерактивного обучения стоит учесть, что такая система не позволяет задать ей вопрос, что значительно сужает спектр применения таких подсистем, поэтому останавливаться здесь нет смысла.

Виртуальный голосовой помощник – достаточно универсальный и многозадачный тип систем поддержки пользователей. В основе такого помощника лежит нейронная сеть, кото-

рая обучаясь со временем, общается с пользователями и подбирает всё более и более подходящие ответы. Помимо того, что с помощником можно общаться через текстовый чат на экране, он поддерживает и голосовое общение. Система голосового помощника может, как и воспринимать речь человека, задающего вопрос, так и синтезировать голосовой ответ. В качестве примера голосового помощника можно привести систему Алиса компании Яндекс (рис. 1) [1].

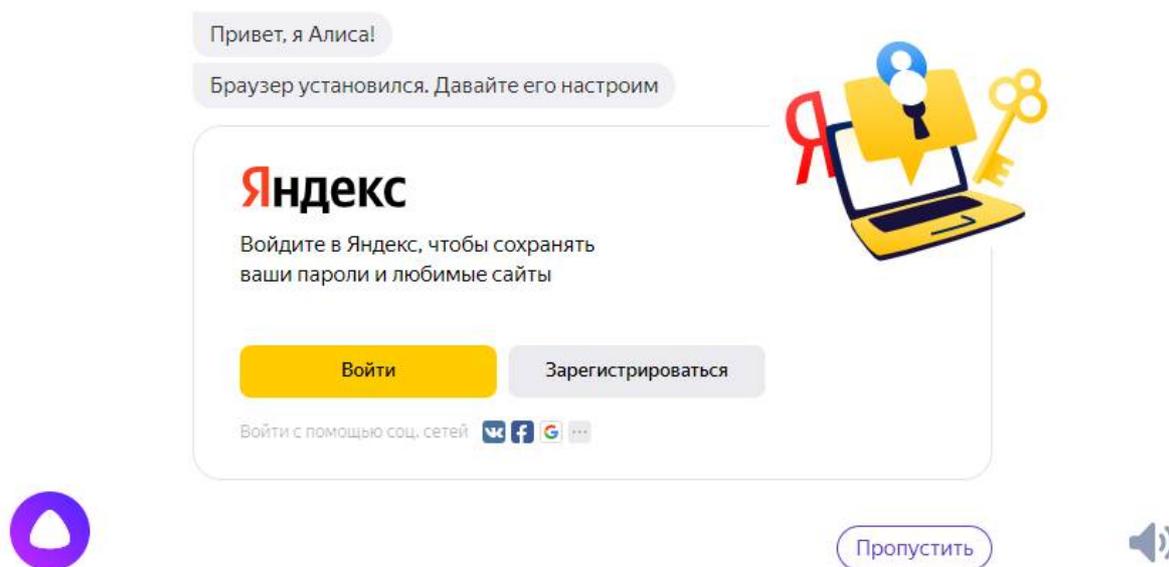


Рис. 1. Голосовой помощник Яндекс Алиса

Такая система характеризуется максимальной гибкостью, как и в способах общения – текст или голос, так и в индивидуальном подходе к каждому пользователю за счёт опыта нейронной сети. Спектр задач виртуальных помощников не ограничивается получением на заданные вопросы типовых ответов или же демонстрации установленной последовательности действий. Главный минус такой системы вытекает из её характеристик – высокая стоимость разработки и сопровождения. В основном, подобные системы могут позволить себе только крупные IT-компании, например, Яндекс или Сбер. Они имеют множество сервисов и услуг, а виртуальный помощник активно помогает пользователю в них сориентироваться. Примерами таких систем на рынке могут служить виртуальные помощники вышеупомянутых крупных компаний Яндекс Алиса, Сбер Салют, Google Assistant и др. [2, 3].

Чат-бота можно классифицировать как промежуточное звено между предыдущими двумя способами. Он является более гибким чем система интерактивного обучения, позволяя пользователю задавать ему вопросы, но в то же время уступает виртуальному помощнику в спектре решаемых задач. В качестве примера можно привести Telegram-бота BotFather, который используется для создания других ботов (рис. 2) [4].

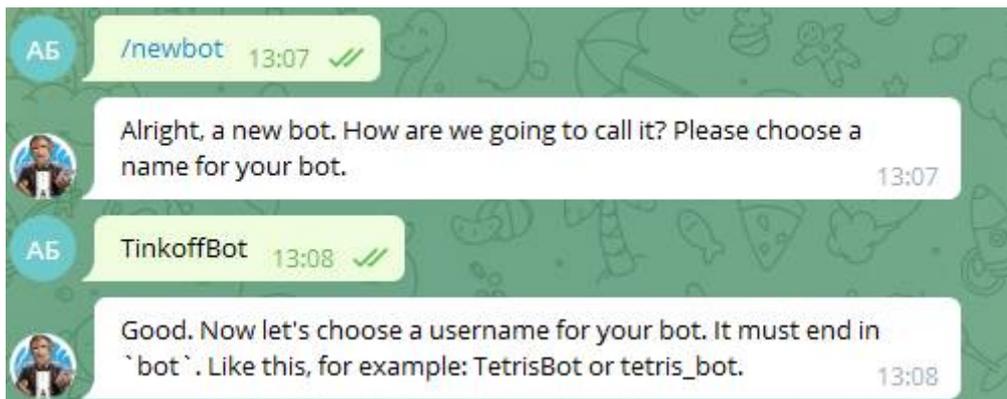


Рис. 2. Telegram-бот BotFather

Такая система имеет более гибкую структуру в сравнении с интерактивной системой обучения; и более низкую стоимость разработки в сравнении с полноценным виртуальным помощником, но в перспективе гораздо меньше функциональных возможностей.

Программный комплекс «Абонент+»

Разрабатываемая система поддержки пользователей не обособленная, она является частью расчётно-платёжного комплекса «Абонент+». Расчётно-платёжный комплекс «Абонент+» - это полнофункциональное решение для автоматизации всего спектра задач, связанных с расчётом и приёмом платежей за жилищно-коммунальные услуги, подготовкой справок, счетов-извещений, счетов квитанций, формированием итоговой отчётности о результатах деятельности. Интегрированное решение, обеспечивающее онлайн взаимодействие между всеми компонентами комплекса. Структура программного комплекса «Абонент+» показана на рисунке 3 [5].

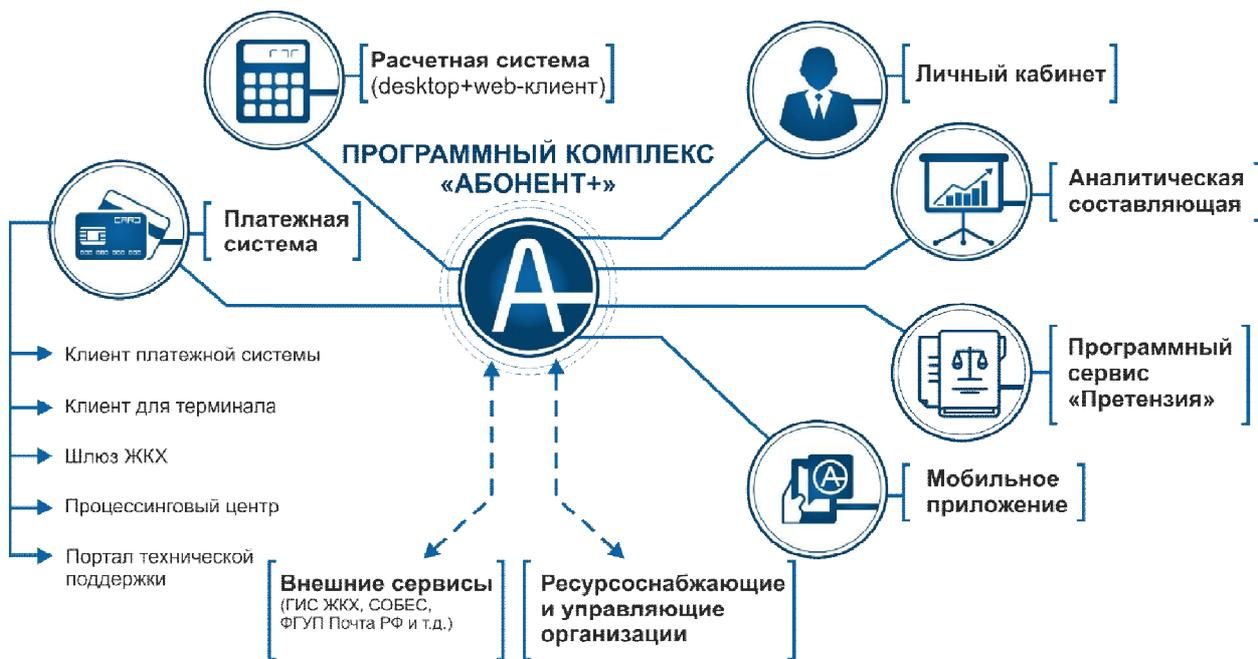


Рис. 3. Структура программного комплекса «Абонент+»

Подсистема поддержки пользователей обязана владеть исчерпывающей информацией о всё программном комплексе в целом и о каждом из его компонентов, в частности.

Проектирование подсистемы

Спроектированная система представлена на рисунке 4.

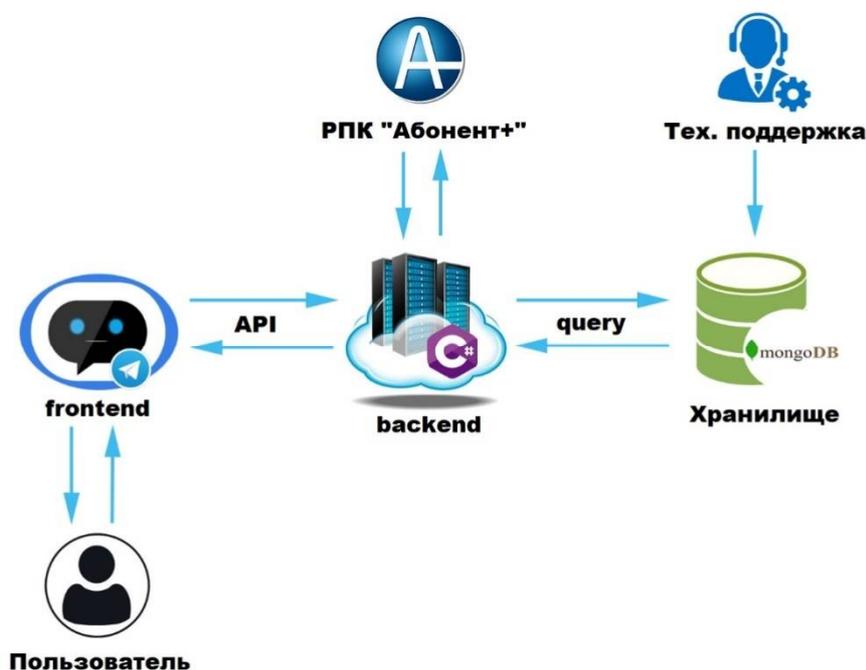


Рис. 4. Спроектированная система поддержки пользователей

Ядром системы является серверное приложение на языке С# [6], оно, через API взаимодействует с интерфейсной частью системы, реализованной в виде чат-бота в мессенджере Telegram [7]. Сам бот общается непосредственно с пользователем по методологии «вопрос-ответ». Также сервер имеет возможность взаимодействовать с другими компонентами программного комплекса «Абонент+». В качестве хранилища в системе используется база данных на СУБД MongoDB [8]. Сервер общается с хранилищем при помощи запросов языка Mongo. Сотрудники поддержки смогут сами вносить изменения в хранилище, тем самым увеличивая количество возможных ответов системы на вопросы пользователей.

Библиографический список

1. Голосовой помощник Алиса | Yandex N.V. [Электронный ресурс]. URL: <https://yandex.ru/alice> (дата обращения 04.04.2022).
2. Голосовой помощник Салют | ООО «СберДевайсы» [Электронный ресурс]. URL: <https://salute.sber.ru/> (дата обращения 04.04.2022).
5. Google Assistant | Google Inc. [Электронный ресурс]. URL: <https://assistant.google.com/> (дата обращения 04.04.2022).
6. Telegram-бот BotFather | Telegram FZ-LLC [Электронный ресурс]. URL: <https://telegram.me/BotFather> (дата обращения 04.04.2022).
7. Расчётно-платёжный комплекс «Абонент+» | ООО «Абонент+» [Электронный ресурс]. URL: <https://www.abonent.plus/> (дата обращения 27.03.2022).
8. Язык программирования С# | Microsoft Corp. [Электронный ресурс]. URL: <https://docs.microsoft.com/ru-ru/dotnet/csharp/> (дата обращения 04.04.2022).
9. Облачный кроссплатформенный мессенджер Telegram | Telegram FZ-LLC [Электронный ресурс]. URL: <https://telegram.org/> (дата обращения 27.03.2022).
10. СУБД MongoDB | MongoDB Inc [Электронный ресурс]. URL: <https://www.mongodb.com/> (дата обращения 27.03.2022).

УДК 51-74: 681.2.087; ГРНТИ 76.13.15

АНАЛИЗ ОСОБЕННОСТЕЙ ПРИМЕНЕНИЯ СЛОЖНЫХ ДИСКРЕТНЫХ ОТСЧЕТОВ ДЛЯ ОБРАБОТКИ ЭЛЕКТРОКАРДИОСИГНАЛА

Ю.А. Булгаков, А.А. Михеев

*Рязанский государственный радиотехнический университет имени В.Ф. Уткина,
Российская Федерация, Рязань*

Аннотация. В работе рассматриваются вопросы влияния вариабельности периода дискретизации на обработку электрокардиосигналов, в частности устранение дрейфа изолинии. Приводится вариант их решения на основе цифровой обработки измерительного сигнала с использованием дискретных отсчетов сигнала в виде сложных дискретных отсчетов.

Ключевые слова: дискретный отсчет, сложный дискретный отсчет, вариабельность периода дискретизации, спектральная зона.

ANALYSIS OF THE FEATURES OF THE USE OF COMPLEX DISCRETE SAMPLES FOR ELECTROCARDIOSIGNAL PROCESSING

Yu.A. Bulgakov, A.A. Mikheev

*Ryazan State Radio Engineering University named after V.F. Utkin,
Russia, Ryazan*

Abstract. Questions of the influence of the sampling period variability on the processing of electrocardiosignals, in particular, the elimination of the isoline drift, have been considered. A variant of their solution based on the use of complex discrete samples for digital processing of the measuring signal has been given.

Keywords: discrete sample, complex discrete sample, sampling period variability, spectral zone.

Основным методом диагностики сердечно-сосудистых заболеваний является электрокардиография. Получение информативного электрокардиосигнала сопровождается действием различных мешающих факторов (помех). Одной из характерных помех для электрокардиосигнала (ЭКС) является дрейф изолинии. Дрейф изолинии – это помеха, представляющая собой низкочастотные колебания с частотой до 1 Гц, обусловленные влиянием на ЭКС аддитивных низкочастотных помех (поляризация электродов, влияние дыхания, артефакты, временной дрейф и т. п.).

Одним из типовых приемов устранения дрейфа изолинии является фильтрация смеси ЭКС и дрейфа изолинии с помощью фильтра верхних частот (ФВЧ). Главный недостаток данного метода заключается в том, что вместе с сигналом помехи удаляются спектральные составляющие ЭКС, лежащие в низкочастотной области. При этом в полученном электрокардиосигнале искажается ST- сегмент [1], положение и форма которого содержат важную диагностическую информацию.

Снизить влияние дрейфа изолинии на электрокардиосигнал без искажения элементов последнего можно с помощью применения сложных дискретных отсчетов (СДО) [2].

Достоинства сложных дискретных отсчетов

Сложные дискретные отсчеты измерительных сигналов образуются добавлением к основным отсчетам одной или нескольких пар дополнительных отсчетов. В каждой паре один дополнительный отсчет сдвинут относительно основного отсчета на τ_c влево, а другой дополнительный отсчет – на τ_c вправо. Дополнительные отсчеты умножаются на соответствующие масштабные коэффициенты. При подавлении одной спектральной зоны к основным отсчетам необходимо добавить одну пару дополнительных отсчетов. Так, например, для подавления двух спектральных зон потребуется к основному отсчету добавить две дополнительные пары отсчетов (рисунок 1).

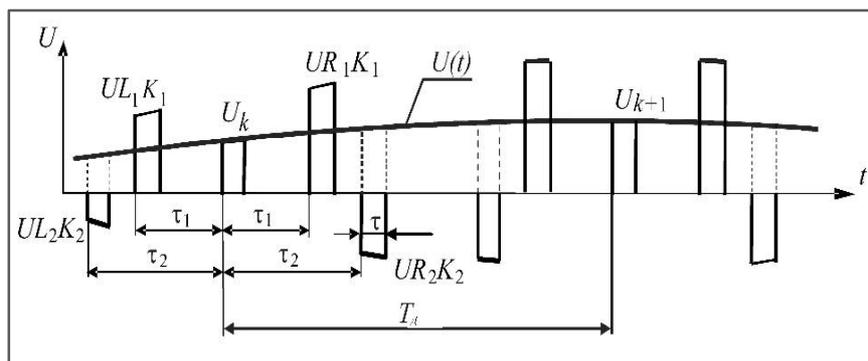
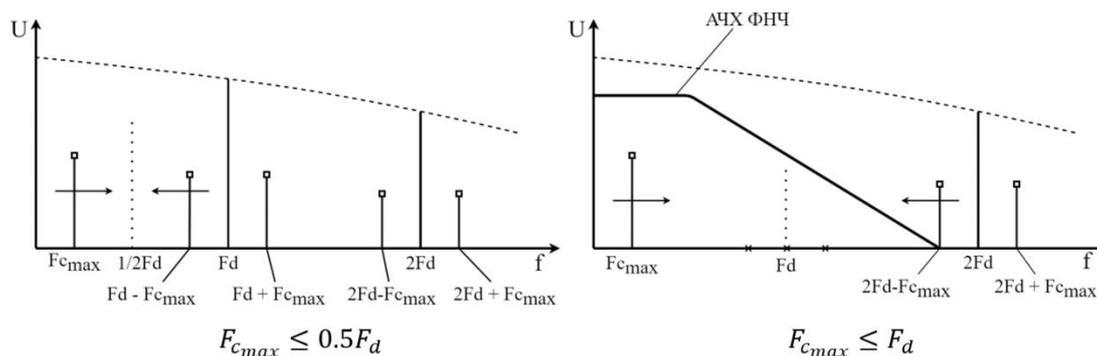


Рис. 1. Формирование СДО для подавления двух спектральных зон

Представление дискретных отсчетов измерительных сигналов сложными дискретными отсчетами позволяет трансформировать спектр полученной последовательности таким образом, что составляющие заданных спектральных зон обращаются в ноль. Это дает возможность или расширить возможный частотный диапазон контролируемых процессов (рисунок 2) или снизить требования к порядку фильтра нижних частот, используемому для восстановления исходного непрерывного сигнала, отображающего контролируемый процесс (рисунок 3).



АЧХ ФНЧ – амплитудно-частотная характеристика фильтра нижних частот
Рис. 2. Спектр исходного сигнала и с подавленной первой спектральной зоной

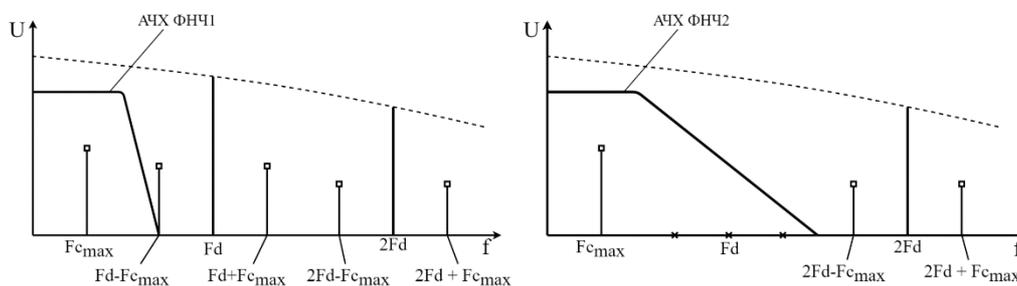


Рис. 3. Спектры сигналов и АЧХ ФНЧ

Данные достоинства СДО позволяют преодолеть указанные выше недостатки метода устранения дрейфа изолинии с помощью ФВЧ. При этом надо учитывать особенности построения СДО на ТР-интервале электрокардиосигнала, обусловленные наличием variability сердечного ритма [3].

Особенности практического применения СДО

Выражение, связывающее амплитуду составляющей k -й спектральной зоны в спектрах последовательности основных отсчетов сигнала дрейфа изолинии и сложных дискретных отсчетов, имеет вид:

$$A_{\text{СДО}k} = U_0 \frac{2\tau}{T_d} \frac{\sin\left(k\pi \frac{\tau}{T_d}\right)}{k\pi \frac{\tau}{T_d}} \left[1 + 2K_1 \cos\left(2\pi k \frac{\tau_c}{T_d}\right) \right]. \quad (1)$$

Чтобы подавить в спектре последовательности СДО составляющие k -ой спектральной зоны, необходимо выполнить условие: $A_{\text{СДО}k} = 0$. Как следует из (1), выполнить это условие можно, приравняв нулю выражение в квадратных скобках

$$\left[1 + 2K_1 \cos\left(2\pi k \frac{\tau_c}{T_d}\right) \right] = 0. \quad (2)$$

При заданных периоде T_d дискретизации и сдвиге τ_c дополнительных отсчетов уравнение (2) решается относительно масштабного коэффициента K_1 . Однако изменчивость T_d в процессе съема отсчетов приводит к нарушению условия (1) при уже выбранном K_1 и, соответственно, к неполному подавлению спектральных составляющих выбранной спектральной зоны.

Ритму сердца физиологически присуща вариабельность. Поэтому отсчеты сигнала дрейфа изолинии, взятые на ТР-интервале, будут иметь вариабельность периода дискретизации. Математической моделью таких отсчетов является сигнал с частотно-импульсной модуляцией (ЧИМ) [4]. Спектр ЧИМ-сигнала содержит составляющие в нулевой спектральной зоне (полоса частот от 0 до $Fd/2$, где Fd – частота дискретизации), на частотах kFd , где $k=1, 2, \dots$ – номер спектральной зоны, и на частотах $kFd \pm nFm$, где Fm – частота модулирующего сигнала, $n=1, 2, \dots$ – номера боковых составляющих в каждой спектральной зоне [5]. При этом амплитуды левых и правых боковых составляющих не равны между собой (рисунок 4).

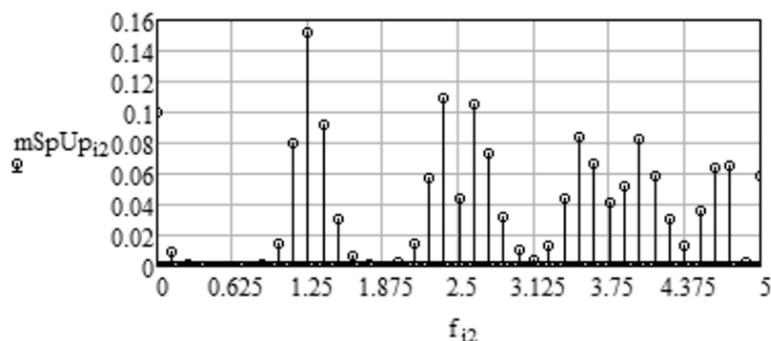


Рис. 4. Спектр последовательности основных отсчетов сигнала с ЧИМ

В этом случае при формировании СДО сигнала дрейфа изолинии в соответствии с (2), как и при равномерной дискретизации, центральная составляющая в выбранной спектральной зоне подавляется полностью, а боковые – не полностью (рисунок 5).

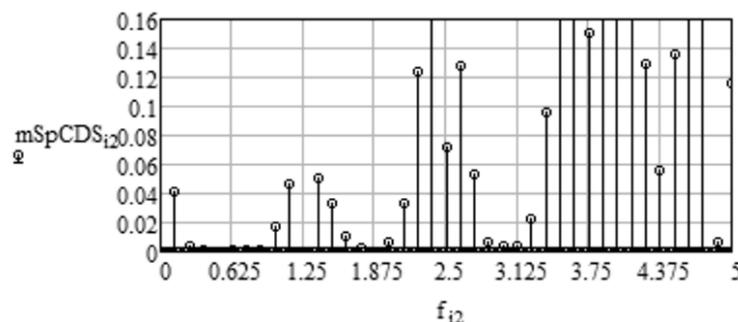


Рис. 5. Спектр СДО

Для того, чтобы подавить все составляющие выбранной спектральной зоны, центральную и боковые составляющие, необходимо, чтобы условие (2) было выполнено для каждой спектральной составляющей [6].

Для выделения составляющих дрейфа изолинии достаточно подавить первую спектральную зону на частоте дискретизации Fd и выделить с помощью ФНЧ составляющие нулевой спектральной зоны.

Количество боковых составляющих в каждой спектральной зоне зависит от индекса частотной модуляции β . При $n > \beta$ амплитуды боковых составляющих быстро уменьшаются [7]. При изменчивости периода дискретизации, определяемой вариабельностью ритма сердца, индекс частотной модуляции для первой спектральной зоны не превышает 1. Поэтому при рассмотрении спектра ЧИМ-сигнала в этом случае можно ограничиться боковыми составляющими с номером $n=1$.

Таким образом, для подавления первой спектральной зоны надо, чтобы были равны нулю составляющие на частотах: Fd , $Fd-Fm$, $Fd+Fm$. В этом случае для построения СДО требуется три пары дополнительных отсчетов. Масштабные параметры K_j для каждой дополнительной пары отсчетов определяются из решения системы трех уравнений с заданными сдвигами τ_{sj} ($j=1, 2, 3$):

$$\begin{cases} 1 + 2 \sum_{j=1}^3 K_j \cos(2\pi Fd \tau_{sj}) = 0 \\ 1 + 2 \sum_{j=1}^3 K_j \cos[2\pi (Fd - Fm) \tau_{sj}] = 0 \\ 1 + 2 \sum_{j=1}^3 K_j \cos[2\pi (Fd + Fm) \tau_{sj}] = 0 \end{cases} \quad (3)$$

Алгоритм формирования СДО при вариабельности периода дискретизации

Для реализации СДО в обработке ЭКС требуется учитывать ряд особенностей:

- для устранения дрейфа изолинии для обработки берутся отсчеты электрокардиосигнала на ТР-интервале;
- так как формирование отсчетов ведется в реальном времени, то формировать отсчеты нужно с крайнего левого дополнительного отсчета.

Алгоритм формирования СДО при вариабельности периода дискретизации приведен на рисунке 6.

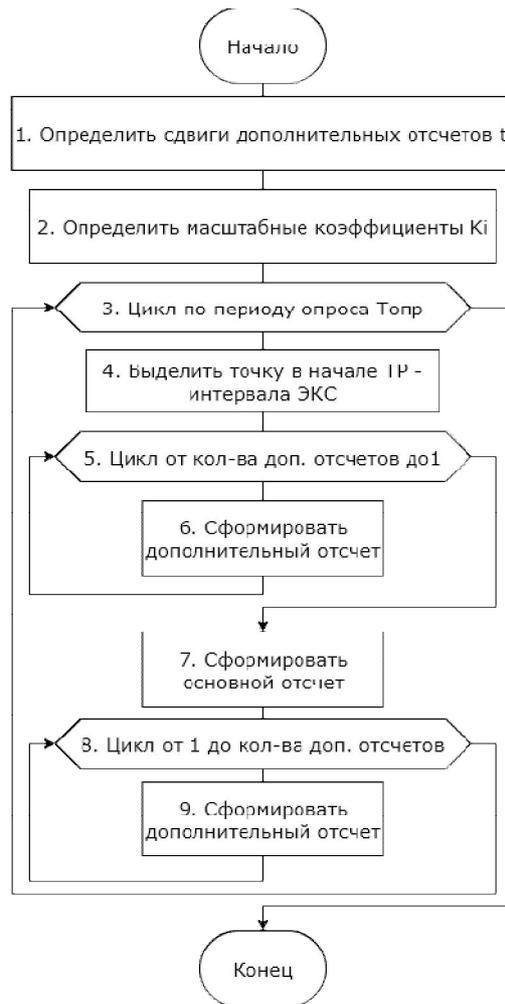


Рис. 6. Алгоритм формирования СДО для обработки ЭКС

Библиографический список

1. Рангайян Р.М. Анализ биомедицинских сигналов. Практический подход/ Пер. с англ. под ред. А.П. Немирко – М.: ФИЗМАТЛИТ. 2007. – 440 с.
2. Патент РФ 2251968, МКИ7 А61 В 5/0402. Способ устранения дрейфа изолинии электрокардиосигнала и устройство для его осуществления Открытия. Заявка № 2003128606 от 23.09.2003. Изобретения. 2005, №14. Оpubл. 20.05.2005.
3. Булгаков Ю.А., Витязева Т.А., Михеев А.А. Сложные дискретные отсчеты в задачах обработки электрокардиосигнала // Биомедицинская радиоэлектроника. 2021. Т. 24. № 4. С. 76–82.
4. L. Sornmo, P. Laguna. Bioelectrical Signal Processing in Cardiac and Neurological Applications. Academic Press. Elsevier. 2005. 688 p.
5. Борисов Ю.П., Пенин П.И. Основы многоканальной передачи информации. –М.: Связь, 1967. –436 с.
6. Yu. Bulgakov, T. Vitiazeva and A. Mikheev Research of the Spectrum of a Complex Discrete Samples with Sample Rate Variability/ 10th Mediterranean Conference on Embedded Computing (MECO – 2021) Proceedings. Budva, Montenegro. IEEE Catalog Number: CFP2039T-ART. pp. 323-326.
7. Гоноровский И.С., Демин М.П. Радиотехнические цепи и сигналы: Учеб. пособие для вузов. 5-е изд., перераб. и доп. М.: Радио и связь, 1994. 480 с.

УДК 004.032.26; ГРНТИ 50.47.02

РАЗРАБОТКА НЕЙРОСЕТЕВОЙ ПОДСИСТЕМЫ ФУНКЦИОНАЛЬНОГО ПРЕОБРАЗОВАНИЯ ЧАСТОТЫ В КОД

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 первичного преобразования неэлектрической величины в электрическую и вторичного преобразования электрической величины в цифровой код. Также представлены методика настройки и алгоритм обучения нейронной сети преобразователя. Приведены полученные результаты обучения.

Ключевые слова: нейронные сети, обучение, алгоритм «онлайн» обучения, аппроксимация данных, вторичное преобразование.

DEVELOPMENT OF NEURAL NETWORK SUBSYSTEM OF FUNCTIONAL FREQUENCY TO CODE CONVERSION

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 considers the process of measuring a non-electric quantity, which consists of the primary conversion of a non-electric quantity into an electrical quantity and the secondary transformation of an electrical quantity into a digital code. A tuning technique and an algorithm for training the neural network of the converter are also presented. The received results of training are given.

Keywords: neural networks, training, «online» learning, data approximation, secondary transformation

Плодотворное управление сложными техническими объектами, автоматизация технологических процессов или развитие научных исследований требуют больших измерений разнообразных физических величин. Для решения вышеупомянутых задач используются информационно-измерительные системы (ИИС).

Для сбора информации об объекте исследования в ИИС применяются датчики (рисунок 1). Датчики преобразуют информацию об исследуемых параметрах в электрические сигналы, изменение параметров которых пропорционально изменению параметров контролируемых процессов. Выходными сигналами датчиков могут быть напряжение, ток или заряд, описываемые такими характеристиками как амплитуда, частота или фаза, которые являются информативными параметрами сигнала.

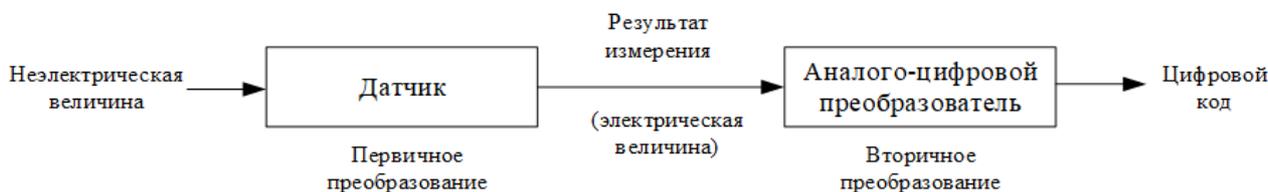


Рис. 1. Процесс измерения неэлектрической величины

Для дальнейшей обработки полученных результатов измерений необходимо трансформировать аналоговую величину в цифровой код. Для точной обработки снятых измерений возникает необходимость решения задачи линеаризации тарировочной характеристики, которая описывает зависимость входной электрической величины от измеряемой неэлектрической (рисунок 2).

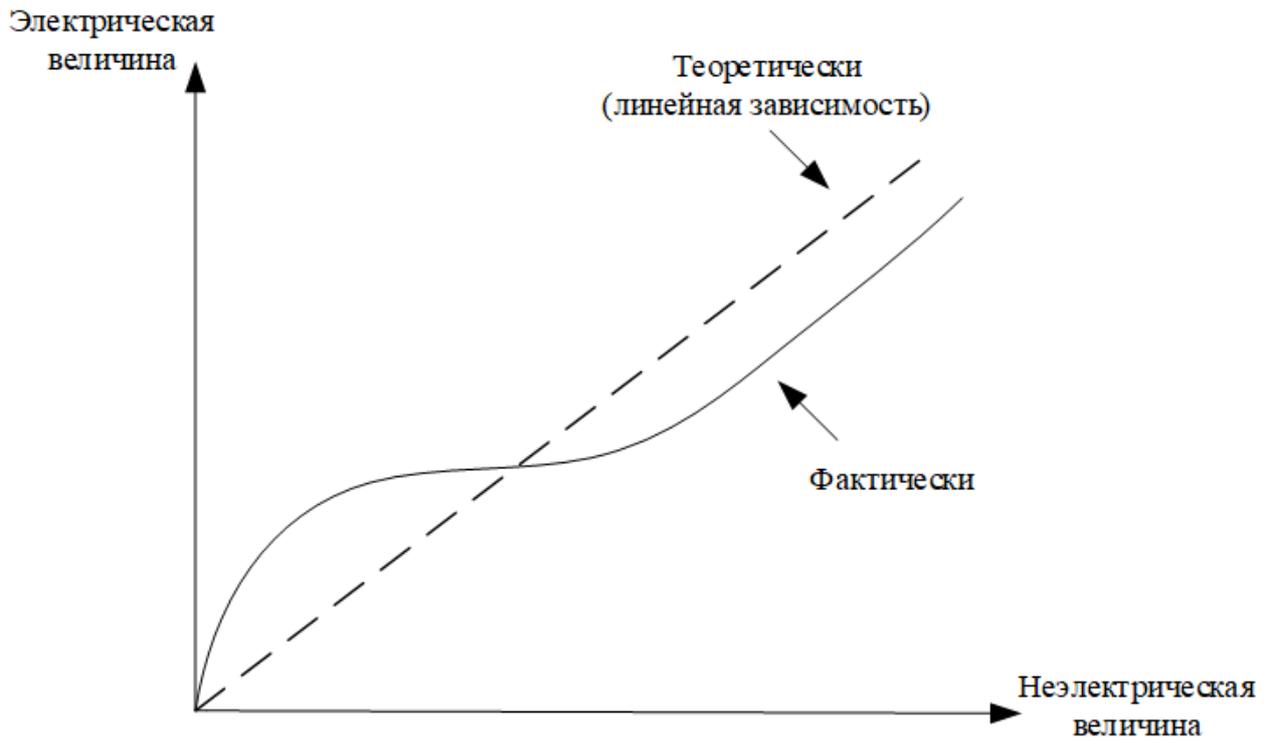


Рис. 2. Пример нелинейной зависимости

Одним из возможных решений задачи линейаризации является применение функциональных преобразователей информации, использующих математический аппарат искусственных нейронных сетей [1].

Для решения задачи проектирования преобразователя «частота – позиционный код» была выбрана упрощённая двухслойная структура персептронной сети следующего вида: нулевой слой осуществляет операцию преобразования «аналог – цифра», первый слой использует сигмоидную функцию активации, а второй слой осуществляет сложение выходных значений первого слоя (рисунок 3).

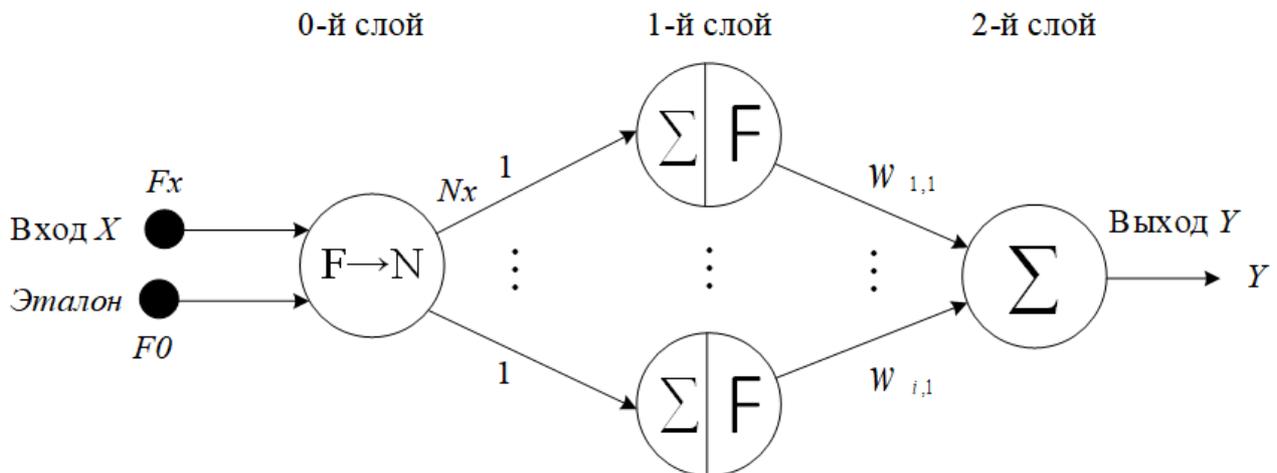


Рис. 3. Структура нейросетевого аппроксиматора

Функционирование нейросетевого преобразователя $x \rightarrow y^*$ на основе однослойного персептрона в качестве аппроксиматора нелинейной функции можно описать выражением:

$$Y = \sum_i w_{i,1} (F(\sum_i X)),$$

где Y – значение выхода нейронной сети;

w_i – весовой коэффициент 2-го слоя нейронной сети;

F – функция активации нейронов.

В качестве передаточной функции нейросетевой структуры была выбрана логистическая функция с двумя параметрами:

$$F = \frac{1}{1 + e^{-\frac{(S-a)}{b}}},$$

где a – аддитивный параметр активационной функции;

b – мультипликативный параметр;

S – взвешенная сумма входных сигналов нейрона.

Для решения задачи нахождения оптимальных весов синаптических связей нейронную сеть необходимо обучить. Как правило, настоящий процесс происходит согласно специальным алгоритмам обучения. Предлагаемая методика настройки нейронной сети преобразователя, а также один из используемых алгоритмов обучения сети представлены на рисунках 4, 5.

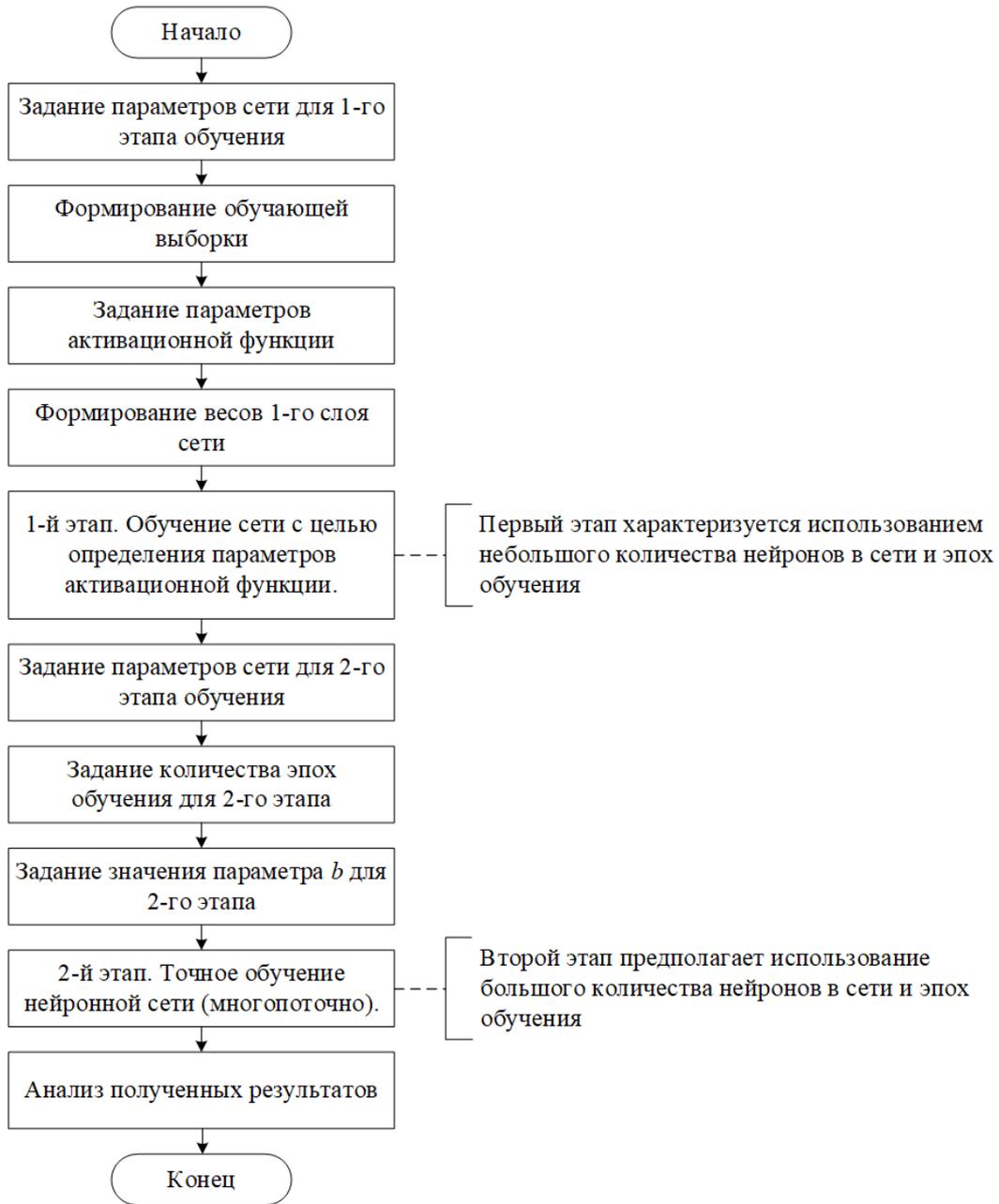


Рис. 4. Методика настройки сети

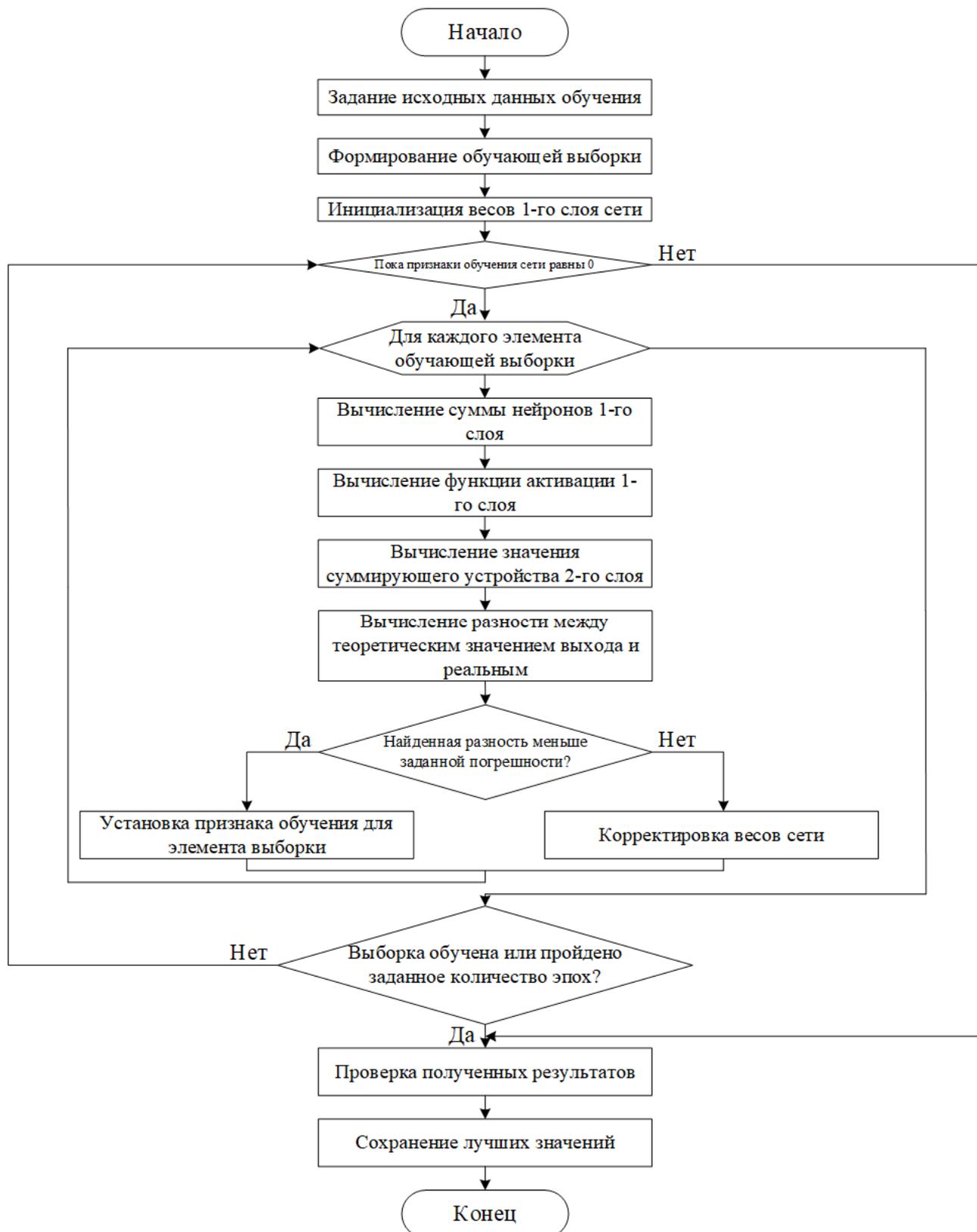


Рис. 5. Блок-схема «онлайн» алгоритма обучения

Согласно разработанной методике на первом этапе обучения задаются основные параметры нейронной сети, например: количество нейронов, объем обучающей выборки, количество эпох обучения, целевая погрешность и т.д. Кроме того, этот этап является «грубым»

обучением и подразумевает задание основных параметров нейронной сети в небольшом количестве. Далее для каждого элемента обучающей выборки вычисляется сумма нейронов первого слоя, а также значение активационной функции каждого нейрона. Следующим шагом вычисляется значение суммирующего устройства второго слоя и вычисляется разность между выходом нейронной сети и теоретическим значением. Если полученная разность меньше целевой погрешности, то обучение считается выполненным и происходит переход к следующему элементу выборки. В ином случае необходимо изменить значение весов нейронных связей. Обучение заканчивается если для каждого элемента из обучающей выборки была достигнута целевая погрешность или количество эпох, заданных в начале обучения. После этого происходит проверка полученных результатов и сохранение лучших значений для: весов нейронных связей, коэффициента a и параметра активационной функции b .

Второй этап обучения подразумевает точное обучение. Алгоритм обучения на втором этапе аналогичен обучению сети на первом этапе. Однако на этом этапе рекомендуется увеличить значения параметров нейронной сети.

Обучение нейросетевой структуры основывается на применении «онлайн» и «оффлайн» алгоритмов обучения. Для алгоритма «онлайн» обучения уточнение весов происходит после предъявления каждой обучающей выборки, в то время как для «оффлайн» алгоритма характерно вычисление весов путём суммирования всех их приращений, полученных после предъявления всех обучающих данных [2].

В таблице 1 представлены полученные результаты обучения нейронной сети для различных сходных сигналов.

Таблица 1. Результаты обучения сети преобразователя на основе «онлайн» обучения

Функция	Диапазон значений	Количество нейронов	Δ , %	Количество значащих разрядов
$\cos\left(\pi \cdot \frac{x}{2}\right)$	$x \in [0; 1]$	4	0,0993987590769297	10
		16	0,0248783033052313	12
$\sin\left(\pi \cdot \frac{x}{2}\right)$	$x \in [0; 1]$	4	0,0509257752130732	11
		16	0,0284542697267298	11
		32	0,0249656924234093	12
$\tan\left(\pi \cdot \frac{x}{2}\right)$	$x \in [0; 0,667]$	4	0,0574409108137645	11
		16	0,014410190532728	12
		32	0,0170964171714126	12
$\log_2(1+x)$	$x \in [0; 1]$	4	3,15201853289433	9
		16	0,0247874864203024	12
$\sin(\pi x)$	$x \in [0; 1]$	4	2,22913449690767	9
		16	0,0983711976737377	10
		32	0,0249378815579182	12
$\cos(\pi \cdot \sqrt{x})$	$x \in [0; 0,25]$	4	0,100734811046566	9
		16	0,0500444691116439	11
		32	0,0249986682331185	12

Полученные результаты говорят о требуемой 12-ти разрядной точности воспроизведения функциональных зависимостей, что является достаточно высоким показателем и может применяться для промышленных измерений.

Библиографический список

1. Локтюхин В. Н., Челебаев С. В., Антоненко А. В. Нейросетевые аналого-цифровые преобразователи. — М.: Горячая линия–Телеком. — 2010 г. — 128 с.
2. Запрягаев С.А., Карпушин А.А. Вычисление и обучение искусственных нейронных сетей прямого распространения на графическом процессоре. — Вестник ВГУ, №1. — 2011 г. — С. 157-164.

УДК 51-7; ГРНТИ 06.81.23

СПОСОБЫ СОКРАЩЕНИЯ ДЛИТЕЛЬНОГО ЦИКЛА ОБРАБОТКИ ЗАЯВОК НА СКЛАДЕ КАНЦЕТОВАРОВ

М.В. Свиридова, С.В. Челебаев

Рязанский государственный радиотехнический университет имени В.Ф. Уткина,
Российская Федерация, Рязань, mari.sviridova1@yandex.ru

Аннотация. В данной работе рассматривается проблема длительного цикла обработки заказов на складе канцетоваров. Теория массового обслуживания и методы оценки основных показателей (KPI - Key Performance Indicator) производственных ресурсов на основе многокритериальной оптимизации являются инструментами, которые позволяют решить актуальную для предприятия проблему.

Ключевые слова: производственные ресурсы, KPI, теория массового обслуживания, многокритериальная оптимизация, моделирование.

WAYS TO REDUCE THE LONG REQUEST PROCESSING CYCLE IN THE STATIONERY WAREHOUSE

M.V. Sviridova, S.V. Chelebaev

Ryazan State Radio Engineering University named after V.F. Utkin,
Russia, Ryazan, mari.sviridova1@yandex.ru

Abstract. The article deals with the problem of a long order processing cycle in a stationery warehouse. The theory of queuing and methods of assessing the main indicators (KPI - Key Performance Indicator) of production resources based on multi-criteria optimization are tools that allow solving an urgent problem for the enterprise.

Keywords: production resources, KPI, queuing theory, multi-criteria optimization, modeling.

Существующая на предприятии система управления складом (WMS) помогает упростить складские бизнес-процессы (БП) – от приемки товара до его отгрузки. Однако она не учитывает, что на складе могут образовываться очереди в ходе обработки заявок, что приводит к замедлению производственного цикла. Определяющую роль в обработке заявок играют производственные ресурсы (ПР) – кладовщики. В данный момент выбор кладовщиков полностью зависит от субъективного мнения руководителей складского комплекса (СК), которые при распределении ПР на БП опираются на рабочий график кладовщиков и примерную загрузку склада в момент выбора. Для решения проблемы можно использовать теорию массового обслуживания, которая позволяет сбалансировать систему так, что задержки и очереди при обработке заявок сократятся, а использование методов многокритериальной оптимизации для выбора кладовщиков на определенный БП по количеству n параметров позволит более объективно подходить к выбору ПР.

Получение заданного эффекта при минимуме затрат или получение максимального эффекта при заданном ограниченном ресурсах – основные задачи при поиске оптимального решения [1]. Добавление в WMS возможности имитации нагрузки системы на основе ТМО позволит ответить на вопрос – как уменьшить число очередей при обработке заявок и какое минимальное количество ПР для этого потребуется.

В качестве исходных данных для моделирования выступают следующие показатели: четыре периода времени (утро, день, вечер, ночь), среднее число заявок (шт) в каждый из промежутков времени (N), время поступления заявок (мин) (T), скорость обработки одной заявки (мин) (t).

С помощью этой информации можно определить основные показатели – интенсивность потока обслуживания (m), интенсивность потока заявок (I), количество человек для обработки заявок в каждый из периодов времени (r) по формулам 1-3:

$$I = N/t \quad (1),$$

$$m = 1/t \quad (2),$$

$$r = I/m \quad (3).$$

В таблице 1 показаны результаты расчетов показателей.

Таблица 1 – Показатели обработки заявок на складе

	Утро	День	Вечер	Ночь
Число заявок (N)	12	36	24	10
Время поступления (T)	25	35	40	50
Интенсивность потока заявок (I)	0,48	1,028571	0,6	0,2
Время обработки одной заявки (t)	5	7	8	16
Интенсивность потока обслуживания (m)	0,2	0,142857	0,125	0,0625
Количество человек (r)	2,4	7,2	4,8	3,2
Результат моделирования (человек)	3	8	5	4

Поступление заявок – случайный процесс (рисунок 1). Для решения задач теории массового обслуживания необходимо построить и проанализировать его математическую модель случайного процесса. Случайный процесс называется Марковским, если для любого момента времени вероятностные характеристики процесса в будущем зависят только от его состояния в данный момент и не зависят от того, когда и как система пришла в это состояние [2].

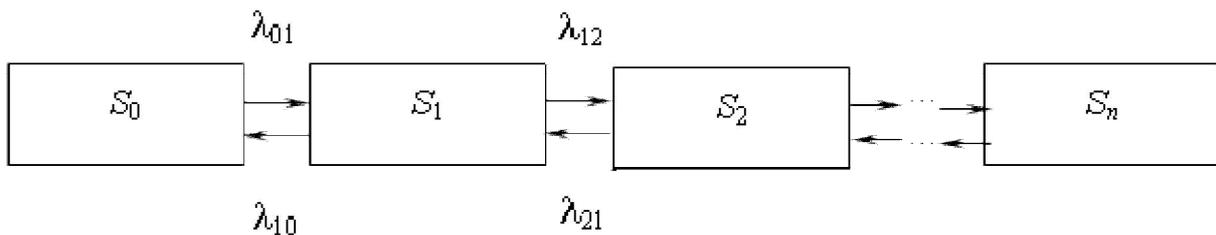


Рис. 1. Случайный процесс

Если максимальное количество заявок в очереди равно 40, то на основе рассчитанных значений вероятность нахождения системы в каждом из состояний в каждом промежутке времени показана на рисунках 2 – 5.

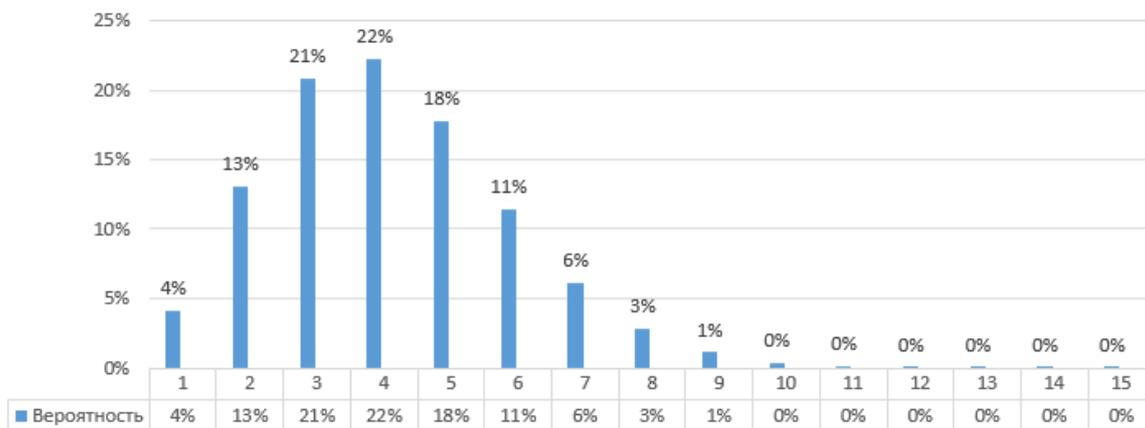


Рис. 2. Вероятность образования очередей утром

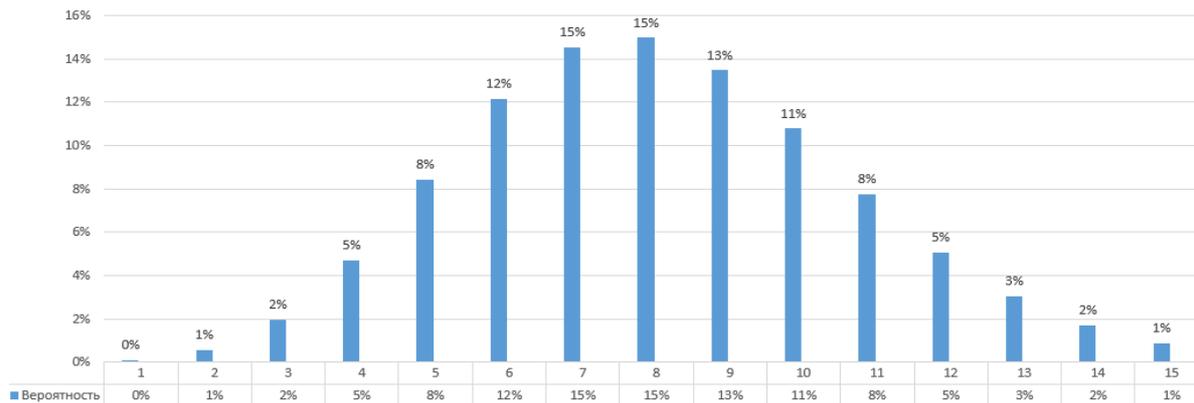


Рис. 3. Вероятность образования очередей днем

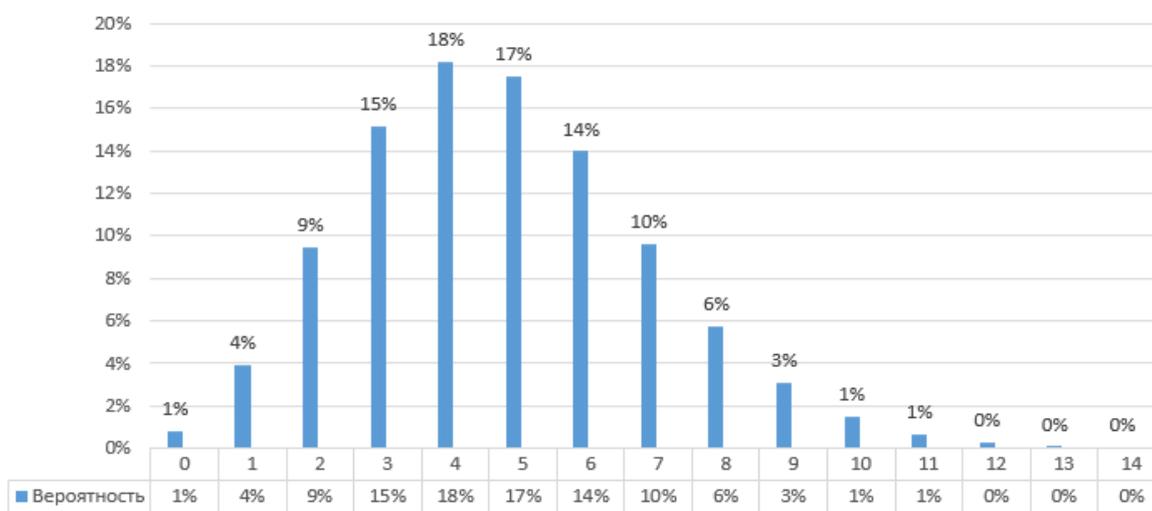


Рис. 4. Вероятность образования очередей вечером

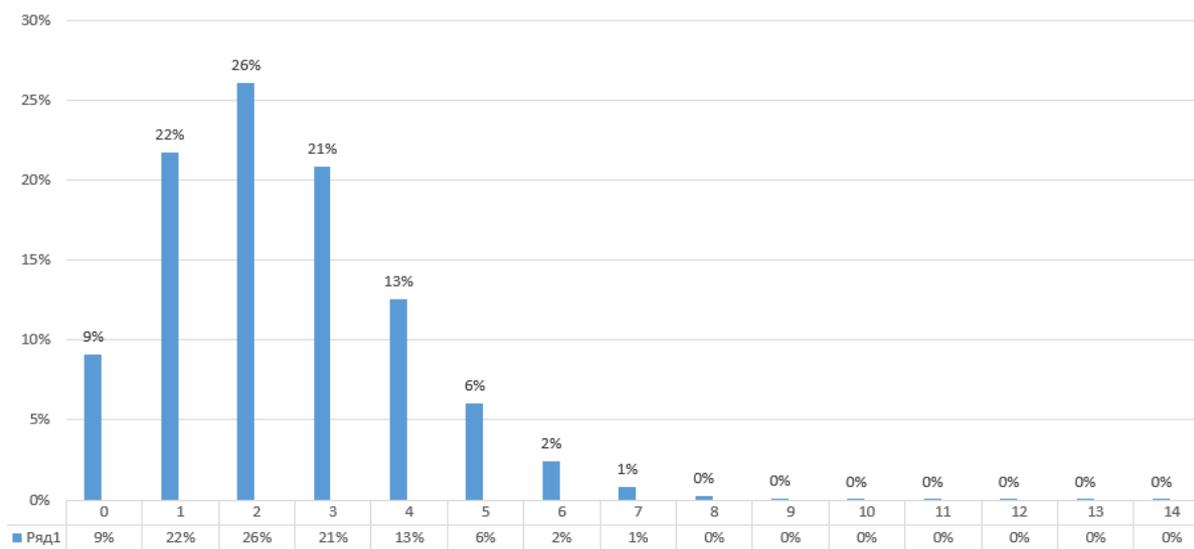


Рис. 5. Вероятность образования очередей ночью

Важными показателями при обработке заявок для предприятия являются: абсолютная (A) и относительная (Q) пропускные способности, среднее количество (n) занятых приборов

(ПР), коэффициент простоя (К). Оценить сокращение очередей можно на основе графиков (рисунки 2-5) и на основе следующих соотношений:

$$A = I * (1 - P_{отк}) \quad (4),$$

$$Q = (1 - P_{отк}) \quad (5),$$

$$n = A/m \quad (6),$$

$$K = 1 - n/r \quad (7).$$

Результаты расчетов показаны в таблице 2. Q напрямую характеризует загруженность ПР, образование очередей сокращается, так как нагрузка на кладовщиков – высокая. Это недостаток метода.

Таблица 2 – Оценка результатов моделирования

	УТРО	ДЕНЬ	ВЕЧЕР	НОЧЬ
A	0,479999895	1,019644	0,599805	0,199998896
Q	100%	99%	100%	100%
f	2,399999475	7,137507	4,798438	3,199982336
K	20%	11%	4%	20%

На основе теории массового обслуживания можно получить количество производственных ресурсов для обработки большого количества заявок, а моделирование БП поможет избежать появления очередей и/или минимизировать их появление.

Имея информацию о количестве кладовщиков и их показателях работы на складе, можно определить из общего числа всех ПР нужное количество людей с наилучшими показателями. При выборе решений важно определить последствия, а предположения человека могут оказаться ошибочным, так как не учитывают все факторы, влияющие на результат [3]. Использование таких методов многокритериальной оптимизации как метод контрольных показателей, идеальной и антиидеальной точки, линейной свертки всех критериев, позволит учесть все критерии (показатели, характеризующие ПР) при выборе кладовщиков для конкретного БП.

Данные о показателях ПР представлены в таблице 3. Выбор кладовщиков будет происходить по семи критериям.

Таблица 3 – Данные о производственных ресурсах

	ФИО	Отборов	Скорость	Опыт	Зона	Двойной контроль	Опозданий	Ошибок
1	Иванов	15	6	5	1	0	1	1
2	Петров	23	4	2	0	0	0	2
3	Сидоров	58	2	1	1	0	0	0
4	Самойлов	22	9	2	2	0	0	0
5	Лосев	99	2	2	1	0	0	1
6	Баранов	15	8	3	3	0	1	1
7	Смирнов	22	3	4	8	0	1	1
8	Горцев	66	2	2	2	0	0	1
9	Калинин	15	1	2	7	0	0	1
10	Петрухин	1	5	1	1	1	0	1
11	Овечкин	3	3	2	5	1	0	1
12	Ломов	59	9	5	6	1	1	1
13	Перевалов	12	1	1	1	1	1	1
14	Димитров	3	5	3	0	0	0	1

Для выбора ПР по двум основным критериям (количество отборов и скорость) необходимо построить график и определить неуправляемые значения. При этом следует учитывать, что количество отборов должно быть максимальным (кладовщик в таком случае считается опытным), а скорость – минимальной (показатель быстрого кладовщика).

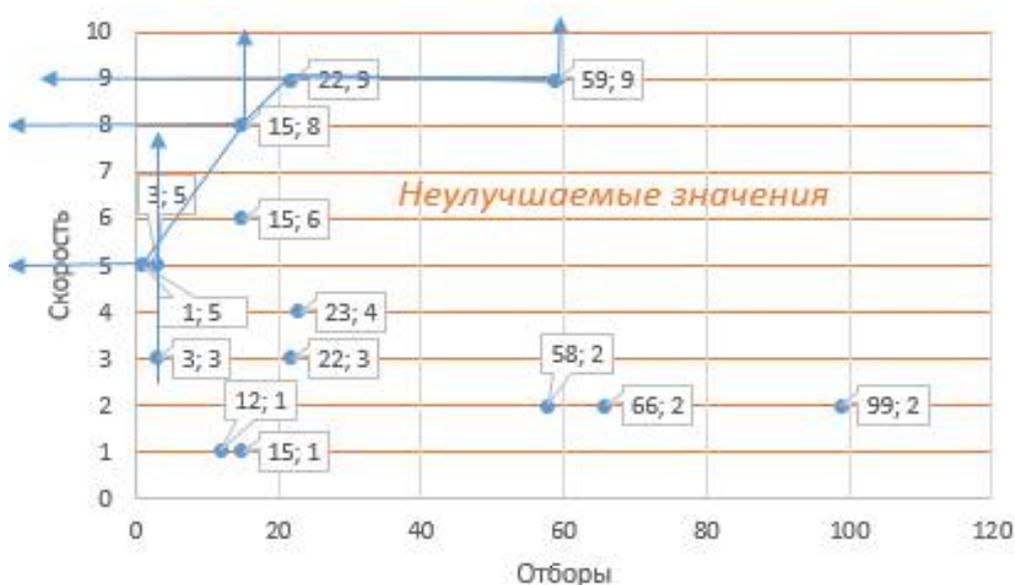


Рис. 6. Выбор кладовщиков по двум критериям

На основе рисунка 6 можно выделить четырех кладовщиков, скорость которых минимальная, а количество отборов наивысшее. Однако такой графический способ выбора не подходит, если необходимо учитывать все критерии, так как в n -мерном пространстве сложно оценивать характеристики показателей. Для решения этой проблемы используется линейная свертка, которая позволяет перейти от количества n критериев к одному. Так же для каждого критерия можно задать значимость, что даст более оптимальный вариант. Сведение к одному показателю происходит путем умножения показателя на значимость, после чего по строке идет суммирование полученных значений. Чтобы учесть характер критерия – стремиться к минимуму или максимуму – в процессе суммирования определяется знак «+» в случае для максимума и знак «-» в случае для минимума. На рисунке 7 показан результат применения такого способа.

	Значимость	0	0,1	0,2	0,3	0,4	0,5	0,6	0,7	0,8	0,9	1
		1	0,9	0,8	0,7	0,6	0,5	0,4	0,3	0,2	0,1	0
	Свертка											
1	6	5,25	4,5	3,75	3	2,25	1,5	0,75	0	-0,75	-1,5	0
2	4	3,37	2,74	2,11	1,48	0,85	0,22	-0,41	-1,04	-1,67	-2,3	0
3	2	1,22	0,44	-0,34	-1,12	-1,9	-2,68	-3,46	-4,24	-5,02	-5,8	0
4	9	7,88	6,76	5,64	4,52	3,4	2,28	1,16	0,04	-1,08	-2,2	0
5	2	0,81	-0,38	-1,57	-2,76	-3,95	-5,14	-6,33	-7,52	-8,71	-9,9	0
6	8	7,05	6,1	5,15	4,2	3,25	2,3	1,35	0,4	-0,55	-1,5	0
7	3	2,48	1,96	1,44	0,92	0,4	-0,12	-0,64	-1,16	-1,68	-2,2	0
8	2	1,14	0,28	-0,58	-1,44	-2,3	-3,16	-4,02	-4,88	-5,74	-6,6	0
9	1	0,75	0,5	0,25	0	-0,25	-0,5	-0,75	-1	-1,25	-1,5	0
10	5	4,49	3,98	3,47	2,96	2,45	1,94	1,43	0,92	0,41	-0,1	0
11	3	2,67	2,34	2,01	1,68	1,35	1,02	0,69	0,36	0,03	-0,3	0
12	9	7,51	6,02	4,53	3,04	1,55	0,06	-1,43	-2,92	-4,41	-5,9	0
13	1	0,78	0,56	0,34	0,12	-0,1	-0,32	-0,54	-0,76	-0,98	-1,2	0
14	5	4,47	3,94	3,41	2,88	2,35	1,82	1,29	0,76	0,23	-0,3	0
max	9	7,88	6,76	5,64	4,52	3,4	2,3	1,43	0,92	0,41	-0,1	0

Рис. 7. Выбор кладовщиков по семи критериям

Стоит отметить, что могут существовать такие значения критериев, которые не позволяют выбрать ПР для БП. Это граничные (или предельные) показатели, при которых теряется актуальность в таком выборе. Результат применения метода антиидеальной точки показан на рисунке 8. В таком случае определяется степени отличия показателя от предельного, и чем она выше, тем сильнее подходит ПР.

Антиидеальная		1	5	5	5	1	1	1		
		min	max	max	max	max	min	min		
ФИО	Отборов	Скорость	Опыт	Зона	ДК	Опозданий	Ошибок		мин	
1 Иванов	0,06667	1,2	1	0,2	0	0	1		0	
2 Петров	0,04348	0,8	0,4	0	0	0	1		0	
3 Сидоров	0,01724	0,4	0,2	0,2	0	0	0		0	
4 Самойлов	0,04545	1,8	0,4	0,4	0	0	0		0	
5 Лосев	0,0101	0,4	0,4	0,2	0	0	1		0	
6 Баранов	0,06667	1,6	0,6	0,6	0	0	1		0	
7 Смирнов	0,04545	0,6	0,8	1,6	0	0	1		0	
8 Горцев	0,01515	0,4	0,4	0,4	0	0	1		0	
9 Калинин	0,06667	0,2	0,4	1,4	0	0	1		0	
10 Петрухин	1	1	0,2	0,2	1	1	1		0,2	
11 Овечкин	0,33333	0,6	0,4	1	1	1	1		0,33333	
12 Ломов	0,01695	1,8	1	1,2	1	1	1		0,01695	
13 Перевалов	0,08333	0,2	0,2	0,2	1	1	1		0,08333	
14 Димитров	0,33333	1	0,6	0	0	0	1		0	
								МАКС	0,33333	

Рис. 8. Выбор кладовщиков на основе антиидеальной точки

Моделирование предполагает рассмотрение идеального течения БП, но следует учитывать, что пренебречь результатами многокритериальной оптимизации можно в случае нехватки ПР на складе.

В ходе исследования были применены описанные методы многокритериальной оптимизации и теории массового обслуживания, произведено моделирование обработки заявок для четырех моментов времени – утро, день, вечер, ночь. Результаты показывают, что длительность цикла обработки заказов, действительно можно сократить, используя данные средства – скорость обработки увеличилась, очереди сократились при использовании меньшего количества ПР.

Библиографический список

1. Математические методы и модели в коммерческой деятельности: учебник / Г.П. Фомин. – 3-е изд., перераб. и доп. – М.: Финансы и статистика; ИНФРА-М. – 2009. – 640 с.
2. Теория случайных процессов / А. В. Булинский, А. Н. Ширяев. М.: Физматлит, 2003. – 33 с.
3. Многокритериальные задачи принятия решений: Учебное пособие / А.В. Лотов, И.И. Поспелова – М.: МАКС Пресс, 2008. – 197 с.

УДК 004.043; ГРНТИ 20.51.01

ПРОЕКТИРОВАНИЕ ПОДСИСТЕМЫ АВТОМАТИЗИРОВАННОГО ТЕСТИРОВАНИЯ WEB-ПРИЛОЖЕНИЯ «АБОНЕНТ»

Д.В. Аникеев, А.В. Маркин, А.И. Петрова

*Рязанский государственный радиотехнический университет имени В.Ф. Уткина,
Российская Федерация, Рязань, getupnastya2000@gmail.com*

Аннотация. В работе рассматривается разработка подсистемы автоматизированного тестирования Web-клиента расчетно-аналитической системы «Абонент». Проводится анализ и выявление требований к подсистеме. Проанализированы существующие подходы к разработке информационных систем. Приводится разработанная архитектура подсистемы автоматизированного тестирования.

Ключевые слова: автоматизированное тестирование, подход к разработке через поведение (BDD), предметно-ориентированное проектирование монолит (DDD), поведенческое тестирование, локаатор, продукционная модель.

DESIGN OF THE SUBSYSTEM OF AUTOMATED TESTING OF THE WEB-APPLICATION «ABONENT»

D.V. Anikeev, A.V. Markin, A.I. Petrova

*Ryazan State Radio Engineering University named after V.F. Utkin,
Russia, Ryazan, getupnastya2000@gmail.com*

Annotation. The paper considers the development of a subsystem for automated testing of the Web-client of the settlement and analytical system "Subscriber". The analysis and identification of requirements for the subsystem is carried out. The existing approaches to the development of information systems are analyzed. The developed architecture of the automated testing subsystem is presented.

Keywords: automated testing, behavioral development approach (BDD), domain-specific monolith design (DDD), behavioral testing, locator, production model.

Цель работы

В современном мире перед компаниями в индустрии информационных технологий, стоит актуальная задача по снижению стоимости разработки программного обеспечения и улучшения качества выпускаемой продукции. Чтобы минимизировать затраченное время на процесс создания программного обеспечения (ПО), требуется своевременное обнаружение недостатков системы, для их устранения, тем самым ускорив процесс разработки и выпуска продукта. В свою очередь повысить качество продукта позволяет процесс тестирования.

Тестирование – неотъемлемый часть производства ПО, которая реализует контроль качества и позволяет проверить и оценить соответствие разработанного ПО требованиям к его функциональности.

Под тестированием понимается процесс, содержащий в себе все активности жизненного цикла, как динамические, так и статические, касающиеся планирования, подготовки и оценки программного продукта и связанных с этим результатов работ с целью определить, что продукт соответствует описанным требованиям, показать, что он подходит для заявленных целей, или выявить дефекты [1].

Целью настоящей работы является автоматизация процесса функционального тестирования и тестирования пользовательского интерфейса программного продукта.

Требования к подсистеме

Расчетно-аналитическая система «Абонент» входит в состав расчетно-платежного комплекса «Абонент+», который является полнофункциональным программным продуктом, обеспечивающим информационную поддержку, решающим расчетно-учетные задачи всех бизнес-процессов предприятия ЖКХ [2].

В свою очередь Web-клиент создан для удаленной работы с расчетной системой, предоставляющий полный доступ к информации об абонентах, позволяющий задать и изменить параметры расчета начислений, а также сформировать требуемый отчет по начислениям и оплатам за услуги ЖКХ [2].

В ходе анализа процесса тестирования данного продукта было выявлено, что тестирование программного средства производится вручную, данную работу производит один сотрудник. В деятельности тестирования не присутствует ведение документации по тестированию, т.е. нет разработанных тестовых случаев, сценариев, планов, по которым должно производиться тестирование ПО. Не формируется отчетность о проведенном тестировании.

Тестирование системы реализуется при внедрении проекта для новых заказчиков, а в случае отдельных доработок по заявкам, тестируются только вновь введенный или измененный функционал. Так как процесс тестирования не формализован, нет четкого плана действий, на основе которого должен выполняться процесс проверки, нет и гарантии выпуска качественного продукта.

Исходя из исследования, к разрабатываемой подсистеме автоматизированного тестирования выдвинуты следующие требования:

- возможность привлечения к разработке и сопровождению инженеров, не обладающих высокими компетенциями в области программирования;
- архитектура системы должна быть построена на простых и понятных всем участникам разработки подходах, методологиях;
- автоматизированные тесты должны проводить проверку, используя возможность работы с элементами Web-страницы, эмулируя действия пользователя;
- для обеспечения гибкости системы автоматизированного тестирования, она должна быть построена по модульному принципу;
- возможность работы с инструментами непрерывной интеграции, для автоматического запуска тестов;
- в результате работы системы должны формироваться обобщенные и детальные отчеты о проведенном тестировании.

Обзор существующих подходов и методологий

В настоящей работе для проектирования подсистемы автоматизированного тестирования рассмотрены такие возможные подходы к разработке, как методология разработки через поведение, предметно-ориентированное проектирование, методология поведенческого тестирования (тестирование «черного ящика»).

Подход к разработке через поведение. Разработка через поведение (BDD – Behavior-driven development) – это стратегия разработки программного обеспечения, являющаяся отвлечением от методологии разработки через тестирование (TDD – Test Driven Development). Основной идеей данного подхода является совмещение в процессе разработки технических интересов и интересов бизнеса, позволяя тем самым управляющему персоналу и программистам говорить на одном языке [3].

Для общения между группами персонала, участвующих в разработке ПО, используется предметно-ориентированный язык, основу которого представляют конструкции из естественного языка, понятные неспециалисту, обычно выражающие поведение программного продукта и ожидаемые результаты. Эти конструкции представляются в виде некой продукционной модели, которая позволяет представить знания о предметной области в виде предложений типа «Если (условие), то (действие)».

Продукционная модель — фрагменты модели предметной области, основанные на временных отношениях между состояниями объектов. В общем случае продукционную модель можно представить в следующем виде:

$i = \langle S; L; A \rightarrow B; Q \rangle$, где

S – описание класса ситуаций;

L – условие, при котором продукция активизируется;

$A \rightarrow B$ – ядро продукции;

Q – постусловие продукционного правила [4].

Методология разработки через поведение не предоставляет каких-либо формальных правил, но настаивает на том, чтобы использовался ограниченный стандартный набор фраз, который включал бы все элементы спецификации поведения.

К положительным качествам данного метода можно отнести:

- спецификация понятна всем участникам разработки системы;
- легко поддерживать изменения и доработки тестовых сценариев;
- тестовые сценарии не зависят от целевого языка программирования, и выбранных средств реализации ПО.

Не смотря на положительные характеристики подхода к разработке через поведение, он имеет и минусы. Такой подход не является методологией тестирования, тестовые сценарии не позволяют отразить модель предметной области, что не позволяет провести проверку ИС на соответствие концептуальной модели предметной области. Планирование и создание тестов происходит уже на этапе разработки тестируемой системы, при таком подходе затрачивается больше времени на разработку системы.

Предметно-ориентированное проектирование. Основоположником предметно-ориентированного проектирования является Эрик Эванс, который в 2003 году выпустил книгу под названием «Предметно-ориентированное проектирование».

Предметно-ориентированное проектирование (DDD – Domain Driven Design) – это набор принципов и схем, направленных на создание оптимальных систем объектов. Сводится к созданию программных абстракций, которые называются моделями предметных областей. В эти модели входит бизнес-логика, устанавливающая связь между реальными условиями предметной области и кодом [5].

Такой подход несколько отличается от традиционного подхода к разработке программ, который включает в себя ряд технических вопросов, связанных с обработкой механики, вводом и выводом данных, связью с сервером, базой данных и интеграцией с другими инструментами.

Одна из главных задач предметно-ориентированного проектирования – выстроить правильные языковые возможности, позволяющие всем участникам разработки: программистам, аналитикам, тестировщикам, заказчикам, общаться на едином языке. Для решения поставленной задачи, подход предполагает описание модели предметной области на предметно-ориентированном языке (DSL – Domain-specific language), который будет понят всем сотрудникам.

В данном подходе под модель понимается *концептуальная модель* предметной области. Компонентами такой модели являются объекты и взаимосвязи. Концептуальная модель служит средством общения между различными пользователями и поэтому разрабатывается без учета особенностей физического представления данных.

Концептуальная модель можно представить в виде абстрактной модели, определяющая структуру моделируемой системы, свойства её элементов и причинно-следственные связи, присущие системе и существенные для достижения цели моделирования.

Предметно-ориентированный язык – это язык программирования с ограниченными выразительными возможностями, ориентированный на некую конкретную предметную область. Различают внутренний и внешний объектно-ориентированный язык.

Внешний DSL представляет собой язык, отделенный от основного языка приложения, с которым он работает. Как правило, внешний DSL имеет пользовательский синтаксис, но при этом достаточно широко применяется синтаксис другого языка. Сценарий во внешнем

DSL, как правило, анализируется кодом принимающего приложения с помощью методов синтаксического анализа текста. Примерами внешнего DSL, с которыми можно столкнуться, являются регулярные выражения, SQL, Awk, а также XML - файлы конфигурации для систем наподобие Struts и Hibernate [5].

Внутренний DSL – это специфический способ использования языка общего назначения. Сценарий во внутреннем DSL представляет собой корректный код языка общего назначения, который использует только подмножество возможностей этого языка в определенном стиле, чтобы работать с одним небольшим аспектом всей системы. Результат должен иметь вид исходного текста пользовательского, а не базового языка. Классическим примером этого стиля является Lisp [5].

Предметно-ориентированное проектирование обладает следующими положительными характеристиками:

- концептуальная модель максимально точно отражает предметную область и входящие в нее бизнес-правила, бизнес-логику;
- легко читаемый код, понятный как для экспертов в предметной области, так и для разработчиков;
- независимость бизнес-логики от используемых инструментов в разработке.

Такой подход позволяет нивелировать отрицательные качества подхода к разработке через поведение.

Поведенческое тестирование или *тестирование «черного ящика»* — стратегия тестирования функционального поведения объекта (программы, системы) с точки зрения внешнего мира, при котором не используется знание о внутреннем устройстве (коде) тестируемого объекта. Иначе говоря, тестированием «чёрного ящика» занимаются тестировщики, не имеющие доступ к исходному коду приложения. Такая техника тестирования, основана на работе исключительно с внешними интерфейсами тестируемой системы. Под стратегией понимаются систематические методы отбора и создания тестов для тестового набора. Стратегия поведенческого теста исходит из технических требований и их спецификаций [6].

Понятие «чёрный ящик» предложил У. Р. Эшби. В кибернетике оно позволяет изучать поведение систем, то есть их реакций на разнообразные внешние воздействия и в то же время абстрагироваться от их внутреннего устройства [6].

На рисунке 1 представлена структурная схема поведенческого тестирования.

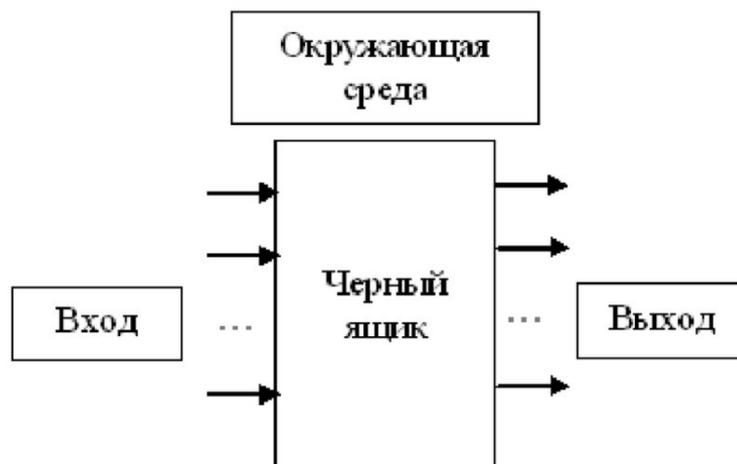


Рис. 1. Схема поведенческого тестирования

Исходя из приведённой схемы следует, что для проведения поведенческого тестирования необходимо выделить тестовые сценарии работы системы, определить какие данные

или действия будут являться входными параметрами ПО, а какие выходными, при этом следует описать правила зависимости выходных параметров от входных параметров.

Преимущества поведенческого тестирования:

- разработчик и тестировщик работают независимо друг от друга, что позволит проводить независимое тестирование;
- тестировщику не нужны знания каких-либо конкретных языков программирования, а значит можно привлечь к тестированию сотрудника не обладающих высокими компетенциями;
- тест выполняется с точки зрения пользователя, а не разработчика, что позволяет обнаружить несоответствия спецификациям;
- тестовые сценарии могут быть разработаны, как только будут разработаны спецификации и требования к разрабатываемой системе.

Архитектура подсистемы

На основе объединения положительных качеств каждого из рассмотренных подходов, спроектирована подсистема автоматизированного тестирования Web-клиента расчетно-аналитической системы «Абонент». Разработанная архитектура подсистемы представлена на рисунках 2-3.

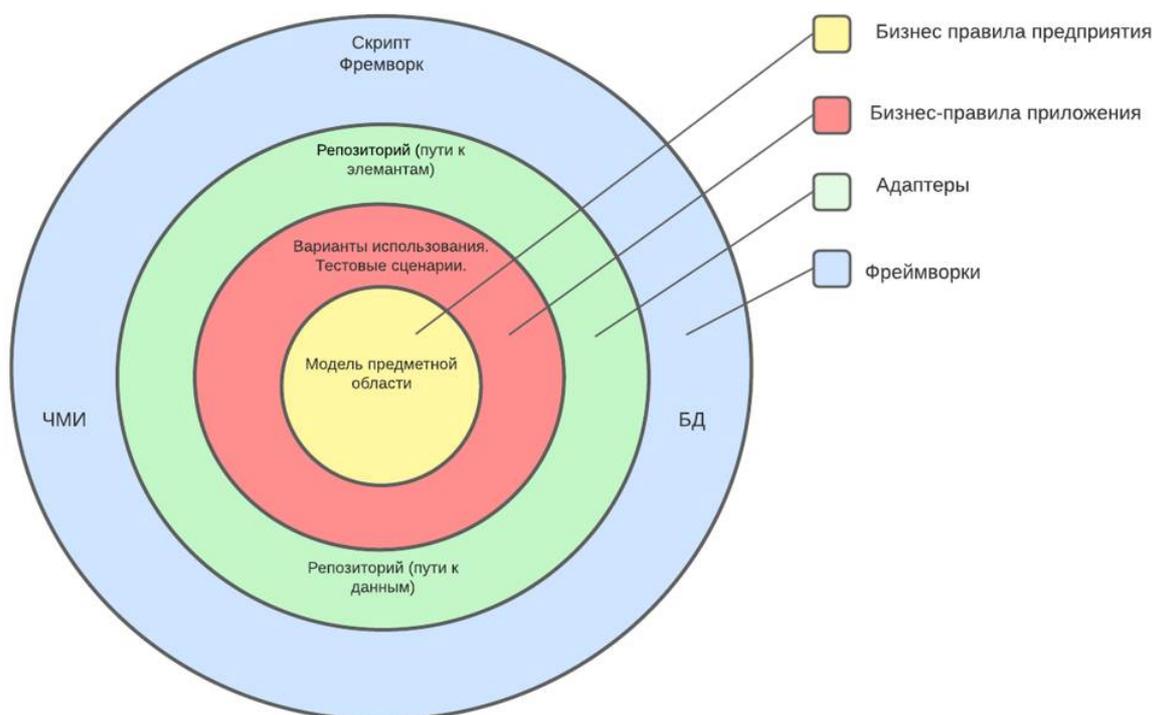


Рис. 2. Общая схема архитектуры разрабатываемой системы

Из графического представления архитектуры подсистемы автоматизированного тестирования, следует:

- подсистема разделена на 4 слоя;
- первым слоем – ядром подсистемы является модель предметной области;
- второй слой включает в себя описание поведения тестируемой системы – тестовые случаи;

- третий слой является неким адаптером между внешним слоем и двумя внутренними;
- конечный четвертый слой включает в себе программный продукт, который и производит тестирование, а также может содержать в себе базы данных и пользовательский интерфейс;
- каждый внутренний слой не зависит от внешнего для него слоя, при этом внешние слои могут зависеть от внутренних слоев, и могут обращаться к любому внутреннему слою.

На рисунке 3 представлен второй вариант схемы архитектуры, демонстрирующий взаимосвязь компонентов и состав слоев.

Для непосредственной разработки подсистемы автоматизированного тестирования требуется:

- первым этапом создать концептуальную модель предметной области на выбранном или разработанном самостоятельно объектно-ориентированном языке;
- на втором этапе, с помощью созданного скрипта на выбранном языке программирования, сгенерировать некий скелет файлов, в которых будут храниться локаторы (команды, уникально идентифицирующие элемент Web-страницы, сообщающие над какими элементами графического пользовательского интерфейса необходимо работать) и SQL запросы к БД;
- третьим шагом создается файл, содержащий в себе тестовые сценарии, описанные на языке предикатов;
- затем при запуске разработанного скрипта на языке программирования, получив на вход данные из файлов, где хранятся тестовые сценарии пути к элементам на странице, генерируется код тестов манипулирующие элементами страницы или же выполняющие проверку данных, используя запрос к БД;
- для выполнения автоматизированных тестов применяется, фреймворк специфичный для каждого языка программирования, позволяющий управлять запуском тестов и получать отчетность о проведении тестирования.

В результате, разработанное программное средство, принимая на вход модель предметной области, тестовые сценарии, локаторы и запросы, на выходе выполняет тестирование и предоставляет отчетность.

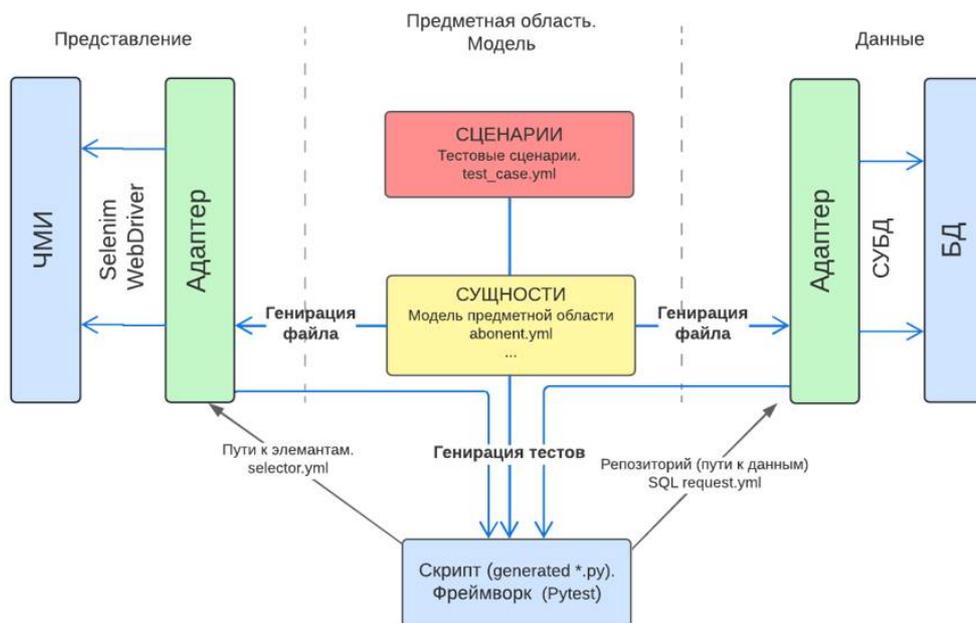


Рис. 3. Схема взаимосвязей компонентов

Апробация

Апробация решения производится на реальных задачах тестирования web-клиента расчетно-аналитической системы «Абонент» в сфере жилищно-коммунальных услуг, что позволяет оценить удобство и практическую применимость разрабатываемой подсистемы автоматизированного тестирования.

Библиографический список

1. ISTQB Glossary (International Software Testing Qualification Board) [Электронный ресурс]. - Режим доступа: <https://glossary.istqb.org/ru>, свободный (дата обращения 06.04.2022).
2. Расчетная система «Абонент» | ООО «Абонент+» [Электронный ресурс]. - Режим доступа: <https://www.abonent.plus/>, свободный (дата обращения 06.04.2022).
3. Карлос Солис, Сяофэн Ван. Исследование характеристик поведенческого развития // IEEE 2011 37th EUROMICRO Конференция: Программная инженерия и передовые приложения (SEAA), 2011 г.: сборник. – 2011. — стр. 383—387.
4. Продукционная модель представления знаний [Электронный ресурс]. – Режим доступа: <https://ru.wikipedia.org/wiki/>, свободный (дата обращения 06.04.2022)
5. Мартин Фаулер. Предметно-ориентированные языки программирования [Текст]/ Перевод с английского канд. техн. наук И.В. Красикова. – Москва, изд. Вильямс, 2011. – 215 с.
6. Бейзер Б. Тестирование чёрного ящика. Технологии функционального тестирования программного обеспечения и систем. [Текст]/ Бейзер Б. — Питер, 2004. — 320 с. — ISBN 5-94723-698-2.

СОДЕРЖАНИЕ

ИНФОРМАЦИЯ О V МЕЖДУНАРОДНОМ ФОРУМЕ «СОВРЕМЕННЫЕ ТЕХНОЛОГИИ В НАУКЕ И ОБРАЗОВАНИИ» СТНО-2022».....	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
Зайцев В.О. Разработка программного обеспечения для прокладки пеших маршрутов по заданным критериям.....	5
Каширин И.Ю. Интеллектуальный анализ данных в ретроспективной диагностике.....	11
Каширин И.Ю. Иерархические числа в проектировании общих онтологий.....	17
Антошкин В.А., Щербакова В.И. Использование javascript интерфейса прикладного программирования для управления NOSQL базой данных «POUCHDB».....	24
Горкин А.В. Анализ классов jdk. Подготовка исходных данных.....	31
Антошкин В.А., Иванов А.И. Использование библиотеки KERAS для обучения нейросетей выполнению простейших логических операций.....	34
Белов В.В., Перехода И.Ю. Метрическое пространство программного кода.....	41
Ципорков Н.И. Способы ускорения вычислений на примере преобразования Уолша.....	45
Головкин Н.В., Шурыгина О.В., Александров В.В., Цуканова Н.И. Иерархическая классификация текстов с помощью нейронных сетей.....	52
Бодров А.О. Использование технологий docker swarm и orbiter для реализации автомасштабирования Сервера.....	56
Белов В.В., Филоненко И.Н. Оценивание результатов обучения по индивидуальным учебным траекториям.....	11
Бобылева Е.В., Буланова И.А., Пылькин А.Н. Семантическая модель рабочей программы учебной дисциплины.....	61
Васильев М.Р., Кузнецова К.И., Шилкина М.В. Разработка пользовательского интерфейса ПО «Онлайн аукцион».....	65
Кибамба Ж. Ж., Пылькин А.Н. Мобильное приложение для размещения объявлений о товарах студентами.....	70

Серов А.П., Торжкова А.О. Обзор кластеризации данных с помощью графовых алгоритмов.....	74
Секция «ЭВМ И СИСТЕМЫ».....	78
Потапов Е.А., Шаповалов Н.В., Шишкин А.И. Разработка архитектуры веб-приложения «Онлайн аукцион».....	78
Чан Туан Зунг Анализ методов исправления ошибок в современных системах хранения данных.....	83
Чан Туан Зунг Эффективность применения символьных многопороговых декодеров в системах хранения данных.....	86
Черненко А.Д. Алгоритм вычисления дескриптора локальных бинарных шаблонов по трем ортогональным плоскостям для распознавания микровыражений лица.....	89
Муратов Е.Р., Епифанов А.С. Принцип работы алгоритма трекинга для задачи распознавания людей в транспортных средствах.....	95
Никифоров М.Б., Калинин Н.В. Реализация двухпроводного протокола передачи данных на VERILOG.....	99
Лаппо А.Ю. Стенд для построения устройств на микропроцессоре с использованием ПЛИС.....	103
Лучкина А.А., Тарасова В.Ю. Разработка тестового стенда для сравнения алгоритмов детектирования границ.....	105
Гринченко Н.Н., Вьюгина А.А., Баранова С.Н. Исследование битового трафика при передаче ландшафтных изображений.....	111
Бастрычкин А.С., Костров Б.В., Гринченко Н.Н., Костров Б.А. Формирование трафика в системах передачи данных.....	115
Иванов И.С., Рябинин А.В., Никифоров М.Б. Интернет вещей в интеллектуальной модернизации бытового газового оборудования.....	120
Кондрашова Т.И., Ефимов А.И. Исследование структуры репозитория унаследованной информационной системы ORACLE DESIGNER/2000.....	125
Пчелкин А.В., Никифоров М.Б. Сопровождение объектов с использованием фильтра MOSSE.....	130
Хруничев Р.В. Применение процедуры прокторинга при проведении вступительных испытаний в дистанционной форме в РГРТУ.....	136
Левитова А.В., Логинов А.А. Об актуальности совмещения реальных и виртуальных изображений в системах технического зрения летательных аппаратов.....	140

Конкин А.А., Костров Б.В., Никифоров М.Б., Никифоров М.С., Лынов Е.В. Совершенствование нормативно-методической документации по вопросу проверки эффективности экранирования экранированных сооружений.....	144
Панина И.С., Вьюгина А.А., Баранова С.Н. Описание концепции применения машинного обучения при обработке большого объема графической информации.....	150
Гринченко Н.Н., Крошила А.А. Использование инфографики во время приемной кампании в ВУЗе.....	154
Гринченко Н.Н., Крошила А.А. Многомерный анализ данных с использованием средств MS SQL SERVER и MS VISUAL STUDIO.....	158
Логинов Д.Ю., Хизриева Н.И. Построение простейшей симуляции жизнедеятельности организмов.....	162
Секция «ИНФОРМАЦИОННЫЕ СИСТЕМЫ И ТЕХНОЛОГИИ».....	166
Журкин Д.С., Дубинин А.В., Дубинин В.Н., Климкина Л.П. Обучение онтологии знаниям о стандарте IEC 61499.....	166
Кремез Я.Ю., Баталова М.А. Современные технологии в транспортной логистике.....	171
Касымов А.А. Современный подход организации процесса обработки информации в системах больших данных.....	176
Николов П. Взаимодействие клиентского и серверного программного обеспечения при исследовании метрологических характеристик солемеров.....	180
Евдокимова Е.Н., Устинов И.В. Обзор современных подходов к оценке рисков научно-образовательного комплекса.....	185
Пономаренко Д.В., Коваленко В.В. Формирование регламентов обеспечения безопасности для автоматизированного КПП общеобразовательных учреждений.....	191
Усик О.Н. Ускорение выполнения программного кода с использованием JUST IN TIME компилятора в языке PHP.....	195
Козлов А.И. Разработка комплексного подхода реабилитации постковидных состояний средствами адаптивной физической культуры и интернет-коммуникаций.....	199
Бойко М.К., Коваленко В.В. Разработка bsc-системы на основе методологии LEAN и процессного подхода для предприятия по разработке программных продуктов.....	203
Аюб Себах Аспекты интеллектуального анализа данных в бизнесе.....	208

Анашкин В.С., Челебаев С.В. Разработка нейросетевого аппроксиматора с использованием языка PYTHON.....	211
Маркин А.В., Баранов А.Ю. Разработка подсистемы поддержки пользователей.....	220
Булгаков Ю.А., Михеев А.А. Анализ особенностей применения сложных дискретных отсчетов для обработки Электрокардиосигнала.....	224
Конькова А.В., Челебаев С.В. Разработка нейросетевой подсистемы функционального преобразования частоты в код.....	229
Свиридова М.В., Челебаев С.В. Способы сокращения длительного цикла обработки заявок на складе канцетоваров.....	235
Аникеев Д.В., Маркин А.В., Петрова А.И. Проектирование подсистемы автоматизированого тестирования web-приложения «АБОНЕНТ».....	241

СОВРЕМЕННЫЕ ТЕХНОЛОГИИ В НАУКЕ И ОБРАЗОВАНИИ

Научное издание

В 10 томах

Том 4

Под общей редакцией О.В. Миловзорова.

Подписано в печать 15.06.22. 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