Иркутск, 2019

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РФ

ИРКУТСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ

ИНСТИТУТ СИСТЕМ ЭНЕРГЕТИКИ ИМ. Л.А. МЕЛЕНТЬЕВА СО РАН

ИНСТИТУТ ДИНАМИКИ СИСТЕМ И ТЕОРИИ УПРАВЛЕНИЯ СО РАН

ВИНЕРОВСКИЕ ЧТЕНИЯ
2019

Материалы Всероссийской молодежной научно-практической конференции

(Иркутск, 6–7 июня 2019 г.)

ИЗДАТЕЛЬСТВО
Иркутского национального исследовательского технического университета
2019

УДК 004 (08)
ББК 32.82.я5

Печатается по решению редакционно-издательского совета ИРНИТУ

Винеровские чтения – 2019 : мат-лы Всероссийской молодежной науч.-практ. конференции. – Иркутск : Изд-во ИРНИТУ, 2019. – 106 с.

В основе лежат материалы Всероссийской молодежной научно-практической конференции «Винеровские чтения 2019». Организаторами являются Иркутский национальный исследовательский технический университет (ИРНИТУ), Институт систем энергетики им. Л.А. Мелентьева (ИСЭМ) СО РАН и Институт динамики систем и теории управления (ИДСТУ) СО РАН. Конференция проведена при поддержке ИРНИТУ.

Рассмотрены методологические и практические вопросы разработки и применения современных информационных технологий для решения научных и практических проблем.

Ориентирован на научных и инженерно-технических работников, занимающихся вопросами создания современных информационных систем и решающих задачи анализа и обработки информации в различных проблемных областях, а также на преподавателей, аспирантов и студентов вузов, специализирующихся в области информационных технологий.

Ответственный редактор:
Проскуряков Д.П. (ИРНИТУ)

Редакторы:
Аршинский В.Л. (ИРНИТУ)
Бучнев О.С. (ИРНИТУ)
Серышева И.А. (ИРНИТУ)

Оформление обложки:
Григорьев С.В.

Печатается с файлов, предоставленных авторами

ISBN 978-5-8038-0910-4
ISBN 978-5-8038-1456-6

© ФГБОУ ВО «ИРНИТУ», 2019

СОДЕРЖАНИЕ

Математическое моделирование и технологии искусственного интеллекта

Применение алгоритмов решения проблемы булевой выполнимости к составлению оптимального расписания работы композиции вычислительных сервисов
Белей Е.Г. 7

Разработка системы политематической классификации на базе алгоритмов машинного обучения 12
Белых А.А.

Применение дубликатов конфликтных ограничений для ускорения работы алгоритмов решения проблемы булевой выполнимости
Кондратьев В.С. 19

Кластеризация точек евклидовой плоскости с ограничением по весу
Сиротинина Е.А., Маланова Т.В. 25

Информационные системы и технологии

Комплексные сети в стилометрии русских литературных текстов
Андреева А.М., Труфанов А.И. 32

Программно-инструментальная среда «ГрЛАО» для оценки эффективности работы организации
Аршинский В.Л., Хуен Ван Лэ 37

Метод формализации требований к входным данным
Камышова Е.А., Бахвалова З.А. 44

Разработка и исследование цифрового генератора случайных последовательностей 53
Копылов М.М.

Особенности реализации вероятностно-событийного поведения агентов-эвакуирующихся в комбинированной модели развития событий при пожаре
Куклин Е.В. 57

Разработка виртуального музея вычислительной техники ИРНИТУ на основе технологии дополненной реальности
Худякова Ю.Н., Куликова Л.Л. 63

Моделирование отражательных характеристик космического аппарата EGP в оптическом диапазоне
Шишкин П.И. 69

Веб-технологии и мобильные приложения

Поиск матрицы расстояний с использованием ГИС систем
Калугин М.В. 74

Архитектура веб-приложения для исследований интегрированных систем энергетики
Орехов Д.О., Барахтенко Е.А. 79

Информационные технологии в образовании и корпоративные информационные системы

«Паттерн автоматизации» поддержки решения организационно-распорядительных задач
Засядко А.А., Лукьянов Н.Д. 85

Разработка инструментального средства для анализа конфигураций 1С:Предприятие 92
Копылова В.В., Черкашин Е.А.

Разработка модуля проведения виртуальных аукционов для тренировочного стенда СТ-ИЭС ООО «Полюс-НТ» 100
Чекан М.А.

CONTENTS

Mathematical modeling and artificial intelligence technologies

Application of algorithms for solving Boolean satisfiability problem to formation of optimal schedule of computing services composition
Beley E.G. 7

Development of a multilabel classification system based on machine learning algorithms 12
Belykh A.A.

Application of duplicates of conflict constraints to speeding up operation of algorithms for solving the Boolean satisfiability problem
Kondratiev V.S. 19

Clustering of points of the Euclidean plane with a limit on the weight
Sirotinina E.A., Malanova T.V. 25

Information systems and technologies

Complex networks in stylometry of Russian literary texts
Andreeva A.M., Trufanov A.I. 32

Software tool "GRLAO" for evaluating effectiveness of organizations
Arshinsky V.L., Le Van Huyen 37

Formalization method for input data requirements
Kamyshova E.A., Bahvalova Z.A. 44

Development and research of digital generator of random sequences
Kopylov M.M. 53

Implementation features of probabilistic-event behavior of evacuating agents in the combined model of the course of fire 57
Kuklin E.V.

Development of a computer technology virtual museum INRTU based on augmented reality technology 63
Hudyakova U.N., Kulikova L.L.

Simulation of the reflective characteristics of EGP satellite in the optical range
Shishkin P.I. 69

Web-technologies and mobile applications

Determining distance matrix using GIS systems
Kalugin M.V. 74

Architecture of web application for research of integrated energy systems
Orehov D.O., Barakhtenko E.A. 79

Information technologies in education and enterprise information systems

«Automation pattern» of the support of solving organizational and administrative tasks
Zasyadko A.A., Lukyanov N.D. 85

Development of a configuration tool for analysis of 1C:Enterprise
Kopylova V.V., Cherkashin E.A. 92

Development of a virtual auctions module for «ST-IES» stand of POLUS-NT
Chekan M.A. 100

УДК 519.7

ПРИМЕНЕНИЕ АЛГОРИТМОВ РЕШЕНИЯ ПРОБЛЕМЫ БУЛЕВОЙ ВЫПОЛНИМОСТИ К СОСТАВЛЕНИЮ ОПТИМАЛЬНОГО РАСПИСАНИЯ РАБОТЫ КОМПОЗИЦИИ ВЫЧИСЛИТЕЛЬНЫХ СЕРВИСОВ

Белей Е.Г.
(г. Иркутск, Иркутский национальный исследовательский технический университет)
e-mail: zhenyha28@mail.ru

Аннотация. В данной работе рассматривается задача построения оптимального плана выполнения композиции вычислительных сервисов. Эта задача рассматривается как частный случай проблемы составления цехового расписания (job-shop scheduling problem). Последняя сводится к задаче о булевой выполнимости (SAT), для вычислительного решения которой используются современные SAT-решатели. Для одной индивидуальной задачи планирования работы композиции вычислительных сервисов удалось построить оптимальный план, который оказался лучше описанного в литературе решения данной задачи.

Ключевые слова. Задача составления плана выполнения композиции сервисов; задача выполнимости булевых формул; SAT; задача цехового планирования; теория расписаний.

APPLICATION OF ALGORITHMS FOR SOLVING BOOLEAN SATISFIABILITY PROBLEM TO FORMATION OF OPTIMAL SCHEDULE OF COMPUTING SERVICES COMPOSITION

Beley E.G.
(Irkutsk, Irkutsk National Research Technical University)
e-mail: zhenyha28@mail.ru

Abstract. In this paper we consider the problem of constructing an optimal plan for the composition of computing services. This task is considered as a special case of job-shop scheduling problem. It reduces to Boolean satisfiability problem for computational solution of which modern SAT solvers are used. For one individual task of planning the composition of computing services it was possible to build an optimal plan that turned out to be better than the solution to this problem described in the literature.

Keywords. The task of constructing a plan for the composition of computing services; Boolean satisfiability problem; SAT; job-shop scheduling problem; theory of scheduling.

Введение. В последние годы активно развивается сервис-ориентированная архитектура вычислений (Service-Oriented Architecture, SOA) [1], которая предполагает представление программных продуктов в виде сервисов и их распределенную обработку в вычислительных средах. Под сервисом мы понимаем некоторою услугу, цель которой удовлетворить запросы пользователя [2]. Веб-сервис – это автономный вычислительный ресурс, предоставляющий свои функции через сеть Интернет (или Интранет) посредством открытых протоколов обмена данными, не зависящих от платформ, как самих ресурсов, так и связываемых с ними программных систем [3]. Процесс объединения веб-сервисов для обеспечения решения какой-либо задачи называется композицией веб-сервисов.

Целью настоящей работы является решение задачи планирования композиции сервисов путем использования алгоритмов решения проблемы булевой выполнимости (SAT) [5]. Несмотря на NP-полноту SAT, современные алгоритмы ее решения успешно применяются к задачам из различных областей, например, в символьной верификации, биоинформатике, теории расписаний, криптоанализе и многих других. Это становится возможным благодаря тому, что многие задачи из перечисленных областей допускают эффективную сводимость к SAT. Данный процесс называется пропозициональным кодированием.

Задача выполнимости булевых формул, сокращенно обозначаемая как SAT, заключается в следующем: можно ли назначить всем переменным, встречающимся в формуле, значения «ложь» и «истина» так, чтобы формула приняла значение «истина». На практике SAT обычно рассматривается относительно формул в конъюнктивной нормальной форме (КНФ). Для решения SAT-задач используются программы, называемые SAT-решателями (SAT-solvers).

Задача планирования выполнения композиции сервисов. В целом ряде работ задача составления плана выполнения композиции сервисов ставится в форме одной из задач теории расписаний, а именно, задачи планирования цехового расписания (job-shop scheduling problem) (впервые описана в работе [4]).

Пусть существует некоторый набор задач (сервисов) $\{1, \dots, I\}$. Для каждой задачи $i \in \{1, \dots, I\}$ определены следующие параметры:

- Стоимость: $cost(i)$ - сколько по времени будет выполняться эта задача. Сумма стоимостей определяет максимальное время решения T для этих задач.
- Зависимости: $dependencies(i) = \{\alpha_1, \dots, \alpha_{n_i}\}$, $\alpha_k \in \{1, \dots, I\} \backslash \{i\}$ – список задач, которые должны быть выполнены перед тем, как может быть запущена задача i.

Если зависимостей нет – полагаем, что задача может быть запущена в любое время.

Далее для каждой задачи будем рассматривать также временной параметр $t \in \{1, \dots, T\}$.

Есть список узлов, на которых могут быть выполнены эти задачи: $j \in \{1, \dots, J\}$.

Требуется найти минимальное время $t_{min} \leq T$, за которое будут выполнены все задачи.

Схема пропозиционального кодирования задачи «job-shop». Опишем схему пропозиционального кодирования задачи «job-shop».

Вводим два множества булевых переменных: множество $X = \{x_{ijt}\}$ и множество $Y = \{y_{ijt}\}$, где переменные x_{ijt} кодируют тот факт, что задача i попала на узел j в момент t, а переменные y_{ijt} кодируют тот факт, что задача i была решена на узле j в момент t.

Представим все условия задачи в виде булевых формул, от которых впоследствии можно перейти к КНФ.

Необходимо задать продолжительность выполнения каждой задачи. Для этого свяжем переменные из множества X с переменными множества Y. Например, если задача i решается на узле j $cost(i)$ моментов времени, то имеем следующую логическую эквивалентность:

$$x_{ijt} \equiv y_{ij(t+cost(i)-1)}. \quad (1)$$

Для приводимых далее условий будем использовать т.н. «ООС-предикат» [6], который задается следующей формулой:

$$OOC\{p_1, p_2, \dots, p_n\} = (\bigvee_{k=1}^{n} p_i) \wedge \left(\bigwedge_{1 \le k < l \le m} (\neg p_i \vee \neg p_j)\right).$$

Данный предикат принимает значение «истина» тогда и только тогда, когда единственная переменная из $\{p_1, p_2, \dots, p_n\}$ принимает значение «истина».

На переменные x_{ijt} при помощи ООС-предиката мы накладываем условие, что каждая задача была взята некоторым узлом в некоторый момент времени лишь однажды. То есть для каждой задачи i должны быть истинны следующие предикаты:

$$OOC_{j \in \{1,\dots,J\}, t \in \{1,\dots,T\}}\{x_{ijt}\}, \quad (2)$$

$$OOC_{j \in \{1,\dots,J\}, t \in \{1,\dots,T\}}\{y_{ijt}\}. \quad (3)$$

Данные ограничения гарантируют (учитывая (1)), что если x_{ijt} – истинна, то это означает истинность $y_{ij(t+cost(i)-1)}$ и ложность переменных $x_{ijk}, k \in \{1, \dots, T\} \backslash t$, и $y_{ijl}, l \in \{1, \dots, T\} \backslash \{t + cost(i) - 1\}$.

Также необходимо задать условие, что если задача i взята на узле j в момент времени t, то $cost(i)$ моментов времени на этом узле не может быть решена никакая другая задача:

$$\bigwedge_{wr} \left(x_{ijt} \to \neg x_{wj(t+r)}\right),$$

где $w \in \{1, \dots, I\} \backslash \{i\}$, $r \in \{0, \dots, cost(i) - 1\}$.

Тот факт, что на узле j в момент времени t может быть только одна задача или не быть задач вообще, кодируется следующей формулой:

$$AMO_{i \in \{1,\dots,I\}, t \in \{1,\dots,T\}}\{x_{ijt}\}.$$

Предикат AMO [7] принимает значение «истина» тогда и только тогда, когда среди набора переменных $\{p_1, p_2, \dots, p_n\}$ максимум одна принимает значение «истина». Он задается следующей формулой $AMO\{p_1, p_2, \dots, p_n\} = \bigwedge_{1 \le k < l \le m} (\neg p_k \vee \neg p_l)$.

Условия на зависимости между задачами кодируются следующим образом. Переменные множества X конкретной задачи связываются с переменными множества Y тех задач, от которых зависят. То есть тот факт, что x_{ijt} принимает значение «истина», означает, что в один из предшествующих t моментов были решены соответствующие задачи из списка зависимостей для задачи i. То есть были истинны соответствующие переменные множества Y. Соответствующие ограничения будут иметь вид:

$$x_{ijt} \to \bigwedge_a \bigvee_{bc} y_{abc},$$

где a «пробегает» $dependencies(i) = \{\alpha_1, \dots, \alpha_{k_i}\}$, $\alpha_j \in \{1, \dots, I\} \backslash \{i\}$ – список задач, которые должны быть выполнены перед задачей i, $b \in \{1, \dots, J\}$, $c \in \{1, \dots, t - 1\}$.

Перейдем к описанию построения целевой функции. Вспоминаем, что y_{ijt} принимает значение 1 тогда и только тогда, когда задача с номером i решена на узле с номером j в момент времени t. Если задача решена в конкретный момент t, то она не может быть решена ни в один из предшествующих и ни в один из последующих моментов (эти условия обеспечиваются ООС-предикатами).

Для каждой задачи заводим дополнительные переменные, которые будут кодировать число, выражающее момент времени, в который она была решена. Допустим, всего существует T моментов времени. Для кодирования данного числа потребуется $q = \lfloor \log T \rfloor +$

1 переменных, обозначим их $z_1, \dots, z_q$. Натуральное число, двоичным представлением которого является конкретный набор значений переменных $z_1, \dots, z_q$, обозначим через z.

Рассмотрим произвольную задачу $i \in \{1, \dots, I\}$. Предположим (в упрощенном варианте), что данная задача может быть решена в любой момент времени на любом узле (на самом деле для начальной стадии решения это не так – некоторые задачи не могут быть решены в первые моменты времени из-за иерархии зависимостей). Свяжем с задачей i q булевых переменных: $v_1^i, \dots, v_q^i$. Пусть τ – произвольный момент времени $\tau \in \{1, \dots, T\}$ и пусть $\left(\alpha_1(\tau), \dots, \alpha_q(\tau)\right)$ вектор двоичного представления числа τ. Предположим, что i решена на узле j в момент времени τ. Тогда истинна следующая формула:

$$y_{ij\tau} \to l_{\alpha_1(\tau)}\left(v_1^i\right) \wedge \dots \wedge l_{\alpha_{q-1}(\tau)}\left(v_{q-1}^i\right) \wedge l_{\alpha_q(\tau)}\left(v_q^i\right).$$

В последней формуле используется обозначение $l_\beta(x)$, где x - булева переменная: $l_\beta(x)$ означает литерал $\neg x$ при $\beta = 0$, и литерал x при $\beta = 1$.

Пусть v^i– натуральное число, заданное набором значений переменных $v_1^i, \dots, v_q^i$. Тогда тот факт, что все задачи решились за время, выраженное конкретным числом z (в форме вектора двоичного представления), равносилен истинности следующего предиката:

$$\bigwedge_{i \in \{1, \dots, I\}} \left(v^i \le z\right).$$

Предикат «$\alpha \le \beta$» для произвольных натуральных чисел α и β допускает эффективное пропозициональное кодирование (например, [8]).

Вычислительный эксперимент. В ходе вычислительного эксперимента рассматривалась задача, поставленная в работе [9].

Для выполнения определенного сервиса (1017) предварительно должны быть выполнены сервисы 1001-1016. Необходимо найти наименьшее время выполнения этих сервисов на 4 узлах, если:

1. Время выполнения сервиса 1001 – 25 секунд;
2. Время выполнения сервисов 1002, 1005, 1008, 1011, 1014 – 20 секунд;
3. Время выполнения сервисов 1003, 1006, 1009, 1012, 1015 – 7 секунд;
4. Время выполнения сервисов 1004, 1007, 1010, 1013, 1016 – 27 секунд.

Зависимости задач показаны на рисунке 1.

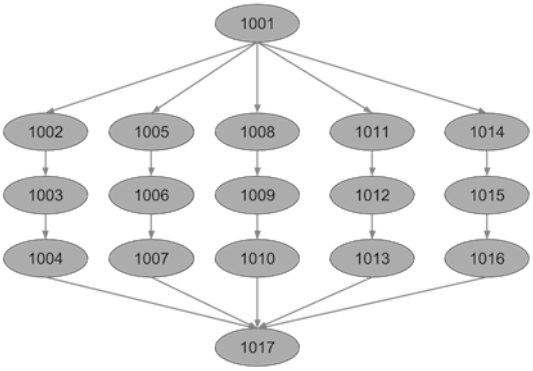


Рис. 1. Направленный ациклический граф зависимости заданий по данным

В работе [9] осуществлялся поиск времени решения, приближенного к оптимальному, с помощью модификации спискового эвристического алгоритма составления расписания

HEFT (Heterogeneous Earliest Finish Time), реализующего метод поиска в глубину. Найденное время составило 114 секунд.

Обозначим через КНФ(t) КНФ, кодирующую задачу построения плана выполнения композиции сервисов за время t. Целью настоящей работы было для приведенных выше исходных данных найти точное минимальное время t_{min} такое, что для $\forall\, t \geq t_{min}$ КНФ(t) – выполнима, а для $\forall\, t < t_{min}$ КНФ(t) – невыполнима.

Для этого использовалась итеративная процедура. На первом этапе происходила генерация файла, содержащего основную КНФ, кодирующую процесс выполнения сервисов для времени T. Затем в этот файл добавлялась КНФ, кодирующая тот факт, что время решения всех задач не превосходит времени t. Для изменения значения параметра t использовалась дихотомическая процедура [8].

С помощью описанной схемы удалось найти минимальное время решения, равное 106 секундам. Также удалось доказать, что плана с меньшим временем решения не существует. Вычисления проводились с помощью SAT-решателя cryptominisat 5.0 [10] на обычном ПК с процессором Intel Core i5. План выполнения композиции сервисов приведен на рисунке 2.

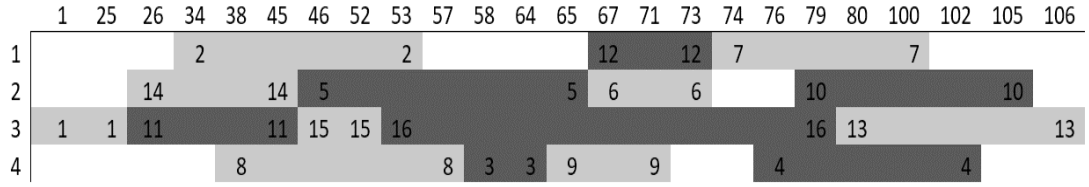


Рис. 2 План выполнения композиции сервисов рассматриваемой задачи

Заключение. В статье приводятся результаты применения алгоритмов решения проблемы булевой выполнимости (SAT) к задаче составления плана выполнения композиции вычислительных сервисов. Была построена пропозициональная кодировка данной задачи и описана итеративная схема решения соответствующей серии SAT-задач. Для известного примера задачи планирования выполнения композиции вычислительных сервисов удалось улучшить имеющийся план и доказать оптимальность нового плана.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. Emig C., Weisser J., Abeck S. Development of SOA-based software systems — anevolutionary programming approach // Advanced Intern. Conf. on Telecommunications /Internet and Web Applications and Services. Guadeloupe, French Caribbean, February 2006.P. 182–195.

2. Новицкий А. В. Подход к построению сервис-ориентированной электронной библиотеки в среде SemanticWeb на основе репланирования. // Труды конференции RCDL'2008. – С. 392–396.

3. Душкин Д. Н. Анализ чувствительности веб-сервисов в задаче выбора оптимальной конфигурации систем с сервисно-ориентированной архитектурой //УБС, 40 (2012), 164–182

4. Graham R. (1966). "Bounds for certain multiprocessing anomalies". // Bell System Technical Journal. 45 (9): 1563–1581

5. Cook Stephen A. (1971). "The Complexity of Theorem-Proving Procedures". // Proceedings of the 3rd Annual ACM Symposium on Theory of Computing: 151–158.

6. Asin R., Nieuwenhuis R., Oliveras A., Rodriguez-Carbonell E. Cardinality networks: a theoretical and empirical study // Constraints. 2011. Vol. 16. Iss. Pp. 195–221.

7. Steffen Hölldobler and Van Hau Nguyen. An Efficient Encoding of the At-Most-One Constraint.//Technical report, Knowledge Representation and Reasoning Group 2013-04, Technische Universität Dresden, 01062 Dresden, Germany, 2013.

8. Заикин О. С., Отпущенников И. В., Семёнов А. А. Параллельные алгоритмы решения проблемы выполнимости в применении к оптимизационным задачам с булевыми ограничениями // Выч. мет. программирование, 12:1 (2011), 205–212

9. Федоров Р.К., Шумилов А.С. Сценарий расчета временной доступности объектов образования // Вестник Бурятского гос. ун-та. — 2017. — Т. 2. — С. 20–32.

10. Soos M., Nohl K., Castelluccia C.: Extending SAT Solvers to Cryptographic Problems. In: Kullmann, O. (ed.) SAT. Lecture Notes in Computer Science, vol. 5584, pp. 244–257. Springer (2009)

УДК 004.855.5

РАЗРАБОТКА СИСТЕМЫ ПОЛИТЕМАТИЧЕСКОЙ КЛАССИФИКАЦИИ НА БАЗЕ АЛГОРИТМОВ МАШИННОГО ОБУЧЕНИЯ

Белых А.А.
(г. Иркутск, Иркутский национальный исследовательский технический университет)
e-mail: Niuta.Belykh@yandex.ru

Аннотация. В данной статье рассматриваются различные методы машинного обучения для решения задачи политематической классификации текстовых документов. Также проводится их сравнительный анализ по некоторым метрикам.

Ключевые слова. Машинное обучение, политематическая классификация, обработка естественного языка

DEVELOPMENT OF A MULTILABEL CLASSIFICATION SYSTEM BASED ON MACHINE LEARNING ALGORITHMS

Belykh A.A.
(Irkutsk, Irkutsk National Research Technical University, Irkutsk,)
e-mail: Niuta.Belykh@yandex.ru

Abstract. The article discusses various methods of machine learning for solving the problem of multilabel classification of text documents. Their comparative analysis on some metrics is carried out.

Keywords. Machine learning, multilabel classification, natural language processing.

Введение. В рамках поддержки продуктов компания ISPsystem постоянно обслуживает обращения от пользователей. Нередко возникают ситуации, когда проблема (или вопрос) одного пользователя уже возникала ранее и в базе имеется ее решение.

Вся необходимая информация представлена в виде тикетов. Каждый тикет характеризуется кодом, датой получения запроса, темой, количеством сообщений от клиента и количеством сообщений от сотрудника, тегами и непосредственно самим сообщением. Так как продукты компании используются за пределами России, заявка может быть как на русском, так и на английском языке.

Теги рассматриваются как ключевые слова, относящиеся к некоторой части запроса пользователя. Они определяют, к какому отделу относится тикет или общую проблему. Если это сообщение является ответом от сервиса, то указывается тег, идентифицирующий данный тикет, как автоматический (не требует ответа или других действий). Также, если в тикете содержится информация о проблеме пользователя, тег должен указывать версию продукта, в которой данная проблема возникла (например, lite, advantage). Тег может характеризовать сложность проблемы – влияет на то, какой квалификацией должен обладать сотрудник для решения данной проблемы. Тег может указывать тип клиента и так далее.

Тегирование позволяет разделять тикеты так, чтобы пользователю оказали требуемую поддержку в самый короткий срок. Благодаря данному механизму обеспечивается удобный поиск и фильтрация тикетов. Другой немаловажной применимостью тегов является оценка качества как продукта, так и документации, руководств пользователей. Чем меньше сообщений адресовано данному отделу, тем меньше проблем возникает у пользователей.

В компании хранятся накопленные за несколько лет тикеты с тегами, расставленными сотрудниками, то есть, имеется некоторый набор данных. Каждый тикет может быть помечен одним тегом, несколькими или быть не отмеченным вовсе. Так как необходимо сопоставлять тикеты с нулем или более тегов, заданных заранее некоторым набором – данная задача является задачей политематической классификации.

Решение такой задачи с применением алгоритмов машинного обучения является реальной возможностью улучшить процесс диспетчеризации.

Целью данного исследования является сравнительный анализ алгоритмов машинного обучения для политематической классификации текстовых данных.

Политематическую классификацию формально можно записать в следующем виде. Пусть X – множество объектов, $\Lambda = \{\lambda_1, \lambda_2, \dots, \lambda_k\}$ – набор меток (классов). $x_i = (f_1, f_2, \dots, f_m) \in X$, где m – количество признаков объекта, ассоциирован с определенным набором меток $L_i \in 2^{\Lambda}$. Можно сопоставить множество соответствующих меток L_i с двоичным вектором $Y_i = (l_1, l_2, \dots, l_k)$, где $l_i = 1 \Longleftrightarrow \lambda_i \in L_i$. Тогда множество всех возможных маркировок $Y^* = \{0,1\}^k$ [1].

Данная задача является задачей обучения с учителем (supervised learning).

Обработка данных. Машина не способна работать с текстовыми данными в удобном для человека виде, поэтому тикеты нуждаются в предобработке. Во-первых, необходимо выделить признаки, характеризующие объект. В данном случае, тема и сообщение запроса являются входными данными, а на выходе ожидается набор тегов, соответствующих данному тикету.

Для перевода текста в машинное представление был разработан специфический конвейер, основанный на методе «мешка слов», выполняющий последовательно следующие процессы:

1. Токенизация (tokenization). Разбиение каждого документа на n-граммы (где $n = 1, 2$), которые встречаются в нем (токены), с помощью пробелов, знаков пунктуации и других специальных символов. Все токены переводятся в нижний регистр.

Пары токенов называют биграммами (bigrams), тройки токенов известны как триграммы (trigrams), а в более широком смысле последовательности токенов известны как

n-граммы (n-grams). Использование n-грамм позволяет учитывать смысловую нагрузку фразы, зависящую от порядка слов [2].

Для избавления от неинформативных для темы слов используется подход на основе словаря – метод стоп-слов. Базируясь на специализированных пакетах, были исключены стоп-слова английского и русского языков.

Среди встретившихся токенов могут быть разные формы одного и того же слова. Например, разные глагольные формы или единственное и множественное число. Во избежание такого дублирования применяется лемманизация (lemmatization). Это процесс определения начальной формы слова на основе словаря с учетом его роли в предложении (то есть принимается во внимание, к какой части речи относится слово). Стандартизированная форма слова называется леммой (lemma). В итоге происходит идентификация или объединение (conflating) всех слов с одной и той же основой.

В качестве альтернативы стеммингу может быть использован и другой способ нормализации – стемминг (stemming). Это процесс обрезки слова до его основы (word stem), который выполняется с помощью эвристик на основе правил (например, удаление общих суффиксов), а не заранее заданных форм слов.

2. Построение словаря (vocabulary building). Сбор словаря всех токенов, которые появляются в любом из документов, и нумерация их. С целью уменьшения словаря, в него заносятся только те слова, которые встретились в двух и более документах.

3. Создание разреженной матрицы (sparse matrix encoding, используется SciPy). Для каждого документа подсчитывается, как часто каждое из слов, занесенное в словарь, встречается в документе.

На начальном этапе данного процесса вместо исключения несущественных признаков применяется их масштабирование. Одним из наиболее распространенных способов такого масштабирования является метод частота термина - обратная частота документа (term frequency - inverse document frequency, tf_idf). Идея этого метода заключается в том, чтобы присвоить большой вес термину, который часто встречается в конкретном документе, но при этом редко встречается в остальных документах корпуса. Если слово часто появляется в конкретном документе, но при этом редко встречается в остальных документах, оно, с большой вероятностью, будет описывать содержимое этого документа лучше.

Для преобразования tf_idf существуют различные варианты взвешивания частот. Значение tf_idf для слова w в документе d вычисляется по формуле:

$$tf_idf(w,d) = tf\,\frac{log(N+1)}{N_w+1} + 1,$$

где N – это количество документов в обучающем наборе, N_w – это количество документов обучающего набора, в которых встретилось слово w, и tf (частота термина) – это частота встречаемости термина в запрашиваемом документе d (документе, который подлежит преобразованию). Кроме того, оба класса применяют L2 нормализацию после того, как вычисляют представление tf_idf. Другими словами, они масштабируют векторизированное представление каждого документа к единичной евклидовой норме (длине). Подобное масштабирование означает, что длина документа (количество слов) не меняет его векторизованное представление [2].

И, наконец, для увеличения обобщающей способности будущей модели, производится отбор лучших k признаков (select k-best features), основываясь на важности признака

(насколько каждый токен влияет на предсказания меток) по функции f_classif (статистическая функция) [3].

Набор тегов также необходимо преобразовать в векторное представление. Однако так как теги являются категориальными признаками, необходимо и достаточно представить в виде последовательных чисел (номеров), а наборы тегов для каждого тикета – в виде массивов, хранящих единичные значения в позициях, соответствующих номерам используемых тегов, и нулевые в позициях остальных тегов.

Применяемые методы. В качестве алгоритмов машинного обучения были рассмотрены методы различных направлений – классическое обучение, ансамблевые методы и нейронные сети.

Классическое обучение строится на статистических алгоритмах и решает вопросы, связанные с принятием решений на основе данных (формальные задачи).

1. Метод опорных векторов (SVM, support vector machine) является самым популярным методом классической классификации. Алгоритмическая парадигма SVM решает проблему сложности выборки путем поиска разделяющей гиперплоскости с максимальным запасом (зазором). Метод применяется как для линейно разделимых наборов данных с использованием классических алгоритмов, так и для линейно не разделимых данных. [4]. Был использован «мягкий» (foft-SVM) вариант метода, основанный на ядрах (kernel), где в качестве ядра использовалась радиальная базисная функция (rbf, гауссовское ядро).

2. Логистическая регрессия (logistic regression, logit regression) является алгоритмом линейной классификации (а не алгоритмом регрессии) и применяется для прогнозирования вероятности возникновения некоторого события путем подстановки данных к логистической кривой (сигмоиде) по значениям множества признаков.

В общем случае объект x можно отнести к классу $y = 1$, если предсказанная моделью вероятность $P\{y = 1 \vee x\} > 0.5$, и к классу $y = 0$ в противном случае. На практике метод используется с регуляризацией (добавление некоторых дополнительных ограничений к условию с целью предотвратить переобучение) штрафной функции (penalty function) [5].

Достаточно часто применяется регуляризация эластичной сетки (Elastic net regularization) – полиномиальная логистическая регрессия, линейно объединяющая параметры l1, l2 методов LASSO (least absolute shrinkage and selection operator, оператор абсолютной наименьшей усадки и выбора) – улучшенный метод наименьших квадратов, и Ridge (метод регуляризации Тихонова) – алгоритм приближённого решения некорректно поставленных операторных задач, соответственно в некотором соотношении [6].

3. Метод стохастического градиента. Стохастический градиентный спуск (SGD, Stochastic Gradient Descent) – простой, но очень эффективный подход к дискриминационному обучению линейных классификаторов при использовании выпуклых функций потерь, таких как линейные SVM и логистическая регрессия.

При стохастическом градиентном спуске в отличие от классического метода градиентного спуска не требуется, чтобы обновление направления спуска было точно основано на градиенте. Вместо этого направление выбирается случайным вектором и его ожидаемое значение на каждой итерации должно совпадать с направлением градиента. Или, в более общем смысле, необходимо, чтобы ожидаемое значение случайного вектора было

субградиентом функции в текущем векторе – вектор для не дифференцируемой функции. Таким образом, вектор w настраивается на каждый вновь выбираемый объект [2].

4. Метод ближайших соседей. Идея метода k-ближайших соседей (kNN, k-nearest neighbours) состоит в том, чтобы запомнить обучающий набор и затем предсказать метку любого нового экземпляра на основе меток его ближайших соседей в обучающем наборе. Обоснование такого метода основывается на предположении, что функции, используемые для описания доменных точек, имеют отношение к их маркировке таким образом, чтобы точки, находящиеся рядом, вероятно, имели одинаковую метку [5].

Кроме того, в некоторых ситуациях, даже если обучающий набор содержит большое количество данных, поиск ближайшего соседа может быть выполнен очень быстро (например, когда обучающий набор представляет собой всю сеть, а расстояния основаны на ссылках).

5. Деревья решений (decision tree) – это непараметрический метод обучения, используемый для классификации и регрессии. Цель состоит в том, чтобы создать модель, которая прогнозирует значение целевой переменной путем изучения простых правил принятия решений, выведенных из функций данных.

Стохастическим вариантом деревьев решения являются extra-trees. Такие деревья при поиске лучшего разделения на две группы, для каждого из случайно выбранных признаков происходит случайное разделение, и выбирается наилучшее среди них [2].

6. Пассивно-агрессивный алгоритм (passive aggressive) относится к группе линейных онлайн алгоритмов для обучения на больших данных (крупномасштабное, large-scale).

Результат работы алгоритма может быть таким же простым, как решение «да / нет», как в случае задач бинарной классификации, и таким же сложным, как строка в большом алфавите. Как только алгоритм сделал прогноз, он получает обратную связь с указанием правильного результата. Затем он может модифицировать свой механизм прогнозирования, предположительно улучшая шансы сделать точный прогноз в последующих раундах [6].

Алгоритм пассивный – если результат верный, то ничего не изменяется (сохранить модель). И агрессивный – если результат ошибочный, то веса изменяются так, чтобы результат был идеальный (ровно 1). То есть, алгоритм всегда корректен для последнего обучающего примера. Так, изменяются веса для значимых признаков выборки.

При решении сложных задач классификации (и не только) часто оказывается, что ни один из алгоритмов не обеспечивает требуемого качества. В таких случаях имеет смысл строить композиции алгоритмов, в которых ошибки отдельных алгоритмов взаимно компенсируются.

Ансамблевые методы – это парадигма машинного обучения, где несколько моделей (часто называемых «слабыми учениками») обучаются для решения одной и той же проблемы и объединяются для получения лучших результатов.

Методы ансамбля – это мета-алгоритмы, которые объединяют несколько методов машинного обучения в одну прогностическую модель, чтобы уменьшить дисперсию – bagging, смещение (bias) – boosting, или улучшить прогнозы – stacking [7].

1. Бэггинг (Bootstrap AGGregatING) – это технология классификации, использующая композиции алгоритмов, каждый из которых обучается параллельно и независимо друг от друга. Основой бэггинга является бутстреп (bootstrap) – статический метод генерации

выборокиз исходного набора данных путем случайного выбора элементов с повторениями в каждом из наблюдений.

Идея бэггинга состоит в том, чтобы подобрать несколько независимых моделей и усреднить их результаты для получения модели с меньшим разбросом. Однако на практике невозможно подобрать полностью независимые модели, так как для этого потребуется действительно много данных. Таким образом, полагаются на хорошие приблизительные свойства бутстреп выборок для подбора моделей, которые практически независимы.

Самым популярным алгоритмом бэггинга (и ансамблевых методов в целом) является случайный лес (Random Forest). Идея данного метода заключается в том, что каждое дерево может довольно хорошо прогнозировать, но скорее всего, переобучается на части данных. Если построить много деревьев, которые хорошо работают и переобучаются с разной степенью, возможно уменьшить переобучение путем усреднения их результатов [8].

2. Бустинг (boosting) представляет собой жадный метод построения последовательности алгоритмов машинного обучения, когда каждый следующий алгоритм стремится компенсировать недостатки композиции всех предыдущих алгоритмов.

Базовые модели, которые часто рассматриваются – это модели с низким разбросом, но с высоким смещением. Как правило, эти модели требуют меньших вычислительных затрат (несколько степеней свободы при подборе гиперпараметров). Однако, поскольку вычисления не могут выполняться параллельно (в отличие от бэггинга), методы слишком ресурсозатратные для последовательного подбора нескольких сложных моделей.

Самыми распространенными алгоритмами данной категории являются адаптивный бустинг (AdaBoost) и градиентный бустинг (Gradient Tree Boosting) [8].

Основным принципом AdaBoost является подбор последовательности слабых учеников в неоднократно измененных версиях данных. Прогнозы от всех них затем объединяются посредством взвешенного большинства голосов (или сумм) для получения окончательного результата. Причем, по мере продолжения итераций, примеры, которые трудно предсказать, получают все большее влияние. Таким образом, каждый последующий слабый ученик вынужден концентрироваться на примерах, которые были пропущены предыдущими в последовательности.

Основное отличие градиентного бустинга от адаптивного, использующего итерационный подход, заключается в определении процесса последовательной оптимизации. Градиентный бустинг сводит задачу к градиентному спуску: на каждой итерации происходит подгонка слабого ученика к антиградиенту текущей ошибки подбора по отношению к текущей модели ансамбля [7].

Большой раздел машинного обучения образуют нейронные сети. Искусственные нейронные сети (artificial neural network) представляют универсальный метод для решения различных задач, в частности, когда в представленных для обработки данных отсутствуют значительные фрагменты информации, а имеющаяся информация предельно зашумлена.

В целом искусственные нейроны образуют сложно связанную сеть, способную при определенном управляемом взаимодействии решать достаточно сложные задачи. Обучение нейронной сети отличается от обучения алгоритмов, описанных выше. Обучение заключается в расчете такого набора весовых коэффициентов, при котором входной сигнал после прохода по сети преобразуется в нужный выходной [9].

Результаты. Разделение исходного набора данных было произведено по модели 9:1 - 90% данных каждого класса образуют обучающую выборку, а оставшиеся 10% каждого класса образуют тестирующую выборку. Модели, основанные на описанных методах, были обучены и протестированы на соответствующих выборках. Для настройки моделей использовались случайный метод подбора параметров по сетке для быстрого отсеивания неподходящих значений и классический в качестве более точного.

Оценки политематической классификации сложнее классификации с несколькими классами в основном потому, что в предсказании с несколькими метками есть дополнительное представление о том, что результат частично корректный. Поэтому для данной классификации используются специфичные метрики (evaluation metrices).

В качестве первичной метрики использовалась «жесткая» достоверность (accuracy, hamming score), которая определяется как отношение прогнозируемых правильных меток к общему количеству (прогнозируемых и фактических) меток для этого экземпляра. Результат представлен на рисунке 1.

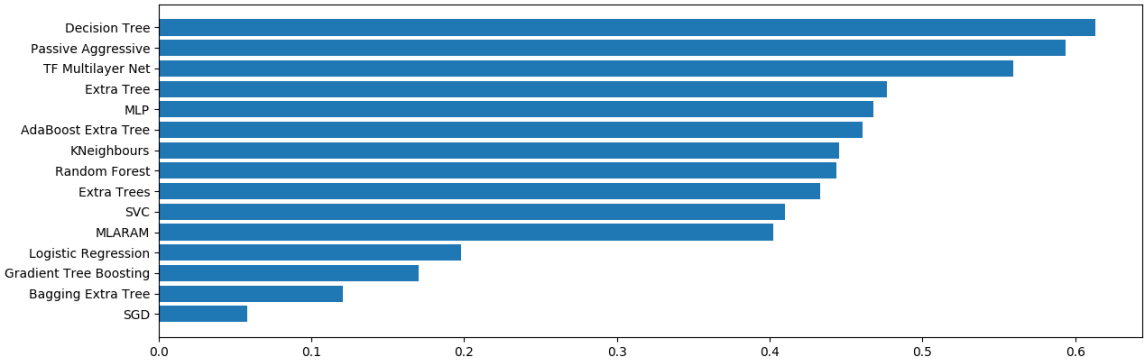


Рис. 1. Достоверность классификации

Данная метрика присваивает всем документам одинаковый вес, что может быть не корректно в том случае, когда распределение документов в обучающей выборке сильно смещено в сторону какого-то одного или нескольких классов. Основными метриками являются точность (precision), как отношение прогнозируемых правильных меток к общему количеству фактических меток, усредненное по всем экземплярам, полнота (recall) как отношение прогнозируемых правильных меток к общему количеству прогнозируемых меток, и f-мера (f-measure), как их среднее гармоническое. Каждая изданных метрик имеет макро, микро и взвешенную вариацию. Результат классификации по основным метрикам (основанным на экземплярах) представлен на рисунке 2.

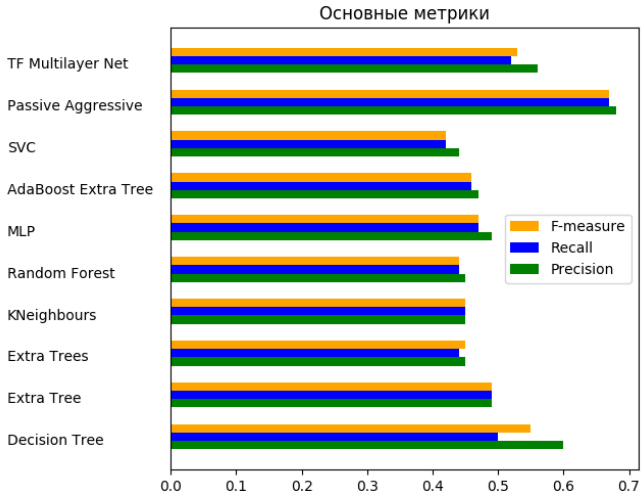


Рис. 2. Значения основных метрик лучших моделей

Так, лучшие результаты классификации показали деревья решений, пассивно-агрессивный алгоритм и многослойная нейронная сеть на основе TensorFlow.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. Sorower S. A. Literature Survey on Algorithms for Multi-label Learning. Mohammad // Department of Computer Science, Oregon State University. - 25p.
2. Мюллер А., Гвидо С. Введение в машинное обучение с помощью Python. Руководство для специалистов по работе с данными // СПб. : ООО «Альфа-книга», 2017. - 480 с.
3. Google developers guide for text classification [Электронный ресурс]. // URL:https://developers.google.com/machine-learning/guides/text-classification/step-3 (дата обращения 15.05.2019).
4. Official documentation for scikit-learn [Электронный ресурс] //URL: https://scikit-learn.org/stable/ (дата обращения 04.05.2019)
5. Shai Shalev-Shwartz, Shai Ben-David. Understanding Machine Learning: From Theory to Algorithms, 2014. - 449p.
6. Crammer K., Dekel O., Keshat J., Shalev-Shwartz S., Singer Y. Online Passive-Aggressive Algorithms // JMLR 7, 2006. - 35p.
7. Cha Zhang, Yunqian Ma. Ensemble Machine Learning. Methods and Applications // Springer, 2012. - 331p.
8. Ансамблевые методы: бэггинг, бустинг и стекинг [Электронный ресурс] // URL: https://neurohive.io/ru/osnovy-data-science/ansamblevye-metody-begging-busting-i-steking/ (дата обращения 05.05.2019)
9. Bernhard Mehlig. Artificial Neural Networks // Department of Physics, University of Gothenburg, Sweden. 2019. - 206 p.

УДК: 519.7

ПРИМЕНЕНИЕ ДУБЛИКАТОВ КОНФЛИКТНЫХ ОГРАНИЧЕНИЙ ДЛЯ УСКОРЕНИЯ РАБОТЫ АЛГОРИТМОВ РЕШЕНИЯ ПРОБЛЕМЫ БУЛЕВОЙ ВЫПОЛНИМОСТИ

Кондратьев В.С.
(г. Иркутск, Иркутский национальный исследовательский технический университет)
e-mail: vikseko@gmail.com

Аннотация. Целью данной работы является изучение эффекта повторного порождения конфликтных ограничений в CDCL-выводе и применение эвристических алгоритмов на основе данного эффекта для ускорения решения задачи булевой выполнимости (SAT). Эвристика повторно порождаемых конфликтных ограничений была реализована в рамках нескольких существующих SAT-решателей. В ходе вычислительных экспериментов удалось продемонстрировать работоспособность предложенной техники.

Ключевые слова. Задача о булевой выполнимости (SAT), алгоритм CDCL, конфликтные дизъюнкты, однопоточные CDCL-решатели.

APPLICATION OF DUPLICATES OF CONFLICT CONSTRAINTS TO SPEEDING UP OPERATION OF ALGORITHMS FOR SOLVING THE BOOLEAN SATISFIABILITY PROBLEM

Kondratiev V. S.
(Irkutsk, Irkutsk National Research Technical University, Irkutsk,)
e-mail: vikseko@gmail.com

Abstract. The purpose of this work is to study the effect of re-generation of conflict constraints in the CDCL output and to apply heuristic algorithms based on this effect to accelerate solving of Boolean satisfiability problem (SAT). Heuristic of re-generated conflict constraints has been implemented within several existing SAT-solvers. During the computational experiments it was possible to demonstrate the operability of the proposed technique.

Keywords. Boolean satisfiability problem (SAT), CDCL algorithm, conflict clause, single-threaded CDCL solvers.

Введение. Проблема булевой выполнимости (*Boolean Satisfiability Problem, SAT*) является одной из центральных комбинаторных проблем, поскольку представляет собой алгоритмическую задачу, тесно связанную с теорией вычислительной сложности и со многими другими областями теории алгоритмов. В общей постановке SAT является NP-трудной задачей, и, исходя из того, что P≠NP, для её решения не может существовать полиномиальных алгоритмов. Тем не менее, поскольку SAT очень удобна с точки зрения сведения к ней обширнейшего класса комбинаторных проблем, разработка алгоритмов, эффективных на различных подклассах SAT, не теряет актуальности на протяжении многих лет. В последние годы было предложено множество алгоритмов решения SAT, из которых особенно следует выделить CDCL (Conflict Driven Clause Learning) [1,2]. Характерная особенность данного алгоритма состоит в том, что он использует память для хранения информации о пройденных участках пространства поиска в форме специальных ограничений, называемых конфликтными дизъюнктами. Для целого ряда случаев было отмечено, что CDCL в процессе работы может многократно стирать и порождать одни и те же конфликтные дизъюнкты. Было экспериментально установлено, что запрет на удаление таких дизъюнктов может приводить к весьма существенному увеличению эффективности работы CDCL.

Основной результат работы состоит в новых техниках использования повторно порождаемых конфликтных ограничений. Применение этих техник позволило существенно ускорить решение широкого класса тестов, использующихся на ежегодных соревнованиях алгоритмов для решения задачи булевой выполнимости. Данные тесты включают в себя практические примеры из широкого спектра областей, такие как верификация, криптография, анализ комбинаторных структур и т.п.

Базовые понятия, подходы и алгоритмы. Пусть F – произвольная булева формула, задающая булеву функцию f над множеством булевых переменных $X = \{x_1, \dots, x_n\}$. Формула F называется *выполнимой*, если найдется такой набор $\alpha \in \{0,1\}^n$ значений переменных из X, что $f(\alpha) = 1$. Набор α с данным свойством называется набором, *выполняющим формулу* F. Если не существует наборов, выполняющих F, то данная формула называется *невыполнимой*. Задача о булевой выполнимости (SAT) заключается в следующем:

по произвольной булевой формуле F определить, является ли данная формула выполнимой. Так как SAT в отношении произвольной формулы может быть эффективно сведена к SAT в отношении формулы в конъюнктивной нормальной форме (КНФ), чаще всего под «SAT» понимается проблема выполнимости в отношении формулы в КНФ.

SAT – классическая NP-полная задача (см. [3,4]), следовательно, в предположении, что $P \neq NP$, она не может быть решена за полиномиальное время в общем случае. Однако, подобно многим другим NP-полным и NP-трудным задачам, SAT решается весьма эффективно для обширных классов своих частных случаев. Благодаря этому, за последние 20 лет алгоритмика SAT стала быстро развивающимся направлением, перечисление даже ключевых статей по которому в рамках данной работы излишне, поэтому уместно сослаться на широко известное руководство по SAT проблематике [5].

Прогресс в разработке практических алгоритмов решения SAT привел к развитию основанных на этих алгоритмах вычислительных методов, широко используемых на сегодня в таких областях, как символьная верификация [6], биоинформатика [7], построение комбинаторных структур [8], а также в криптоанализе [9].

Одна из наиболее плодотворных концепций решения SAT – это CDCL (*Conflict Driven Clause Learning*) и основанные на ней программные комплексы, называемые CDCL-решателями [10]. Далее будут кратко описаны основные составляющие архитектуры современных CDCL-решателей.

В основе концепции CDCL лежит алгоритм Девиса-Патнема-Лоджмана-Лавленда (DPLL) [11], применяющийся в автоматическом доказательстве теорем с 60-х годов прошлого века. DPLL – это алгоритм, направленного обхода дерева поиска, который использует простой бэктрекинг для выхода из тупиковых точек. В 1996 году, в работе [12] была реализована процедура запоминания тупиковых ветвей дерева поиска в виде новых ограничений, являющихся дизъюнктами (т.н. «конфликтные дизъюнкты»). Именно эта техника позже получила название Conflict Driven Clause Learning (CDCL). Принципиальное отличие алгоритмов DPLL и CDCL заключается в том, что последний использует память для хранения истории поиска, что позволяет совершать существенно более глубокие откаты, чем это происходит при бэктрекинге. Данный эффект получил название нехронологического бэктрекинга или бэкджампинга (*backjumping*).

Из вышесказанного следует негативная черта CDCL: огромная скорость генерации конфликтных дизъюнктов приводит к быстрому переполнению оперативной памяти. Использование же дисковой памяти ведет к резкому падению эффективности поиска. Исходя из этого, важной составной частью современных CDCL-решателей становятся процедуры чистки баз конфликтных дизъюнктов. Первым SAT-решателем, использующим подобные процедуры, стал minisat [13]. На сегодняшний день процедуры чистки конфликтных баз используются во всех CDCL-решателях, участвующих в проходящих ежегодно соревнованиях SAT-решателей. Стоит отметить, что все меры, оценивающие «полезность» дизъюнктов, являются эвристическими, и, соответственно, нет гарантий, что удаленные из конфликтной базы дизъюнкты не будут выведены в ходе дальнейшего поиска. В последующих разделах детально исследуется эффект повторного порождения конфликтных дизъюнктов современными CDCL-решателями и предлагается специальный механизм обнаружения и последующего использования таких дизъюнктов.

Порождение повторов конфликтных ограничений. Пусть C – произвольная КНФ над множеством булевых переменных $X = \{x_1, \dots, x_n\}$, задача выполнимости которой рассматривается. Будем решать ее при помощи SAT-решателей, основанных на концепции CDCL. Фундамент данной концепции составляет идея записи информации о тупиковых путях в дереве DPLL-поиска в форме конфликтных дизъюнктов. Конфликтный дизъюнкт хранит информацию о некоторых переменных и их значениях, одновременный выбор которых привел к конфликту.

Далее используется следующая простая идея: если некоторый конфликтный дизъюнкт D был выведен дважды в процессе работы алгоритма CDCL, использующего чистку конфликтной базы, то решатель неоднократно пытается пойти по соответствующему пути и, соответственно, D представляет собой ценную информацию. Имеет смысл запрещать удаление такого рода «ценных дизъюнктов». Далее в общем виде сформулирована схема эвристики, которая использует порождаемые повторно конфликтные дизъюнкты.

Эвристика повторно порождаемых конфликтных ограничений: анализируются все сгенерированные конфликтные дизъюнкты, и те из них, которые сгенерировались повторно, добавляются в качестве неудаляемых ограничений. Ниже реализация рассматривается более подробно, но сначала важно отметить некоторые особенности решателей на основе MapleCOMSPS [14], которые позволяют естественным образом реализовать предложенную эвристику.

Архитектура SAT-решателей, основанных на MapleCOMSPS. Победители ежегодных соревнований SAT Competition 2017 и 2018 годов в категории main track, Maple_LCM_Dist и MapleLCMDistChronoBT, являются вариациями решателя MapleCOMSPS, который выиграл SAT Competition 2016 года в той же категории. MapleCOMSPS построен на основе COMinisatPS [15], в свою очередь основанном на Minisat 2.2 и Glucose [16].

Особенность COMinisatPS и его производных, которые будут использованы ниже, состоит в том, что они разбивают все изученные конфликтные ограничения на три уровня полезности в зависимости от их lbd и core_lbd_cut параметров. Параметр lbd обозначает для каждого дизъюнкта число уровней решения (см., например, [17]), соответствующих его литералам. Уровень Core содержит ограничения с lbd ≤ core_lbd_cut, и соответствующие конфликтные дизъюнкты хранятся постоянно (они никогда не удаляются). Конфликтные дизъюнкты с core_lbd_cut < lbd ≤ 6 переходят на уровень Tier_2, а все остальные конфликтных дизъюнкты – на уровень Local. Конфликтные дизъюнкты, удалённые во время процедуры чистки базы конфликтных ограничений (reduceDB), переходят с Tier_2 на уровень Local. При этом дизъюнкты, уже находящие на уровне Local, удаляются из базы конфликтных ограничений. Во время процесса минимизации дизъюнктов размеры конфликтных ограничений могут быть уменьшены и, следовательно, могут быть уменьшены их lbd. Если новые значения lbd удовлетворяют условиям Core и Tier_2, то соответствующие дизъюнкты перемещаются на соответствующие уровни. Наконец, параметр core_lbd_cut по умолчанию равен 3, но увеличивается до 5, если после первых 100000 конфликтов уровень Core содержит менее 100 дизъюнктов.

Данные особенности идеально подходят для предложенной эвристики повторяющихся обучений: можно переместить обнаруженный дублирующий конфликтный

дизъюнкт в уровень Core независимо от его lbd и от того, было ли его появление результатом минимизации конфликтного дизъюнкта или процесса нахождения новых конфликтных ограничений.

Эвристика повторно порождаемых конфликтных ограничений: детали реализации. Обнаружение повторяющихся конфликтных ограничений (дизъюнктов) включает в себя сохранение базы данных (в форме хэш-таблицы) всех дизъюнктов, созданных за некоторый период времени. Следовательно, дизъюнкты для анализа на наличие дубликатов должны быть отфильтрованы (зачем анализировать дизъюнкт, который вряд ли когда-нибудь повториться), а период времени для накопления дизъюнктов в хеш-таблицу должен быть ограничен (в противном случае вычислительные затраты превысят потенциальную выгоду).

Первый момент может быть решен вполне естественно. Статистически, чем больше lbd, тем меньше шансов, что дизъюнкт когда-либо повторится. Таким образом, отбор дубликатов только для дизъюнктов с lbd, меньшим некоторого lbd-лимита, представляется рациональным.

Второй момент сложнее, потому что, если не ограничить время, когда условия конфликта проверяются на наличие дубликатов, размер хеш-таблицы будет очень быстро расти при разумных значениях lbd-limit. На текущем этапе была реализована простая схема с ограничениями по времени. В данной реализации, чтобы гарантировать воспроизводимость экспериментов, время напрямую зависит от количества рестартов.

Таким образом, в дальнейшем используется следующая простая стратегия: отмечать и добавлять дубликаты на уровень *Core* только при нескольких первых частых рестартах в процессе сбора, так как при уменьшении частоты рестартов, польза от повторных дизъюнктов может уменьшиться. После этого очищать хеш-таблицу и полностью прекращать работу с ней.

Хеш-таблица, используемая для поиска и хранения дубликатов, была реализована поверх класса unordered_map из стандартной библиотеки шаблонов C ++. Литералы в каждом изученном предложении сортируются в порядке возрастания, а затем обрабатываются хеш-таблицей.

Вычислительный эксперимент. Для реализации алгоритма были созданы модификации решателей Maple_LCM_Dist и MapleLCMDistChronoBT, названные Maple_LCM_Dist–mod и MapleLCMDistChronoBT–mod, соответственно, с описанной ниже встроенной эвристикой.

В течение некоторого времени, ограниченного числом рестартов, происходит анализ базы конфликтных ограничений и добавление всех полученных ограничений в хэш-таблицу. В случае, если ограничение уже содержится в хэш-таблице, оно добавляется напрямую в уровень Core. По истечении времени хэш-таблица удаляется, и работа данной эвристики прекращается.

Результаты работы построенных решателей сравнивались с их немодифицированными версиями на различных тестах, взятых с ежегодных соревнований алгоритмов для решения задачи булевой выполнимости. Они включают в себя практические примеры из широкого спектра областей, такие как верификация, криптография, анализ комбинаторных структур и т.п. Полученные результаты приведены в таблице 1.

В эксперименте в роли вычислительной платформы использовались узлы кластера “Академик В.М. Матросов”, установленного в ИДСТУ СО РАН. Конфигурация каждого узла: два 18-ядерных процессора Intel Xeon E5-2695 v4. Все решатели запускались в однопоточном режиме.

Таблица 1. Статистика количества решенных задач на тестах SAT Competition 2017 и 2018, задачи решались с ограничение по времени решения в 5000 секунд.

	SC 2017			**SC 2018**		
Решатель	Solved	SAT	UNSAT	Solved	SAT	UNSAT
Maple_LCM_Dist	204	100	104	227	127	100
Maple_LCM_Dist–mod	205	103	102	237	136	101
MapleLCMDistChronoBT	212	99	113	233	131	102
MapleLCMDistChronoBT–mod	211	100	111	239	136	101

Таблица 1 демонстрирует для каждого решателя: в столбцах Solved – общее число решенных задач, в столбцах SAT – число решенных выполнимых задач, в столбцах UNSAT – число решенных невыполнимых задач. Как можно понять из таблицы 1, построенные модифицированные решатели Maple_LCM_Dist–mod и MapleLCMDistChronoBT–mod показали сопоставимые результаты с оригинальными немодифицированными решателями на тестах с SAT Competition 2017, но существенно выиграли у них на задачах с SAT Competition 2018. Таким образом, предложенная эвристика дает существенный выигрыш при работе с выполнимыми задачами, но не дает преимуществ при решении невыполнимых тестов.

Заключение. В настоящей статье представлена реализация идеи о том, что повторяющиеся дизъюнкты, полученные в CDCL-выводе, могут быть использованы для повышения производительности SAT-решателя. Предложен вариант её возможной реализации в современных SAT-решателях на основе победителей SAT Competitions 2017 и 2018. Результаты экспериментов показывают, что эвристика хорошо работает с выполнимыми задачами, мало влияя на производительность решателя на невыполнимых задачах.

В дальнейшем предполагается улучшить обработку повторяющихся дизъюнктов и исследовать их влияние на CDCL-вывод более детально, в частности, проанализировать работу эвристики в сочетании с различными настраиваемыми параметрами SAT-решателей.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. Marques-Silva J., Sakallah K. GRASP: A Search Algorithm for Propositional Satisfiability // IEEE Transactions on computers. 1999. 45. N 5. 506–521.
2. Marques-Silva J., Lynce I., Malik S. Conflict-Driven Clause Learning SAT solvers // In Handbook of Satisfiability. 2009. 131–153.
3. Cook S.A. The complexity of theorem-proving procedures // Third annual ACM symposium on Theory of computing. 1971. 151–159.
4. Garey M., Johnson D. Computers and intractability: A guide to the theory of NP-completeness. 1979. Русский перевод: Гэри М., Джонсон Д. Вычислительные машины и труднорешаемые задачи // М.: Мир, 1982.

5. Biere A., Heule M., van Maaren H., Walsh T. Handbook of Satisfiability. IOS Press, 2009.

6. Biere A. Bounded model checking // In Handbook of Satisfiability. IOS Press, 2009. 457–481.

7. Lynce I., Marques-Silva J. SAT in Bioinformatics: Making the Sense of Haplotype Inference // Lecture Notes in Computer Science. 2006. 4121. 136–141.

8. Zhang H. Combinatorial Designs by SAT solvers // In Handbook of Satisfiability. IOS Press, 2009. 533–568.

9. Bard G. Algebraic cryptanalysis // Springer Publishing Company, Incorporated, 1st edition, 2009.

10. Marques-Silva J., Lynce I., Malik S. Conflict-driven clause learning SAT solvers // Handbook of Satisfiability. Amsterdam: IOS Press, 2009. 131–153.

11. Davis M., Logemann G., Loveland D. A machine program for theorem-proving // Communication of the ACM. 1962. 5, Issue 7. 394–397.

12. Marques-Silva J., Sakallah K. GRASP – a new search algorithm for satisfiability // Proceedings of ICCAD'96. 1996. 220–227.

13. Een N., Sorensson N. An extensible SAT-solver // Lecture Notes in Computer Science. 2004. 2919. 502–518.

14. Liang J.H., Ganesh V., Poupart P., Czarnecki K. Learning rate based branching heuristic for SAT solvers. // SAT. LNCS, vol. 9710. 2016. 123–140.

15. Oh C. Between SAT and UNSAT: The fundamental difference in CDCL SAT. // SAT. LNCS, vol. 9340. 2015. 307–323.

16. E´en N., Biere A.: Effective preprocessing in SAT through variable and clause elimination. // Bacchus, F., Walsh, T. (eds.) SAT. LNCS, vol. 3569. 2005. 61–75.

17. Marques-Silva J., Sakallah K. GRASP: a search algorithm for propositional satisfiability // IEEE Transactions on Computers, vol. 48, issue: 5. May 1999, 506–521.

УДК 004.9

КЛАСТЕРИЗАЦИЯ ТОЧЕК ЕВКЛИДОВОЙ ПЛОСКОСТИ С ОГРАНИЧЕНИЕМ ПО ВЕСУ

Сиротинина Е.А., Маланова Т.В.
(г. Иркутск, Иркутский национальный исследовательский технический университет)
e-mail: katrin.kobre@gmail.com

Аннотация. В статье рассматриваются два алгоритма для решения задачи кластеризации с ограничением на суммарный вес точек в каждом полученном кластере. Предложена их модификация и выполнен сравнительный анализ по точности разбиения и скорости работы.

Ключевые слова. Кластеризация, алгоритм k-means, алгоритм DBSCAN.

CLUSTERING OF POINTS OF THE EUCLIDEAN PLANE WITH A LIMIT ON THE WEIGHT

Sirotinina E.A., Malanova T.V.
(Irkutsk, Irkutsk National Research Technical University, Irkutsk)
e-mail: katrin.kobre@gmail.com

Abstract. The article considers two algorithms for solving the clustering problem with a restriction on the total weight of points in each obtained cluster. Their modification is proposed and comparative analysis on accuracy of clustering and speed of operation is performed.

Keywords. Cluster analysis, k-means clustering, DBSCAN algorithm.

Введение. В практических приложениях существует множество задач, решение которых связано с понятием кластеризации – объединения объектов в непересекающиеся группы. Разработанные методы кластеризации различаются подходами к созданию групп [1], поэтому применяются во многих областях: планирование перевозки грузов, маркетинговый анализ, анализ текстов и т.д. В большинстве из них не предусмотрено ограничение по количеству элементов в кластере. Такое ограничение устанавливается в зависимости от условий поставленной задачи [2, 3].

В настоящей работе рассматриваются задачи кластеризации, в которых для каждой точки определены значения весов (масса объекта, объем, какой-либо коэффициент и пр.), и задано ограничение на суммарное значение веса в каждом кластере. Предложена модификация существующих алгоритмов кластеризации точек евклидовой плоскости с ограничениями по весу и проведен сравнительный анализ полученных результатов.

Постановка задачи. Задача кластеризации точек плоскости a_i с координатами (x_i, y_i) и значениями веса каждой точки v_i, в которой заданы ограничения на суммарный вес точек кластера, заключается в следующем.

Найти функцию кластеризации f^* такую, что

$$Q(f^*, C, d) = \min_f Q(f, C, d), \tag{1}$$

где C – множество кластеров, $Q(f, C, d)$ – среднее внутрикластерное расстояние:

$$Q(f, C, d) = \frac{\sum_l \sum_{a_i, a_j \in C_l} d_{ij}}{L}. \tag{2}$$

Здесь l – номер кластера; L – число кластеров; d_{ij} – расстояния между точками a_i и a_j, которые вычисляются по формуле:

$$d_{ij} = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2}. \tag{3}$$

Пусть $V^l = \sum_{i=1}^{n^l} v_i^l$ – суммарный вес точек a_i в кластере под номером l; n^l, v_i^l – число точек l-го кластера, вес каждой точки l-го кластера соответственно. Число элементов в каждом кластере ограничивается следующим неравенством:

$$V^l < V_0 \tag{4}$$

где V_0 – заданное число.

Решение задачи кластеризации с учетом ограничения по весу. Для решения поставленной задачи (1) – (4) в качестве базовых выбраны следующие алгоритмы:

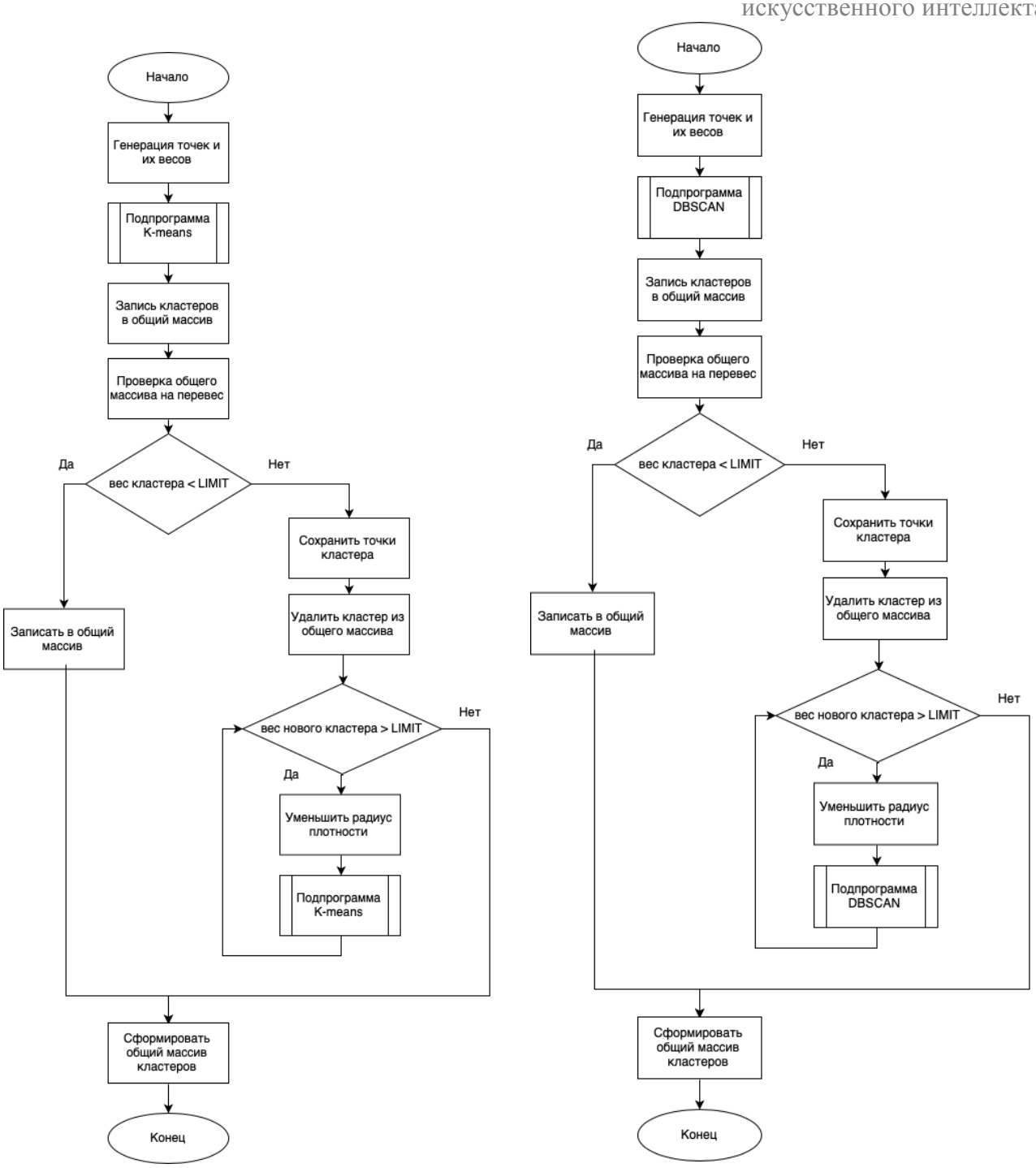
- Алгоритм k-means (к-средних). Задаются k центроидов со случайными координатами. Далее каждую точку присоединяют к ближайшему центроиду. Евклидово расстояние определяют по формуле (3). Центроиды перемещают в центры сформированных кластеров, исходя из среднего евклидова расстояния от всех точек, и выполняют очередная итерация проверки расстояний от точек до центроидов. Алгоритм заканчивает работу, если позиции точек и центроида на предыдущей итерации идентичны текущим. Главным достоинством k-means является простота реализации. Алгоритм быстро работает на сравнительно небольших объемах данных и справляется с любым расположением точек на плоскости. [4].
- Алгоритм DBSCAN (density based spatial clustering of application with noise – плотностной алгоритм кластеризации). Для каждой точки определяет всех ее соседей в круге определенного радиуса. Расстояние до точек вычисляют по формуле (3), после чего сравнивают с заданным радиусом. Кластер формируется из точек, которые оказались внутри. Если вокруг точки не образовался кластер из k точек, то она считается шумом и исключается из множества. Алгоритм DBSCAN не требует предварительно заданного числа кластеров и обеспечивает быстрый поиск групп точек на плоскости.

Поскольку в рассматриваемых алгоритмах не предусмотрено выполнение условия (4), то необходимо их модифицировать. Основной идеей является повторное применение алгоритма к первичным кластерам и разбиение их на меньшие кластеры до тех пор, пока вес каждого кластера не будет соответствовать ограничению. Блок-схемы с применением алгоритмов k-means и DBSCAN для решения задачи (1) – (4) представлены на рисунке 1.

Проведение экспериментов. Проведем сравнительный анализ рассматриваемых алгоритмов по следующим критериям:

1) число шагов кластеризации;
2) число получившихся кластеров (чем их меньше, тем лучше);
3) внутрикластерное расстояние;
4) межкластерное расстояние;
5) минимальное значение $V_0 - V^l$.

Алгоритм k-means зависит от количества начальных кластеров. Чем их больше, тем меньше итераций требуется произвести. Алгоритм DBSCAN зависит от двух параметров – радиуса поиска соседей и минимального числа точек в кластере. Чем больше радиус, тем меньше итераций и меньше кластеров. Точность разбиения характеризуется следующим образом: чем больше межкластерное расстояние, меньше внутрикластерное расстояние и меньше максимальный остаток по весу, тем лучше алгоритм разбил множество точек на кластеры.



а) *б)*

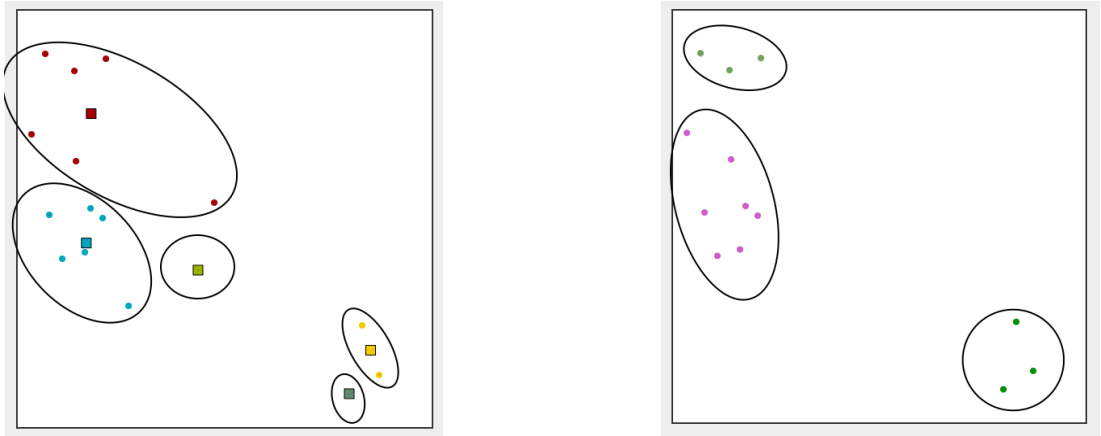
Рис. 1. Блок-схемы с применением алгоритмов
а) k-means; б) DBSCAN

Далее описан ряд экспериментов, представляющих типичные случаи.

1. Дано незначительное количество точек и их неравномерное распределение на плоскости.

Таблица 1. Результаты проведения эксперимента 1

Критерий	Алгоритм k-means	Алгоритм DBSCAN
Число кластеров: 3; Максимальный суммарный вес кластера: 35; Число точек: 16; Радиус: 80.		
число шагов кластеризации	4	0
число получившихся кластеров	5	3
внутрикластерное расстояние	20	19
межкластерное расстояние	61	181
$\min_l (V_0 - V^l)$	3	7



а) *б)*

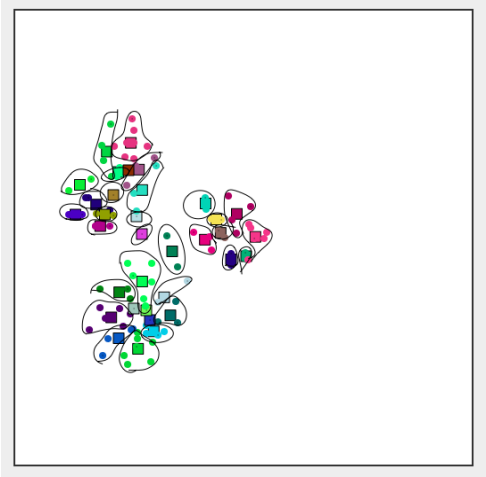
Рис. 2. Незначительное количество точек, неравномерное распределение
а) k-means; б) DBSCAN

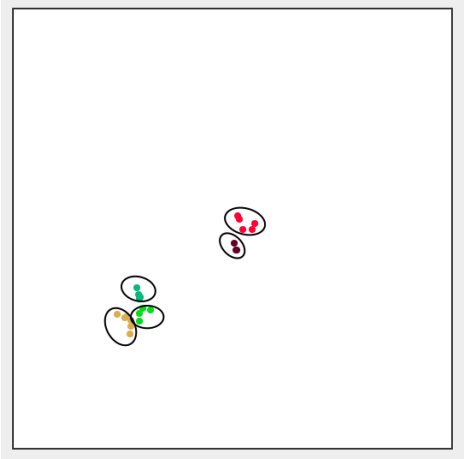
По данным таблицы 1 и рисунку 2 можно сделать вывод, что DBSCAN выполнил кластеризацию по всем параметрам лучше, чем k-means

2. Дано большое количество точек и их неравномерное распределение на плоскости.

Таблица 2. Результаты проведения эксперимента 2

Критерий	Алгоритм k-means	Алгоритм DBSCAN
Число кластеров: 3; Максимальный суммарный вес кластера: 35; Число точек: 102.		
число шагов кластеризации	46	32
число получившихся кластеров	33	6
внутрикластерное расстояние	6	2
межкластерное расстояние	13	140
$\min_l (V_0 - V^l)$	7	8





а) *б)*

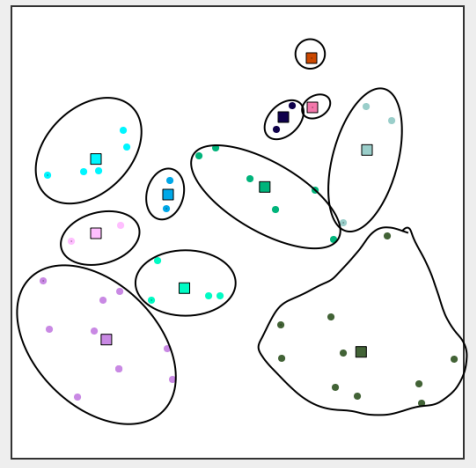
Рис. 3. Значительное количество точек, неравномерное распределение
а) k-means; б) DBSCAN

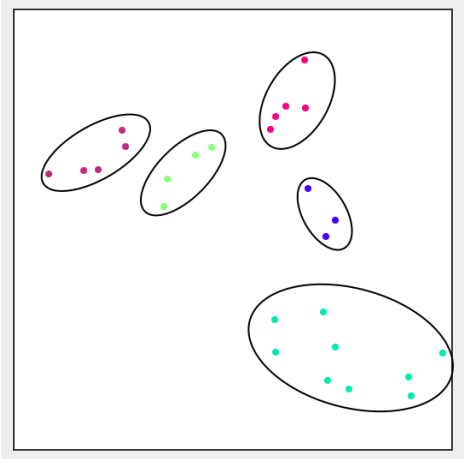
Не смотря на то, что DBSCAN дал лучшие результаты по большинству параметров чем k-means, он удалил большую часть точек.

3. Дано незначительное количество точек и их равномерное распределение на плоскости.

Таблица 3. Результаты проведения эксперимента 3

Критерий	Алгоритм k-means	Алгоритм DBSCAN
Число кластеров: 3; Максимальный суммарный вес кластера: 35; Число точек: 47.		
число шагов кластеризации	17	14
число получившихся кластеров	11	5
внутрикластерное расстояние	15	16
межкластерное расстояние	12	104
$\min_l (V_0 - V^l)$	4	2





а) *б)*

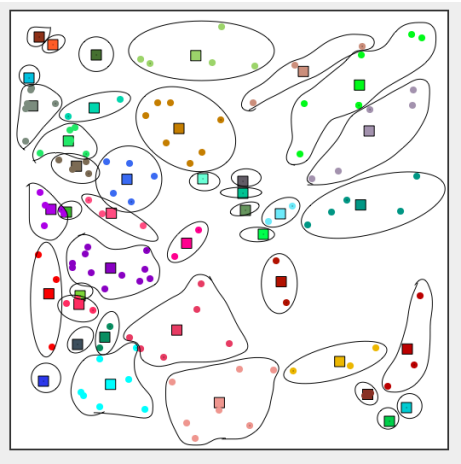
Рис. 4. Незначительное количество точек, равномерное распределение
а) k-means; б) DBSCAN

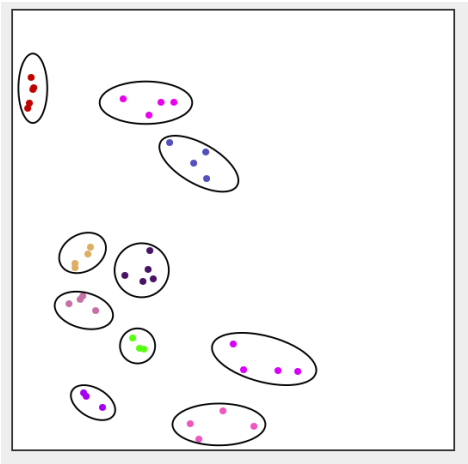
В целом алгоритм DBSCAN показал результаты лучше, чем k-means.

4. Дано большое количество точек и их равномерное распределение на плоскости.

Таблица 4. Результаты проведения эксперимента 4

Число кластеров: 3; Максимальный суммарный вес кластера: 35; Число точек: 130.		
число шагов кластеризации	60	32
число получившихся кластеров	43	10
внутрикластерное расстояние	8	4
межкластерное расстояние	20	40
$\min_l (V_0 - V^l)$	3	2





а) *б)*

Рис. 5. Значительное количество точек, равномерное распределение
а) k-means; б) DBSCAN

Несмотря на то, что k-means работает медленнее, он дал более точные результаты, чем DBSCAN.

Вывод. Алгоритм DBSCAN дает более точные результаты по выбранным критериям, чем алгоритм k-means, то есть формируются более плотные группы точек (кластеры) с меньшим внутрикластерным и большим межкластерным расстоянием, вследствие того, что наиболее удаленные от сформированных кластеров точки исключаются из рассмотрения. Однако не во всех прикладных задачах такое допустимо, и это является существенным недостатком данного алгоритма.

K-means, при сравнимом значении плотности точек, уступает DBSCAN по параметру межкластерного расстояния и по скорости, поскольку разбивает кластеры с перевесом не на произвольное число групп, а на заданное.

Также следует отметить, что минимальное значение $V_0 - V^l$ в результате кластеризации по рассматриваемым алгоритмам может получиться большим, и, если при конкретной постановке задачи это окажется существенным недостатком, то необходимо провести дополнительную модификацию данного алгоритма при условии $\min_l (V_0 - V^l) \to 0$.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. Дж.-О. Ким, Мьюллер Ч. У., Клекка У. Р. Факторный, дискриминантный и кластерный анализ / Ред. И. С. Енюков, Пер. с англ. А. М. Хотинский, С. Б. Королев. - М. : Финансы и статистика, 1989. - 215 с. : ил.
2. Лавренко А.С., Никульчев Е. В. Организационно-экономические механизмы и инструменты принятия решений при ассортиментном планировании в торговых сетях // Монография. – М.: ВГНА Минфина России, 2011. – 130 с.
3. Киреев В. С. Оценка результатов кластеризации при использовании различных критериев качества // Программные продукты и системы, 2009, №3, С. 36-39.
4. Нейский И.М. Классификация и сравнение методов кластеризации // Интеллектуальные технологии и системы. Сборник учебно-методических работ и статей аспирантов и студентов. – М.: НОК CLAIM», 2006. – Выпуск 8. – С. 130-142.

УДК 519.173.5:81.322

КОМПЛЕКСНЫЕ СЕТИ В СТИЛОМЕТРИИ РУССКИХ ЛИТЕРАТУРНЫХ ТЕКСТОВ

Андреева А.М., Труфанов А.И.
(г. Иркутск, Иркутский национальный исследовательский технический университет)
e-mail: nastic.1997@gmail.com

Аннотация. Статья посвящена проблеме подлинности литературных произведений и их авторстве. Подготовлена и опробована методика стилометрии русских литературных текстов на основе использования комплексных сетей. Методика опробована на примере атрибуции отрывков из сочинений М.Шолохова и Ф.Крюкова.

Ключевые слова. Сетевой анализ, атрибуция текста, русская литература, сетевой отпечаток, топологические метрики.

COMPLEX NETWORKS IN STYLOMETRY OF RUSSIAN LITERARY TEXTS

Andreeva A.M., Trufanov A.I.
(Irkutsk, Irkutsk National Research Technical University, Irkutsk,)
e-mail: nastic.1997@gmail.com

Abstract. The article is devoted to the problem of authenticity of literary works and their authorship. The method of stylometry of Russian literary texts based on the use of complex networks was prepared and tested. The technique is tested on the example of attribution of excerpts from the works of M. Sholokhov and F. Kryukov.

Keywords. Network analysis, text attribution, Russian literature, network imprint, topological metrics.

Введение. Споры о подлинности литературных произведений и их авторстве хорошо известны в отечественной истории так и за рубежом. Этот вопрос пробуждает интерес у многих экспертов: филологов, писателей, криминалистов, историков, да и просто пытливых читателей.

Более двух столетий проблема подлинности адресуется к сочинению «Слово о полку Игореве» [1]. В СССР получил значимый общественный отклик, не утихающий и в современной России, комплекс литературоведческих, и связанных с ними этических, политических и иных вопросов и споров, возникших после публикации в 1928 году романа «Тихий Дон» М. Шолохова, авторство которого по разным поводам и причинам одной частью исследователей подвергается сомнению, а другой считается неоспоримым [2,3].

Презумпция авторства является одним из наиболее простых вариантов правовой трактовки. Суть ее, согласно ст. 1257 ГК РФ, п. 1 ст. 15 Бернской конвенции по охране литературных и художественных произведений, заключается в том, что поскольку автором признаётся то лицо, чьё имя указано на экземпляре произведения, и, пока не доказано иное, автор подтверждает своё авторство путем предъявления наиболее раннего экземпляра с таким указанием [4].

Для иных доказательств нужны различные документы, материалы и свидетельства которые появились в процессе написания произведения.

Проведение экспертизы, как менее популярного способа подтверждения авторства, применяется тогда, когда возникают дополнительные сомнения в авторстве.

Существует множество методов для определения особенностей (стиля) автора. Причем основными можно назвать: латентное размещение Дирихле [5], мера Хмелёва [6] и сетевой анализ. Следует отметить, что при помощи семантического сетевого анализа могут быть установлены ключевые слова, центральные герои произведения, межтекстовые связи. Так, в [7] представлен алгоритм расчетной индикации контекста слов и цельных текстов, использующего семантическую сеть, описываемую графом – деревом.

Иной, не семантический анализ текстов, использует онтологию комплексных сетей. Этот метод был предложен как для сравнения особенностей различных языков [8] , так и для стилометрии (атрибуции) – для выявления стилей авторов одного и того же языка [9].

Целью настоящего исследования являлась разработка методики, основанной на применении комплексных сетей, нового популярного приема, но практически не применявшегося для идентификации авторства текстов, составленных на русском языке.

Модель и данные. Для достижения цели были поставлены следующие задачи:

- обосновать выбор модели сетевого представления текста и подготовить реализующее ее программное средство (программный код);
- осуществить выбор текстов конкретных художественных произведений, приписываемых двум различным авторам;
- настроить программу сетевого анализа и визуализации сети;
- провести расчет и сравнение метрик сетевых отпечатков, выбранных текстов;
- проанализировать полученные результаты.

Решение первой задачи основано на использовании сетевого описания [10-12]. По своей сути эта модель («co-occurrence») близка к модели «соседства» при сетевой идентификации диктора на основе аудиозаписи [13].

В качестве тестируемых текстов рассмотрены произведения Михаила Шолохова и Федора Крюкова. Из каждого произведения полагалось достаточным выбрать одну страницу, наиболее четко отражающую особенности выражения мысли автора.

Основные результаты и обсуждение. В настоящей работе был подготовлен программный код, создающий сетевой отпечаток для заданного на входе текста. Код написан на ЯВУ C# с использованием интегрированной среды Visual Studio 2015.

Предложена методика, для которой результат выполнения этого программного кода – файл в формате .csv импортируется в Gephi – бесплатный эффективный программный инструмент сетевого анализа [14] . Gephi позволяет произвести расчет ключевых метрик сети и ее визуализацию (см. рис. 1).

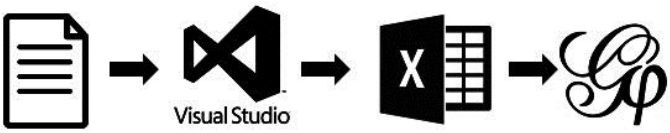


Рис. 1. Схема методики сетевого анализа

Методику тестирования можно рассмотреть на примере одного предложения из произведения «Тихий Дон»: «Мелеховский двор – на самом краю хутора». После использования программы удаляются все слова меньше 3 символов, т.к. чаще всего эти слова не несут смысловой нагрузки, все заглавные буквы заменяются на строчные. Создаются две таблицы в MS Excel. Одна из которых представляет узлы, а другая – связи (см. рис. 2).

	A
1	Id,Label,timeset
2	0,мелеховский,
3	1,двор,
4	2,самом,
5	3,краю,
6	4,хутора,

1	Source,Target,Type,Id,Label,timeset,Weight
2	0,1,Undirected,0,,,1
3	0,2,Undirected,1,,,0.5
4	0,3,Undirected,2,,,0.25
5	0,4,Undirected,3,,,0.125
6	1,2,Undirected,4,,,1
7	1,3,Undirected,5,,,0.5
8	1,4,Undirected,6,,,0.25
9	2,3,Undirected,7,,,1
10	2,4,Undirected,8,,,0.5
11	3,4,Undirected,9,,,1

Рис. 2. Таблицы узлов и связей

Веса связей рассчитываются по формуле:

$$P = \left(\frac{1}{2}\right)^{a}, \quad (1)$$

где P – вес, a – число посредников.

Для визуализации сетевой структуры используется инструментарий Gephi (Рис. 3).

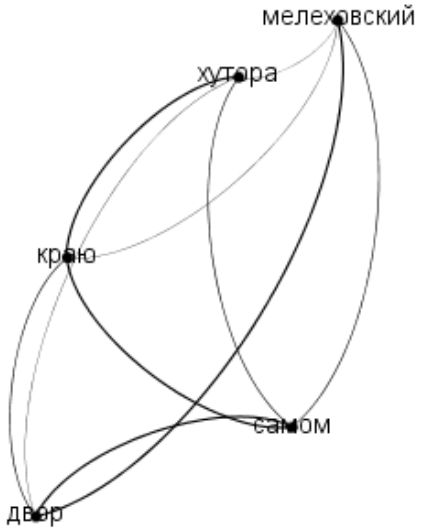


Рис. 3. Визуализации сети выбранного предложения

Отметим, что для полного исследования были взяты произведения М.Шолохова – «Родинка» и «Тихий Дон», Ф.Крюкова - «Будни». Так как объем произведений очень разнится, то из каждого произведения отобраны отрывки близкого объема.

Казалось бы, при визуализации сетевые отпечатки этих произведений (соответствующие комплексные сети) почти не отличаются, так как количество узлов и связей в них близко (Рис. 4).

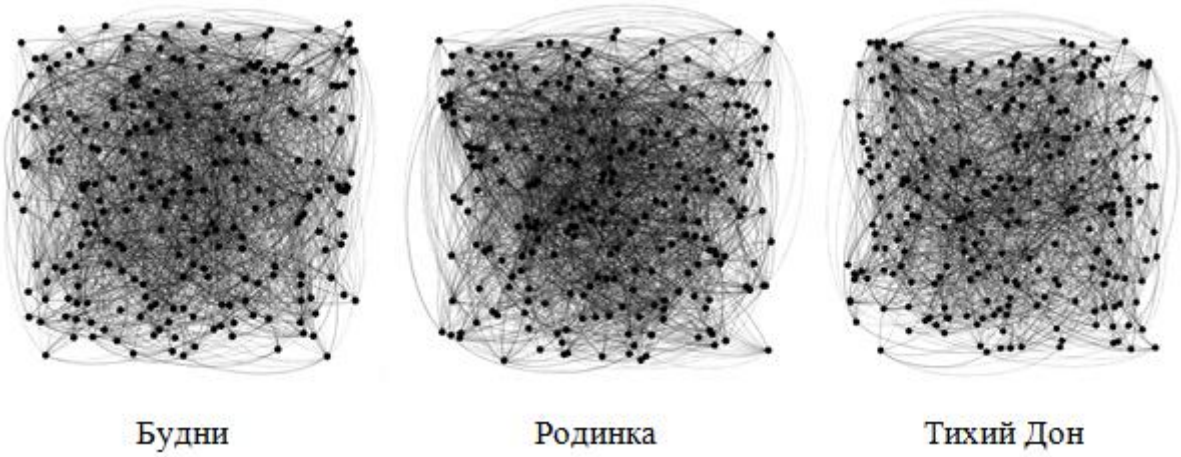


Рис. 4. Визуализации сетевых отпечатков литературных произведений

Однако, топологические характеристики соответствующих сетей – метрики – дают возможность провести это различие (в табл.1 и на рис. 5. приведены результаты расчета десяти ключевых метрик для сетевых отпечатков исследуемых произведений).

Таблица 1. Метрики сетевых отпечатков исследуемых произведений

	Тихий Дон	Родинка	Будни
Средняя степень	12,284	19,035	16,92
Средняя взвешенная степень	3,387	3,799	3,626
Диаметр графа	8	8	6
Плотность графа	0,046	0,074	0,068
Модулярность	0,888	0,823	0,827
Связные компоненты	16	2	9
Средний коэффициент кластеризации	0,97	0,967	0,97
Средняя длина пути	2,863	3,286	2,667
Кол-во узлов	269	258	250
Кол-во рёбер	1653	2483	2148

Характерно, что прослеживается близость значений сетевых метрик для отпечатков выбранных отрывков произведений предполагаемых разных авторов – «Тихий Дон» и «Будни», нежели, возможно, одного автора – «Тихий Дон» и «Родинка».

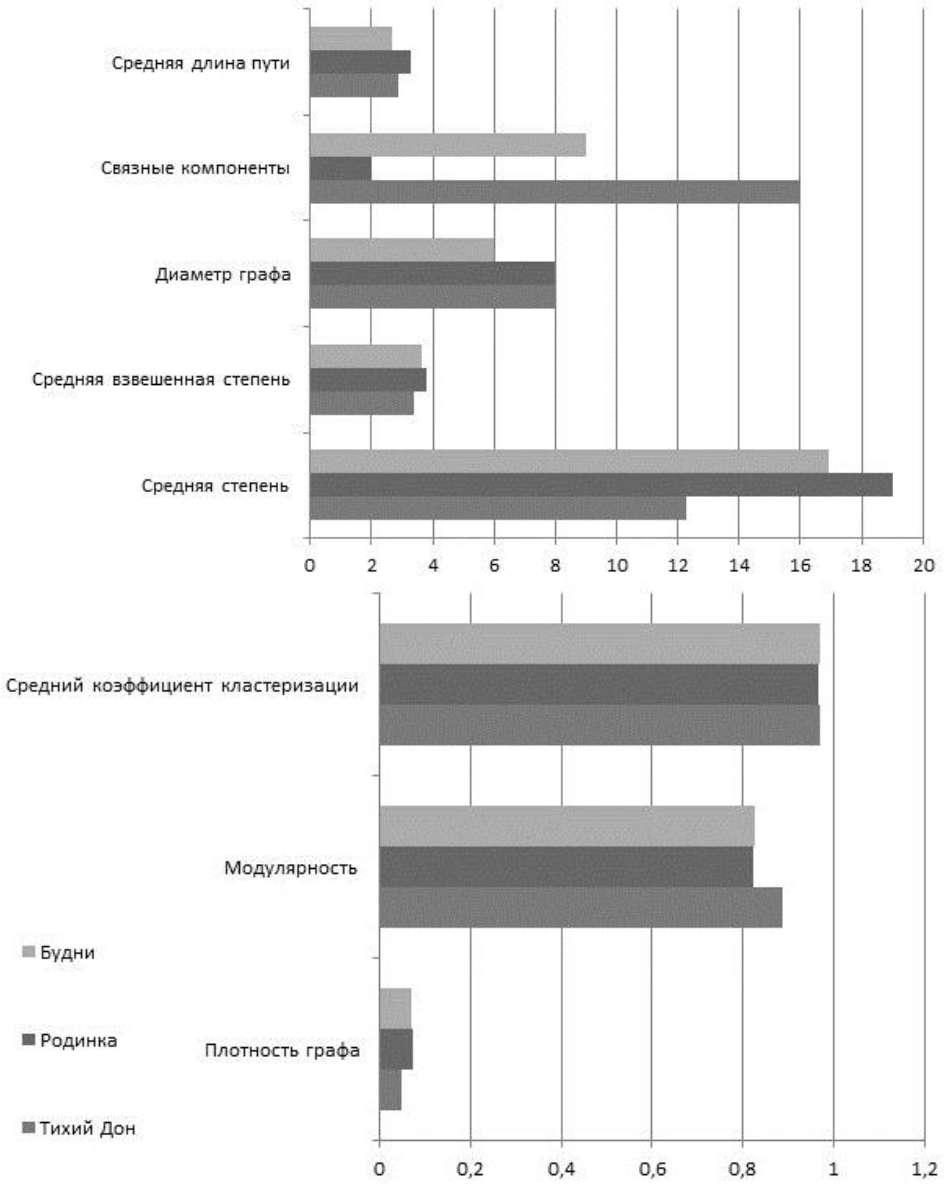


Рис. 5. Диаграмма результатов расчета сетевых метрик для соответствующих отпечатков трех исследуемых произведений

Выводы. Таким образом, в настоящей работе подготовлена и опробована методика стилометрии – количественной оценки стиля русских литературных текстов на основе использования комплексных сетей;

Сравнение сетевых метрик, характеризующих стили отрывков произведений известных авторов Ф. Крюкова и М. Шолохова, указывает на значительную вероятность первого к авторству романа «Тихий Дон» и тем самым, на незаконченность известного спора;

Уверенный ответ об авторстве знаменитого произведения потребует дальнейшего совершенствования сетевой методики стилометрии и ее масштабного тестирования.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. Алексеев, С. В. Игорь Святославич. - М. : Молодая Гвардия, 2013– 348 с.
2. Кофырин,Н. Правда о «Тихом Доне» [Электронный ресурс] // URL: http://blog.nikolaykofyrin.ru/?p=366 (дата обращения 27.05.2019).

3. Академик. Проблема авторства произведений Михаила Шолохова [Электронный ресурс] // URL: https://dic.academic.ru/dic.nsf/ruwiki/1107188 (дата обращения 27.05.2019).

4. Как доказать авторство [Электронный ресурс]// URL: https://bardov.legal/faq/kak-dokazat-avtorstvo

5. Академик. Скрытое распределение Дирихле [Электронный ресурс] // URL: https://dic.academic.ru/dic.nsf/ruwiki/1629260 (дата обращения: 27.05.2019).

6. Сравнительный анализ эффективности алгоритмов распознавания авторства текстов по частотам переходов [Электронный ресурс] // URL: http://sun.tsu.ru/mminfo/000063105/290/image/290-232.pdf (дата обращения: 27.05.2019).

7. Доренская, Е. А., Семёнов, Ю.А. Метод определения контекстных значений слов и документов // Международный научный журнал «Современные информационные технологии и ИТ-образование», 14, (4), 896-902

8. Sheng L. , Li C. English and Chinese languages as weighted complex networks // Physica A .388 2561–2570

9. Amancio D. R., Aluisio S. M., Oliveira O. N. Jr , Costa L. F. Complex networks analysis of language complexity Europhys // Lett. 100 58002

10. De Arruda H. F., Marinho V. Q., Costa L. da F., Amancio D. R. Paragraph-based representation of texts: A complex networks approach // Information Processing & Management, 56(3), 479–494.

11. De Arruda H. F., Marinho V. Q., Lima T. S., Amancio D. R., Costa L. da F. An image analysis approach to text analytics based on complex networks // Physica A: Statistical Mechanics and Its Applications, 510, 110–120.

12. Amancio D. R. Authorship recognition via fluctuation analysis of network topology and word intermittency // Journal of Statistical Mechanics: Theory an.-d Experiment, 2015(3), P03005

13. Куулар Э.К., Тихомиров А.А., Труфанов А.И.. Идентификация источников звуковой информации методом сетевого анализа // Безопасность информационных технологий. 2016, № 2, 43-48.

14. Gephi [Электронный ресурс] // URL: https://gephi.org/users/download/ (дата обращения: 27.05.2019).

УДК 004.42:004.042

ПРОГРАММНО-ИНСТРУМЕНТАЛЬНАЯ СРЕДА «ГРЛАО» ДЛЯ ОЦЕНКИ ЭФФЕКТИВНОСТИ РАБОТЫ ОРГАНИЗАЦИИ

Аршинский В.Л., Лэ Ван Хуен
(г. Иркутск, Иркутский национальный исследовательский технический университет)
e-mail: arshinskyv@mail.ru

Аннотация. В рамках данной статьи рассматриваются вопросы создания программно-инструментальной среды для агрегированной оценки эффективности деятельности организации «ГрЛАО». Расчеты эффективности системы и её подсистем в среде «ГрЛАО» реализуются на основе логико-аксиологического подхода к оценке состояния произвольной системы.

Ключевые слова. Программно-инструментальные средства; среда моделирования, оценка состояния системы; логико-аксиологический подход.

SOFTWARE TOOL "GRLAO" FOR EVALUATING EFFECTIVENESS OF ORGANIZATIONS

Arshinsky V.L, Le Van Huyen
(Irkutsk, Irkutsk National Research Technical University)
e-mail: arshinskyv@mail.ru

Abstract. The article deals with the development of software tool "GrLAO" for the aggregate evaluation of the effectiveness of an organization. Calculations of the efficiency of the whole system and its subsystems implemented on the basis of a logical-axiological approach to assessing the state of an arbitrary system.

Keywords. Software tools, graphical modeling environment, system state assessment, logical-axiological approach.

Введение. Применение системного подхода для управления деятельностью организаций и оценки эффективности принимаемых управленческих решений на основе таких инструментов как, например, ключевые показатели эффективности (Key Performance Indicator, сокращённо – KPI) становится всё более широко распространённой практикой [1]. В настоящее время системы управления предприятиями на основе KPI широко используются во многих областях, таких как машиностроение, производство, продажи, образовательные услуги и т.д. Использование системного подхода на практике в организациях даёт множество преимуществ в управлении их деятельностью и рабочими процессами. Системный подход представляет большой интерес для управленцев, поскольку они прямо влияют на деятельность и прибыль организаций.

Для того чтобы принимать решение об обслуживании любой системы, необходимо определить текущее состояние её работы с использованием некоторых показателей. При этом чем более разнообразная система индикаторов используется, тем лицам принимающим решения (ЛПР) – руководителям организации и отдельных подразделений – становится труднее увидеть текущую ситуацию в целом. Решение данного противоречия может быть найдено в использовании агрегированных оценок, позволяющих на верхнем уровне охватить одним взглядом всю картину в целом, а также выдвигать и проверять гипотезы о возможных способах повышения эффективности работы. В рамках данной работы описываются результаты попытки разработки программно-инструментального средства для оценки эффективности системы, опирающейся на аппарат логико-аксиологического подхода к агрегированной оценке состояния систем, предложенного в [2] и развиваемого в работах [3, 4, 5].

Актуальность данной работы обуславливается тем, что существующие программные инструменты, реализующие данный подход, не могут быть использованы ЛПР в силу причин которые будут рассмотрены ниже.

Обзор существующих программных средств. В настоящее время существует только два инструмента реализующих возможность агрегированной оценки на основе логико-аксиологического подхода: программный комплекс «лАос» (свидетельство о регистрации программ для ЭВМ №2016662218 от 02.11.16), разработанный автором и идеологом подхода Аршинским Л.В., и программная библиотека «jLA», реализованная Балановским Д.К.[6].

Программный комплекс «лАос». Этот программный комплекс разработан создателем логико-аксиологического подхода к оценке состояния сложных систем Аршинским Л.В. –

доктор технических наук, заведующий кафедрой информационных систем и защиты информации, Иркутского государственного университета путей сообщения (ИрГУПС). Первая версия была разработана в 2015 году. На данный момент времени актуальная версия «лАос 2.0» (см. рис. 1).

Данный программный комплекс реализует логико-аксиологический подход к построению агрегированных оценок систем различного назначения.

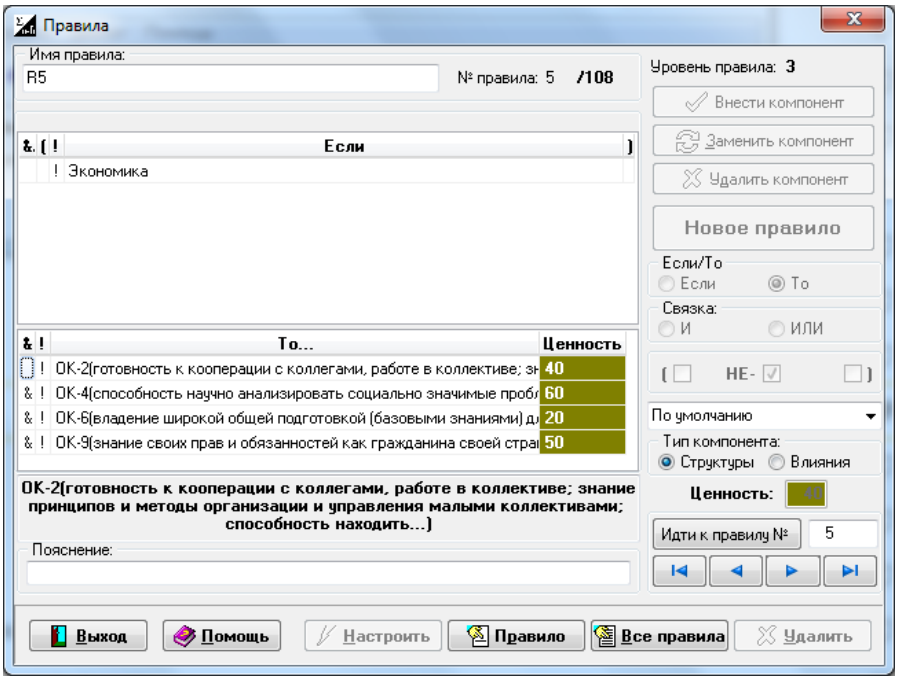


Рис. 1. Пример интерфейса окна ввода компонентов системы в «лАос 2.0»

Программный комплекс «лАос» на данный момент является основным программным средством, которое может использоваться для оценки состояния системы на основе логико-аксиологического подхода. Одним из преимуществ лАос является то, что он показывает пользователям вычислительные процессы логико-аксиологического подхода, которые основывает на продукционной модели представления знаний [7]. Это делает программный комплекс более подходящим, если он используется в образовательном процессе и научных исследованиях. Одним из недостатков этого программного комплекса является то, что он достаточно сложен в процессе использования, так как рассчитан скорее на исследователя, чем на ЛПР. Хотя автор предусмотрел подробные инструкции по использованию программного обеспечения, но руководитель предприятия, который не имеет большого опыта работы в области информационных технологий, может встретить немало трудностей при использовании этого инструмента.

Сложность использования «лАос» также усиливается тем, что его пользовательский интерфейс не предусматривает наглядного моделирования оцениваемой системы. Программный комплекс «лАос» предназначен для получения агрегированных оценок иерархических систем, но он не позволяет наглядно представлять структуру этих систем для просмотра. Представление результатов оценки состояния системы, а также значение потери эффективности отдельных её компонентов при расчёте осуществляется в виде текстового протокола работы программного комплекса (см. рис. 2).

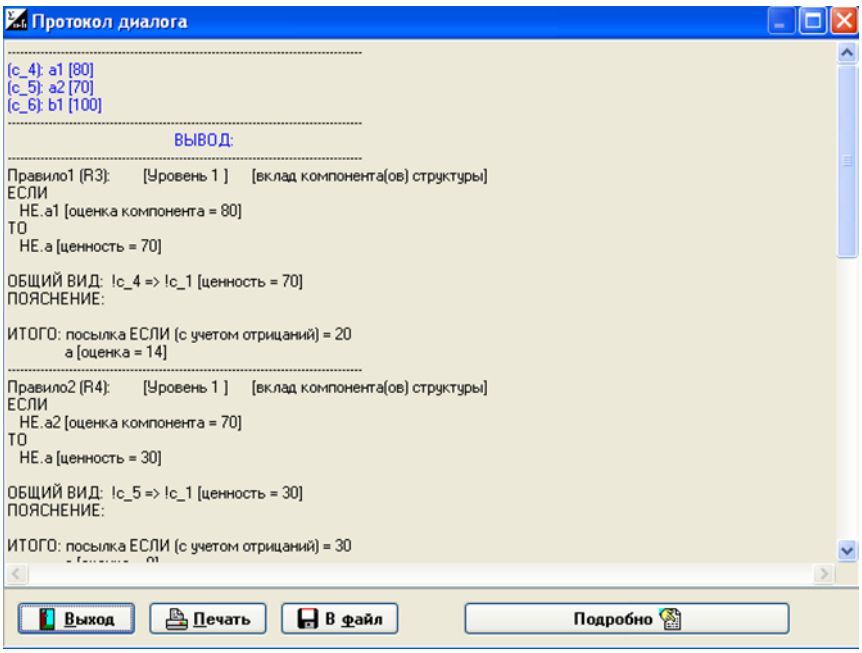


Рис. 2. Пример протокола работы «лАос 2.0»

Программная библиотека Балановского Д.К. Эта библиотека была реализована Балановском Д.К. в 2018 году в рамках его выпускной квалификационной работы (уровень магистратуры). Идея автора заключалась в разработке программной библиотеки, реализующей расчет агрегированных оценок на основе логико-аксиологического подхода, чтобы уже на основе этого можно было разрабатывать более продвинутые программные инструменты способные предоставлять пользователю возможность наглядного моделирования системы или расширяющие возможности применения данного подхода. Данная библиотека не предоставляет никакого графического интерфейса, только прикладной программный интерфейс (API). Эта разработка не может быть использована ЛПР на предприятии по понятным причинам.

Программно-инструментальная среда «ГрЛАО». Основываясь на опыте предыдущих разработок, авторами данной работы была предпринята попытка разработать программно-инструментальную среду «ГрЛАО» реализующую следующую функциональность для поддержки методики изложенной в [5]:

- визуальное моделирование структуры системы для просмотра в виде онтологической модели;
- оценивание эффективности работы системы на основе логико-аксиологического подхода;
- отображение входных данных и результата оценивания на графической модели структуры системы;
- сохранение модели построенной пользователем, для дальнейшей работы с нею;
- загрузка и редактирование ранее сохранённых моделей (добавление нового компонента, добавление связи между компонентами, удаление компонента и т.д.);
- выдача сценариев повышения эффективности целевого компонента.

Программно-инструментальная среда «ГрЛАО» реализована на платформе Java SE1.8 в виде настольного приложения. Для отрисовки графического интерфейса и обеспечения интерактивного взаимодействия пользователя с графическими объектами в ходе построения

модели структуры организации, используется ядро системы графического моделирования «GrModeling» (Свидетельство о государственной регистрации программы для ЭВМ №2013611154) разработанное Копайгородским А.Н. [8].

Иллюстрация работы «ГрЛАО». В рамках данной работы мы рассмотрим простой модельный пример для иллюстрации основных возможностей «ГрЛАО». Предположим, что филиал компании городского такси состоит из диспетчерского отдела и автопарка. Диспетчер отвечает за взаимодействие между клиентами и водителями: он получает звонки от клиентов, после чего ищет ближайшее к местонахождению клиента такси и если автомобиль свободен, то диспетчер свяжется с водителем и попросит его забрать клиента. Для простоты примера предположим, что в автопарке филиала имеется всего два автомобиля. Положим, что эффективность работы диспетчера составляет 90%. Это может быть установлено с помощью какой-либо методики оценки эффективности работы персонала (например, системы KPI этой компании), причем как именно – для нашего примера не принципиально. Пусть точно таким же образом руководитель филиала установил, что эффективность работы первой машины из автопарка составляет 50%, а второй – 60%. В рамках рассматриваемого примера поставим задачу определить эффективность работы филиала в целом, оценить его рабочее состояние и предложить меры по улучшению ситуации. На рисунке 3 представлена графическая модель структуры исследуемой системы, которая была построена в среде «ГрЛАО».

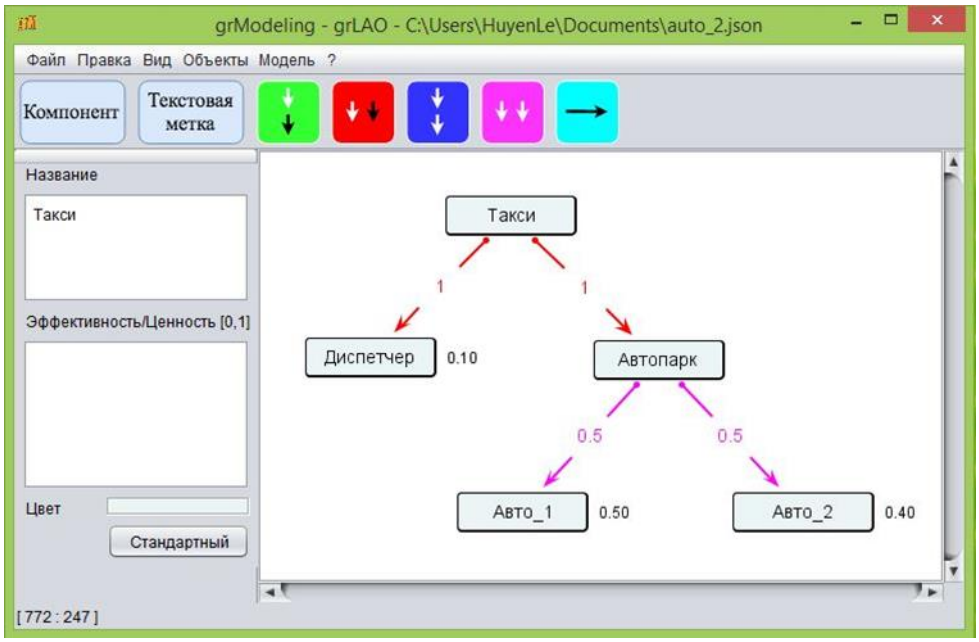


Рис. 3. Модель структуры исследуемой системы

Красные стрелки, соединяющие компоненты «Такси» и «Диспетчер», «Такси» и «Автопарк», обозначают вид отношения «разные работы – зависимость» между «Диспетчер» и «Автопарк». Данный тип отношений устанавливает прямую зависимость значения эффективности работы всего автопарка от эффективности работы диспетчера. Значение ценности связи между диспетчером и автопарком равно 1, так как для функционирования филиала такси нужны оба компонента, простой любого из них снижает эффективность работы всей системы до ноля. Сиреневые стрелки, соединяющие компоненты «Автопарк» и «Авто_1», «Автопарк» и «Авто_2», указывают на отношения «одинаковая работа –

независимость» между «Авто_1» и «Авто_2». Этот тип отношений подразумевает что «Авто_1» и «Авто_2» параллельно выполняют одну и ту же работу независимо друг от друга [6]. С точки зрения эффективности работы автопарка оба его компонента равноценны, что и отражено значениями 0,5 их отношений.

Результат оценки эффективности работы системы при заданных входных значениях, рассчитанной на основе логико-аксиологического подхода, представлен на рисунке 4.

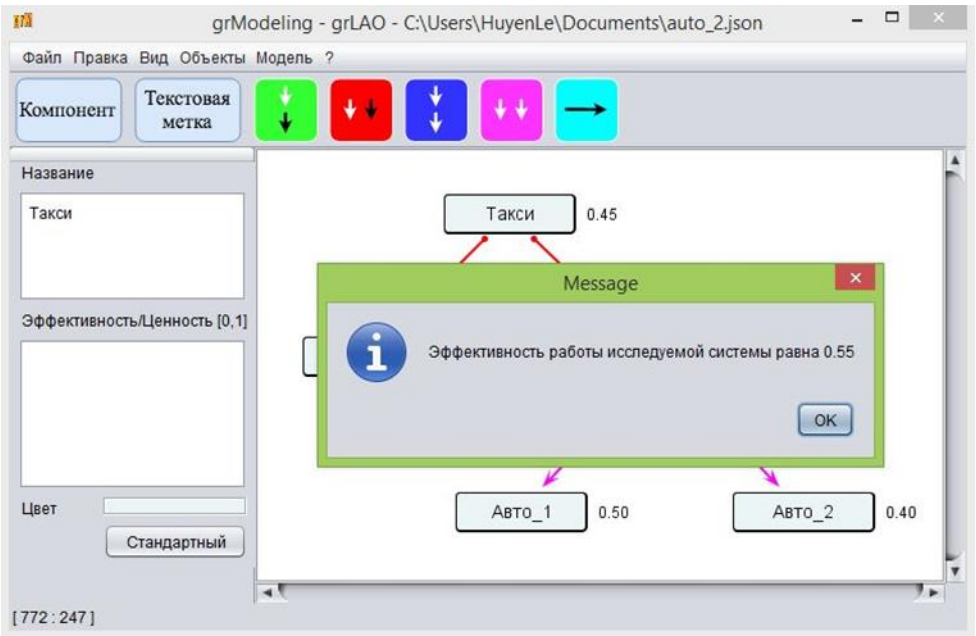


Рис. 4. Результат оценки эффективности работы системы

Как видно из результатов работы программы – эффективность работы филиала такси составляет 55%. Сравнивая это значение со шкалой для оценки состояния, мы определяем текущее значение эффективности филиала как «плохое» (см. рис. 5).

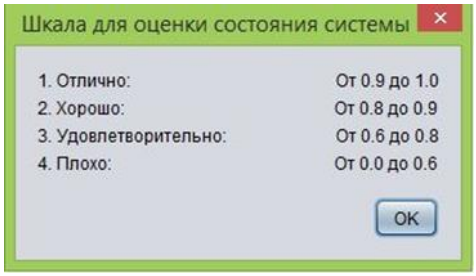


Рис. 5. Шкала для оценки состояния системы

С помощью инструмента «Поиск сценариев» среда «ГрЛАО» предлагает два сценария повышения эффективности работы филиала (см. рис. 6).

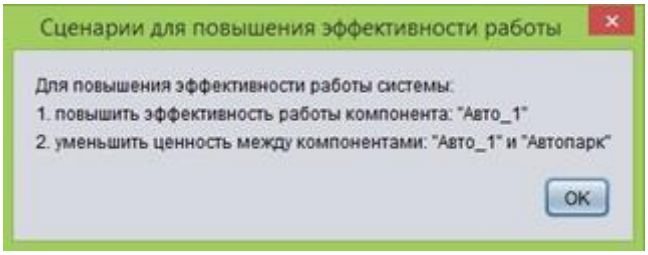


Рис. 6. Сценарии повышения эффективности системы

На рисунке 7 представлены результаты анализа системы в случае проведения мероприятий по реализации первого сценария. Конкретное содержание мероприятий –

замена водителя или автомобиля, создание мотивации для водителя работать эффективнее, или что-то иное – определяется ЛПР, исходя из его знаний и опыта работы.

В случае успешной реализации принятых мер (увеличение эффективности работы «Авто_1» до 0,7) эффективность работы филиала повысится до 65% («удовлетворительно» по принятой шкале).

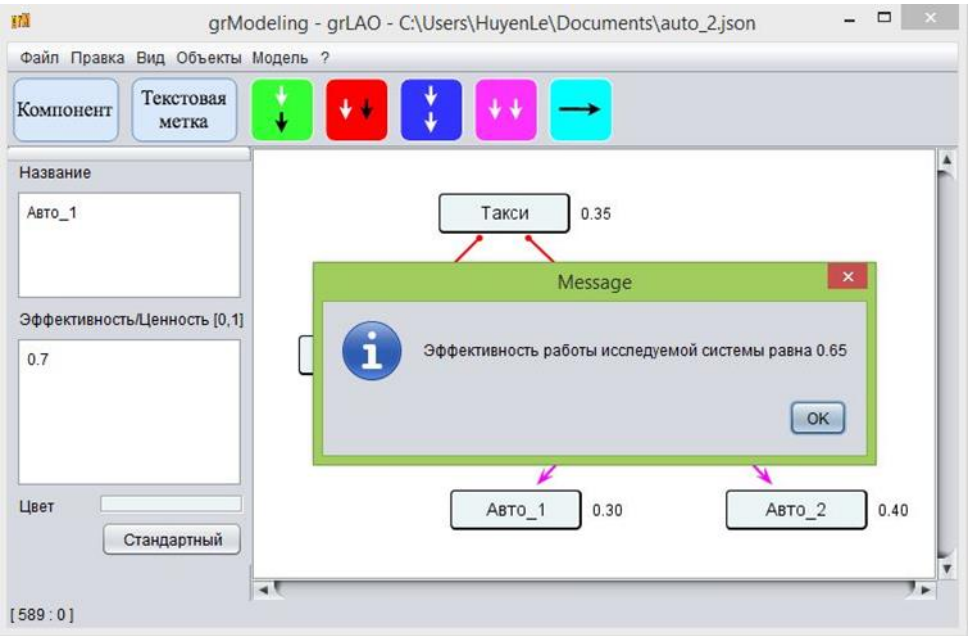


Рисунок 6 – Эффективность филиала после повышения эффективности работы «Первый автомобиль» от 0.5 до 0.7

Заключение. В статье представлены результаты разработки программно-инструментальной графической среды для оценки состояния системы на основе логико-аксиологического подхода («ГрЛАО»). Технология работы с реализованными программными инструментами рассмотрена на примере решения демонстрационной задачи повышения эффективности работы филиала городского такси.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. Панов М.М. Оценка деятельности и система управления компанией на основе KPI // М.: НИЦ Инфра-М, 2013. – 255 с.

2. Аршинский Л.В. Логико-аксиологический подход к оценке состояния систем // Современные технологии. Системный анализ. Моделирование. – Иркутск: ИрГУПС. –2013. – № 3(39). – С. 140–146.

3. Аршинский Л.В. Методика агрегированного оценивания систем с поддержкой ключевых компонентов // Онтология проектирования. 2015. №2 (16) [Электронный ресурс] // URL: https://cyberleninka.ru/article/n/metodika-agregirovannogo-otsenivaniya-sistem-s-podderzhkoy-klyuchevyh-komponentov (дата обращения: 15.05.2019).

4. Аршинский В.Л., Аршинский Л.В., Доржсурэн Хишигсурэн. Методика агрегированной оценки состояния производственно-экономической системы на примере станции Улан-Баторской железной дороги // Вестник ИрГТУ. 2018. Т.22. №2 (133). – С. 34 – 44.

5. Аршинский В.Л., Лэ Ван Хуен. Методика оценки качества работы организации на основе логико-аксиологического подхода // Молодёжный вестник ИрГТУ. 2019. Т. 9. № 1. – С. 7 – 14.

6. Балановский Д.К. Разработка программных средств для поддержки логико-аксиологического подхода к оценке состояния системы // Молодежный вестник ИрГТУ.2018. Т. 8. № 2 (25). – С. 7-11.

7. Аршинский В.Л., Аршинский Л.В., Доржсурэн Хишигсурэн. Оценка качества функционирования станции Улан-Баторской железной дороги как производственной системы на основе продукционной базы знаний // Информационные и математические технологии в науке и управлении. 2018. №2(10). – С. 61 – 70.

8. Копайгородский А.Н. Проектирование и реализация системы графического моделирования // Труды XV Байкальской Всероссийской конференции "Информационные и математические технологии в науке и управлении", Часть III. – Иркутск: ИСЭМ СО РАН, 2010. – С. 22-28.

УДК 004.054

МЕТОД ФОРМАЛИЗАЦИИ ТРЕБОВАНИЙ К ВХОДНЫМ ДАННЫМ

Камышова Е.А., Бахвалова З.А.
(г. Иркутск, Иркутский национальный исследовательский технический университет)
e-mail: leka96@mail.ru, zinand@inbox.ru

Аннотация. В статье рассмотрен ряд проблем, с которыми сталкиваются тестировщики при подготовке наборов тестовых данных. Известно, что для решения этой задачи используются способы подготовки данных, применение которых всегда влечет за собой необходимость структуризации требований, но ни один из существующих способов формализации не направлен на описание требований к входным данным программных продуктов с целью их дальнейшего преобразования в тестовые данные. В статье описан авторский метод формализации входных данных и приведен пример его использования. Использование этого метода позволит повысить качество подготовки тестовых данных

Ключевые слова. Тестирование ПО, требования к ПО, формализация требований, подготовка тестовых данных, входные данные ПО

FORMALIZATION METHOD FOR INPUT DATA REQUIREMENTS

Kamyshova E.A., Bahvalova Z.A.
(Irkutsk, Irkutsk National Research Technical University)
e-mail: leka96@mail.ru, zinand@inbox.ru

Abstract. The article discusses a number of problems that testers encounter in test data sets preparation. It is known that to solve this problem, methods of data preparation are used. Application of these methods always entails the need to structure requirements, but none of the existing formalization methods is aimed at describing the requirements for the input data of software products with the goal of further converting them into test data. The article describes the authors' method of formalizing input data and provides an example of its use. Use of this method will improve the quality of test data preparation.

Keywords. Software testing, software requirements, requirements formalization, test data preparation, software input data.

Введение. Тестирование программного обеспечения в настоящее время является неотъемлемой частью процесса разработки программных продуктов и имеет своей целью повышение качества программного обеспечения [1]. Данная статья посвящена описанию способа формализации требований к входным данным программных продуктов, что позволит повысить качество подготовки наборов тестовых данных Актуальность этой темы основана на том, что подготовка тестовых наборов данных является неотъемлемой составляющей тестирования каждой функции приложения, и поэтому повышение качества подготовки тестовых наборов данных ведет к повышению качества разрабатываемых программных продуктов.

Проблема подготовки тестовых данных. Одной из ключевых проблем при формировании тестовых данных является наличие ошибок, следующих из неверного восприятия или описания требований к допустимым и недопустимым входным данным программных продуктов, и формализация этих требований позволит решить обозначенную проблему.

Независимо от вида осуществляемого тестирования, специалисты тратят много времени на то, чтобы подготовить наборы тестовых данных (то есть данных, которые будут использоваться в процессе тестирования), что зачастую является даже самой сложной задачей в процессе тестирования. Для подготовки тестовых данных тестировщикам требуется проанализировать требования к определенной функциональности, которая подвергается тестированию, составить все возможные комбинации допустимых и недопустимых входных данных, а затем минимизировать полученный перечень комбинаций, отказываясь от схожих между собой наборов данных, а также от таких наборов, которые не могут быть получены ни при каком сценарии использования приложения. При этом существует риск допустить ошибку ввиду человеческого фактора, в случае возникновения которого вся проделанная работа оказывается выполненной впустую и потребует повторных временных затрат.

Существуют техники, которые облегчают и систематизируют подготовку данных для тестирования, как ручные [2,3], так и автоматизированные [1]. Специалист должен самостоятельно проанализировать требования, структурировать и детализировать их (иначе говоря – формализовать); затем для каждого из них составить наборы данных, как обеспечивающих верное выполнение задачи, так и приводящих к исключительной ситуации с обработкой полученной ошибки. Эти наборы затем комбинируются либо вручную, либо с использованием программных средств.

В настоящее время существует ряд способов, которые позволяют формализовать некоторые требования:

- требования к поведению [4, 5] системы, которые можно использовать для описания сценариев и для отслеживания выполнения требований и привязки к ним задач, людей, кода и т.д.
- требования к функциональности [6], то есть если система отнесена к определенному классу систем, она должна иметь определенную функциональность
- требования к структуре [7] хранимой информации (статическая структура)

Ни один из существующих способов формализации не направлен на описание требуемых для тестирования входных данных программных продуктов. Поэтому возникла необходимость разработки такого способа формализации требований, который позволит структурировать описание входных данных программных продуктов и повысить качество подготовки на их основе тестовых наборов.

Способ формализации требований, описанный ниже, позволит создать механизм формального описания требований к входным данным программного продукта таким образом, чтобы при осуществлении тестирования преобразовывать эти требования в тестовые данные с как можно меньшим количеством ошибок.

Описание способа. Основой предлагаемого способа является то, что для каждого требования, описывающего допустимые и недопустимые значения входных данных, можно выделить атрибуты (количественные характеристики требований) и атрибуты-связи, описывающие зависимость данного требования от других требований. Поскольку атрибуты могут быть многуровневыми, принята следующая иерархия обозначений:

атрибут {параметр {свойство}}

То есть атрибут – это высокоуровневая характеристика требования, параметр – это характеристика, описывающая атрибут, а свойство описывает параметр.

Кроме того, если атрибут имеет не одно значение, а несколько, или же он описывается набором однотипных параметров, то можно использовать квадратные скобки, обозначающие массив:

атрибут: [
  значение1,
  значение2
]

Или:

атрибут: [
  параметр: значение1,
  параметр: значение2
]

Аналогично можно описывать наборы параметров и свойств при необходимости.

Предлагаемый способ формализации требований к входным данным предоставляет возможность:

- подробно описать требования к входным данным любого типа (строки, числа, файлы и т.д.);
- организовать вертикальную зависимость требований друг от друга (т.е. описать те случаи, когда допустимые значения одного входного параметра зависят от значения другого), а также иерархию свойств внутри требования (например, для чисел часть описываемых атрибутов будет отличаться от строк);
- организовать горизонтальную взаимосвязь (через объединение полей в группы можно проиллюстрировать их связь и совместное использование);
- применяться к системам любого класса, независимо от наличия графического интерфейса и назначения.

Входные данные, то есть данные, вводимые в систему обработки информации, могут быть введены посредством:

- заполнения значений в полях на форме (если предусмотрен графический интерфейс);
- вызова функции с определенными параметрами (если программа не имеет графического интерфейса или нужно протестировать отдельную программную функцию).

Поэтому далее для описания объекта, предназначенного для ввода данных, будет использоваться термин «поле», а для места, содержащего набор полей – термин «форма».

Поскольку методика основана на том, что каждое требование, определяющее входные данные системы, может быть описано набором атрибутов, характеризующих само требование и его связи с другими требованиями, требуется определить ряд таких характерных атрибутов:

- уникальный идентификатор поля – значение, характеризующее поле, как правило является именем поля, например, «телефон» или «пароль». Его необходимо указывать для того, чтобы и человек, и машина могли определять, к какому полю относятся все перечисляемые далее атрибуты;
- статус обязательности поля – определяет, допустимо ли оставить поле незаполненным, или ввод значения обязателен. Как правило, обязательность приходится определять для каждого поля, так как от этого зависит дальнейшая обработка его значения системой;
- количество значений – необходимо с использованием параметров задать минимальное и максимальное количество значений, которое можно ввести в одно поле, а также определить разделители между отдельными значениями, так как работа приложения может подразумевать ввод нескольких экземпляров данных в одно и то же поле;
- формат – определяет, каким образом поле заполняется значениями, например, выбираются ли они из определенного списка или вводятся вручную; от формата будет зависеть, какие дополнительные атрибуты потребуется указать для того, чтобы охарактеризовать допустимые параметры вводимого значения.

Формат имеет важное значение при описании требований, поскольку с его определения начинается преобразование качественных характеристик описываемых требованиями сущностей в количественные. Таким образом, тестировщик при работе с тестовыми данными не будет анализировать требования в виде «правильная строка» – само слово «правильная» будет преобразовано к ряду свойств, описываемых количественными величинами.

В качестве форматов ввода значений в поля можно выделить:

- перечисление – выбор значения (или нескольких) из определенного списка, помимо которого никакие другие данные недопустимы;
- файл – загрузка файла, с которым далее будет работать приложение;
- дата (или дата и время) –значение (выбираемое с использованием графических компонентов или вводимое самостоятельно) является либо датой, либо датой с определенным временем;
- вводимое вручную – значение вводится пользователем в соответствии с определенными правилами, по которым впоследствии осуществляется проверка

указанного значения; этот формат используется в том случае, если не подходит ни один из предыдущих, например, если необходимо описать числовое значение в определенном диапазоне. Этот формат является наиболее гибким и подходит для описания любых чисел и строк.

Для каждого из перечисленных форматов значение описывается своим набором атрибутов, дополняющим перечисленные выше универсальные атрибуты, который позволяет описать все характеристики выбранного формата входных данных. Самым сложным среди перечисленных является описание поля с форматом «вводимое вручную»: его значение описывается составным атрибутом «валидация», определяющим, каким образом проверяется вводимое в поле значение (иначе говоря, какие требования к нему предъявляются).

В свою очередь, атрибут «валидация» включает в себя параметр «тип», принимающий два возможных значения:

- по регулярному выражению – этот вариант используется в том случае, когда вводимое значение должно соответствовать определенному шаблону, то есть регулярному выражению;
- по разрешениям и ограничениям – применяется для описания числовых и строковых значений, которым требуется наиболее гибкое описание (допустим, для описания числа, находящегося в промежутке от -10 до 10, не включая 0, или строки, состоящей из символов кириллицы, не начинающейся с гласной буквы). Под разрешением понимается набор параметров, которым должно удовлетворять поле. Под ограничением – набор, которому поле, напротив, не должно удовлетворять.

В зависимости от выбранного типа определяются другие параметры, входящие в атрибут «валидация», и описывающие их свойства.

Все атрибуты, которые были перечислены выше, служат для описания самостоятельных, не зависимых друг от друга требований. Но кроме них описываются также атрибуты-связи, которые характеризуют наличие связи данного требования с другими:

- группа, к которой принадлежит поле – используется для горизонтальной взаимосвязи полей, то есть в тех случаях, когда поле (или несколько полей) с одинаковыми характеристиками могут появляться на форме несколько раз. Примером может являться группа, содержащая несколько полей для ввода контактных номеров телефонов: все они допускают ввод информации в строго заданном формате, и их значения не зависят друг от друга;
- статус зависимости поля – используется для описания зависимости значения одного поля от другого (т.н. вертикальная зависимость). Например, если на форме есть два поля, «тип доставки» и «дата доставки», а «тип доставки» может принимать значения «курьер» и «самовывоз», то значение поля «дата доставки» будет зависеть от выбранного типа: для варианта «курьер» это может быть любая дата, а для «самовывоза» – пустое значение.

Использование групп позволяет избегать повторения описаний требований в случае, если несколько полей имеют одинаковые характеристики. Как правило, это одно поле (например, на форме может быть 2 или 3 одинаковых поля для указания контактного номера телефона), но их может быть и несколько (к примеру, юридический и фактический адрес организации могут включать в себя поля «страна», «индекс», «город», «адрес»).

Зависимости полей указываются в следующих случаях:

- значение данного поля зависит от значения, которое было указано на этой же форме;
- значение данного поля зависит от существования в системе экземпляра какой-либо сущности, ранее описанной требованиями (например, на форме создания виртуальной машины (ВМ) в списке виртуальных дата-центров, в которых доступно создание ВМ, есть только созданные ранее дата-центры);
- значение данного поля зависит от наличия хотя бы одного экземпляра какой-либо сущности с подходящим параметром (к примеру, если есть хотя бы один метод оплаты через банковскую карту, можно выбрать его в поле со способом оплаты);
- значение данного поля зависит от наличия конкретного экземпляра какой-либо сущности с подходящим параметром (например, если для определенного метода оплаты указан минимальный платеж 100 рублей, то в данное поле можно ввести число от 100 рублей).

Для того, чтобы проще описывать то, какое поле и от чего зависит, необходимо ввести терминологию:

- родительское поле – поле, от значения которого зависит данное поле;
- дочернее поле – поле, значение которого зависит от данного поля;
- родительская сущность – сущность (форма, объект), от которой зависит значение данного поля.

В случае, если в качестве атрибута «статус зависимости» указывается значение «зависимое», состав атрибутов поля значительно меняется, поскольку теперь все значения описываются в зависимости от родительских полей и сущностей. Поэтому, за исключением атрибута «группа» и «статус зависимости», никакие из ранее перечисленных атрибутов не используются. Их заменяют другие:

- родительские элементы – перечень полей, сущностей и значений их параметров, которые оказывают влияние на дочернее поле. Они будут использоваться для дальнейшего описания зависимостей;
- значения – составной атрибут, в который входит массив зависимостей – правил, по которым на основании родительского поля определяется значение дочернего.

Каждая зависимость в рамках значения включает в себя два параметра: «если» и «то». «Если» содержит перечень условий, которым должно удовлетворять родительское поле / родительская сущность, чтобы дочернее поле имело значение, описанное в «то»; эти условия описываются по-своему для каждого формата родительского поля.

При этом параметр «то» включает в свой состав те же атрибуты, что описывают обычное независимое поле, но в данном случае они уже являются свойствами:

- статус обязательности;
- количество значений;
- формат;
- значение, описываемое различными атрибутами на основании выбранного формата.

Если требования к входным данным записать в соответствии с перечисленными количественными атрибутами и атрибутами-связями, будет получено полное, недвусмысленное описание всех характеристик, которыми должны обладать входные данные в зависимости от их типа.

Пример. Предположим, с помощью предложенной методики необходимо описать следующие требования: «Способ связи – обязательное, email или телефон. Контактные данные – обязательное, описываются в зависимости от выбранного способа связи; если способ связи – email, то должен быть введен email, если телефон, то вводится телефон. Проверка осуществляется по регулярному выражению».

Тогда в формализованном виде эти требования будет иметь вид:

```
"Способ связи": {
   "Зависимое": "независимое",
   "Обязательное": "обязательное",
   "Формат": "перечисление",
   "Количество значений": {
     "Допустимо несколько значений": "нет"
   },
   "Допустимые значения": [
     "email",
     "телефон"
   ]
},
"Контактные данные": {
   "Зависимое": "зависимое",
   "Родительские элементы": [
     "Способ связи"
   ],
   "Значения": [
     {
       "Зависимость1": {
         "Если": {
           "Условия": [
             {
               "Условие1": {
                 "Родительский элемент": "Способ связи",
                 "Признак": "Точное совпадение с родителем",
                 "Значение": [
                   "email"
                 ]
               }
             }
           ]
         },
         "То": {
           "Обязательное": "обязательное",
           "Формат": "вводимое вручную",
           "Количество значений": {
             "Допустимо несколько значений": "нет"
```

```
          },
          "Валидация": {
            "Тип": "по регулярному выражению",
            "Регулярное выражение": [
              "[-a-z0-9_.]+@([-a-z0-9]+\\.)+[a-z]{2,6}(\\s|$)"
            ]
          }
        }
      }
    },
    {
      "Зависимость2": {
        "Если": {
          "Условия": [
            {
              "Условие2": {
                "Родительский элемент": "Способ связи",
                "Признак": "Точное совпадение с родителем",
                "Значение": [
                  "телефон"
                ]
              }
            }
          ]
        },
        "То": {
          "Обязательное": "обязательное",
          "Формат": "вводимое вручную",
          "Количество значений": {
            "Допустимо несколько значений": "нет"
          },
          "Валидация": {
            "Тип": "по регулярному выражению",
            "Регулярное выражение": [
              "[8-9]{1}[0-9]{9}"
            ]
          }
        }
      }
    }
  ]
}
```


Заключение. Описанный метод предлагает подробное описание формальных требований, которое направлено на повышение качества подготовки тестовых данных. Во-

первых, тестировщик получает полную информацию о параметре, позитивные и негативные значения которого ему необходимо сформировать и использовать для подготовки тестовых наборов. Во-вторых, после формализации явно выделяются все взаимосвязи значений входных данных, которые ранее могли быть не выявлены или не описаны.

Несомненно, методика имеет свои недостатки:

- она не может быть применена для описания поведения системы, будучи направленной исключительно на описание данных;
- она не позволяет описать требования к содержанию входных данных, представленных в виде файлов.

Но в то же время методика имеет и значительные преимущества:

- структурирует процесс формализации требований (перевод качественных требований в количественные) с учетом связей, существующих между требованиями;
- описывает требования к входным данным единым формальным языком;
- в дальнейшем позволит автоматизировать процесс формализации требований к входным данным;
- допустимо использование не только на этапе тестирования, но и на этапе формирования требований.

Однако в своем настоящем виде она удовлетворяет поставленной цели – повышению качества подготовки тестовых данных.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. Куликов С.С. Тестирование программного обеспечения // Базовый курс: практ. пособие. – Минск: Четыре четверти, 2015. – 294 с.

2. Глоссарий международной организации по сертификации тестирования программного обеспечения ISTQB: [Электронный ресурс] // URL: http://glossary.istqb.org/ (дата обращения: 10.05.2019).

3. Савин Р. Тестирование Дот Ком, или Пособие по жестокому обращению с багами в интернет-стартапах // М.: Дело, 2007. – 312 с.

4. Lee Copeland. A Practitioner's Guide to Software Test Design // Artech House, 2004. – 358 с.

5. Липко Ю.Ю. Алгоритм формализации требований при разработке информационных систем // Известия Южного федерального университета. Технические науки. – 2014. – №6(155). – С. 153-158.

6. Минюров С., Минюрова А. Формализация требований на основе базы знаний. Разработка требований для программного обеспечения // М: 2012. – 55 с.

7. Веряев А.С., Фадин А. А. Формализация требований безопасности информации к средствам анализа защищенности // Вопросы кибербезопасности. – 2015. – №4(12). – С. 23-27.

8. Формализация требований на практике. В.В. Кулямин и др. [Электронный ресурс] // URL: http://panda.ispras.ru/~kuliamin/docs/Req-2006-ru.pdf (дата обращения: 05.01.2018).

УДК 621.373.54

РАЗРАБОТКА И ИССЛЕДОВАНИЕ ЦИФРОВОГО ГЕНЕРАТОРА СЛУЧАЙНЫХ ПОСЛЕДОВАТЕЛЬНОСТЕЙ

Копылов М.М.
(г. Иркутск, Иркутский национальный исследовательский технический университет)
e-mail: michail.kopylov.94@mail.ru

Аннотация. В данной работе рассматривается разработка и исследование генератора случайных последовательностей на ПЛИС.

Ключевые слова. ГПСЧ; генератор случайных чисел.

DEVELOPMENT AND RESEARCH OF DIGITAL GENERATOR OF RANDOM SEQUENCES

Kopylov M.M.
(Irkutsk, Irkutsk National Research Technical University)
e-mail: michail.kopylov.94@mail.ru

Abstract. In this paper, we consider the development and study of the random sequence generator on FPGAs.

Keywords. Field-Programmable Gate Array (FPGA), random number generators.

Введение. Генерация действительно случайных чисел играет очень важную роль в самых разных приложениях – в криптографии, в области численного моделирования, в игровой индустрии и других областях. В последние пятьдесят лет, в связи с расширением области применения компьютеров число таких приложений постоянно растет.

Генератор последовательности случайных чисел (ГПСЧ) – один из ключевых элементов любой криптографической системы. Большинство криптографических приложений, защищающих транзакции, используют генераторы случайных чисел для создания одноразовых или сеансовых ключей. Однако именно используемые генераторы являются самым слабым местом в большинстве систем шифрования. Это связано с тем, что программные генераторы полностью детерминированы и получаемые последовательности являются псевдослучайными и, в принципе, пусть и с трудом, могут быть предсказаны.

Качественные генераторы истинно случайных последовательностей основываются на различных физических процессах и явлениях, таких как радиоактивный распад, шумы аналоговых сетей, тепловые шумы, флуктуационные шумы и т.д.

В данной работе реализованы цифровые методы измерения параметров процессов перехода из одного стабильного состояния (например, n логическая «1») в другое стабильное состояние (например, логический «0») через пребывание в «запрещенной зоне». Параметры запрещенной зоны зависят от: напряжения питания, температуры, трассировки элементов схемы в ПЛИС и других неконтролируемых параметров. Все это выражается в случайную зависимость, как частоты, так и длительности импульсов, генерируемых ядром ГПСЧ по нижеприведенной схеме (рис. 1).

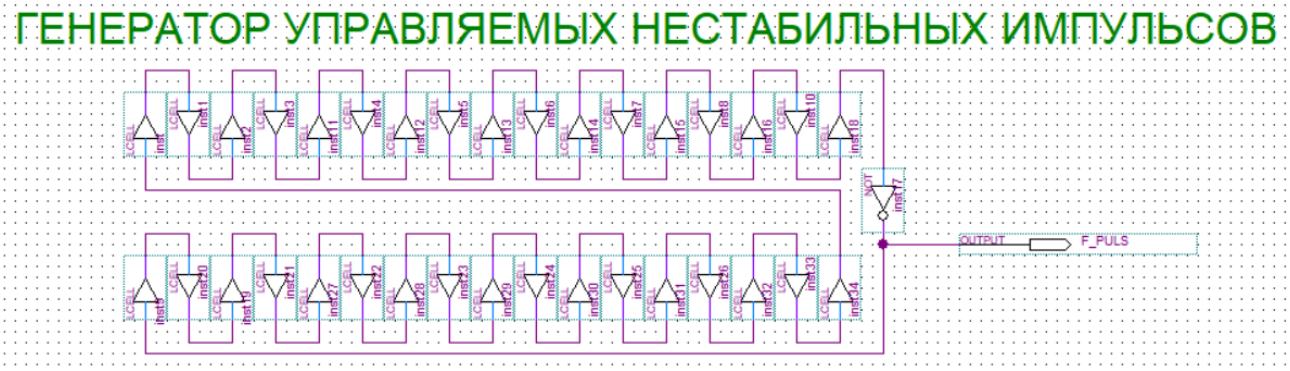


Рис. 1. Принципиальная схема ядра ГПСЧ

Полная схема генератора приведена на рис. 2. Длительность импульсов определяется числом элементов LCELL ядра генератора и составляет приблизительно 1мс. Измерение периода следования случайных импульсов (T) осуществляется путем подсчета числа опорных импульсов, вырабатываемых кварцевым синтезатором частоты PLL (500МГц), с помощью счетчика CNT, запускаемого фронтом каждого случайного импульса.

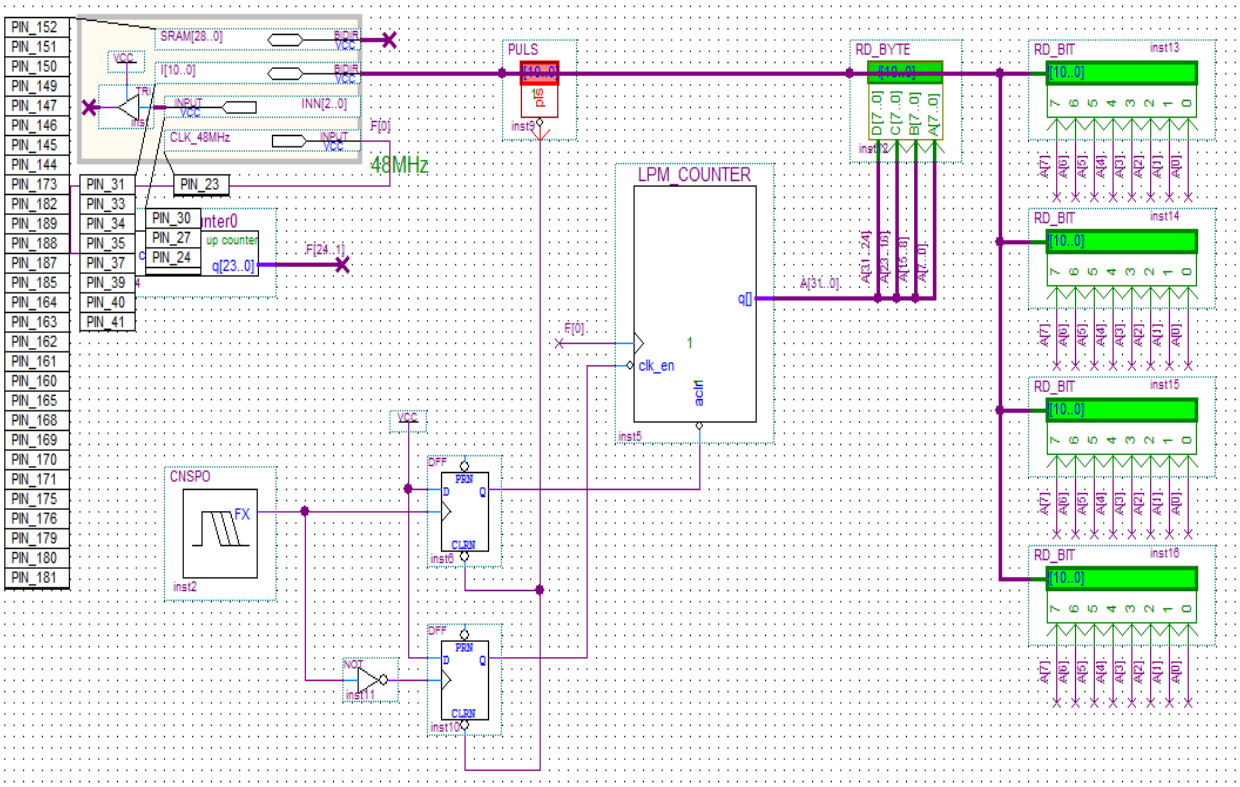


Рис. 2. Схема генератора ГПСЧ

Также была написана программа на Delphi для получения и сохранения файла с данными. Интерфейс программы представлен на Рис.3.

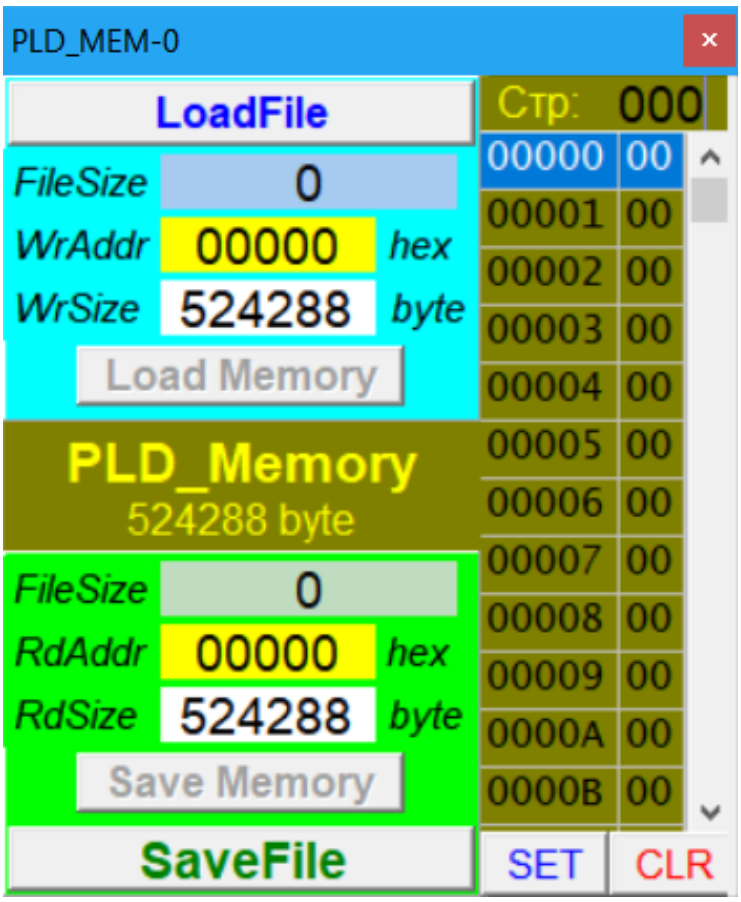


Рис. 3. Интерфейс созданной программы

Анализ данных. После генерации последовательности импульсов устройство сохраняет ее в файл. С этой последовательностью проводим статистические тесты для оценки ее характеристик.

Графические тесты. Графические методы статистического анализа последовательностей являются весьма эффективными при выявлении существенных недостатков генерируемой последовательности. С их помощью можно быстро отбросить ГПСЧ, чьи результаты не удовлетворяют критериям.

Гистограмма распределения элементов. Данный тест позволяет оценить равномерность распределения символов в исследуемой последовательности, а также определить частоту появления конкретного символа.

Для анализа характеристик ГСЧ, используем полученный ранее файл, содержащий полученные данные в необработанном виде. На рис. 4. показаны данные в программе Microsoft Excel.

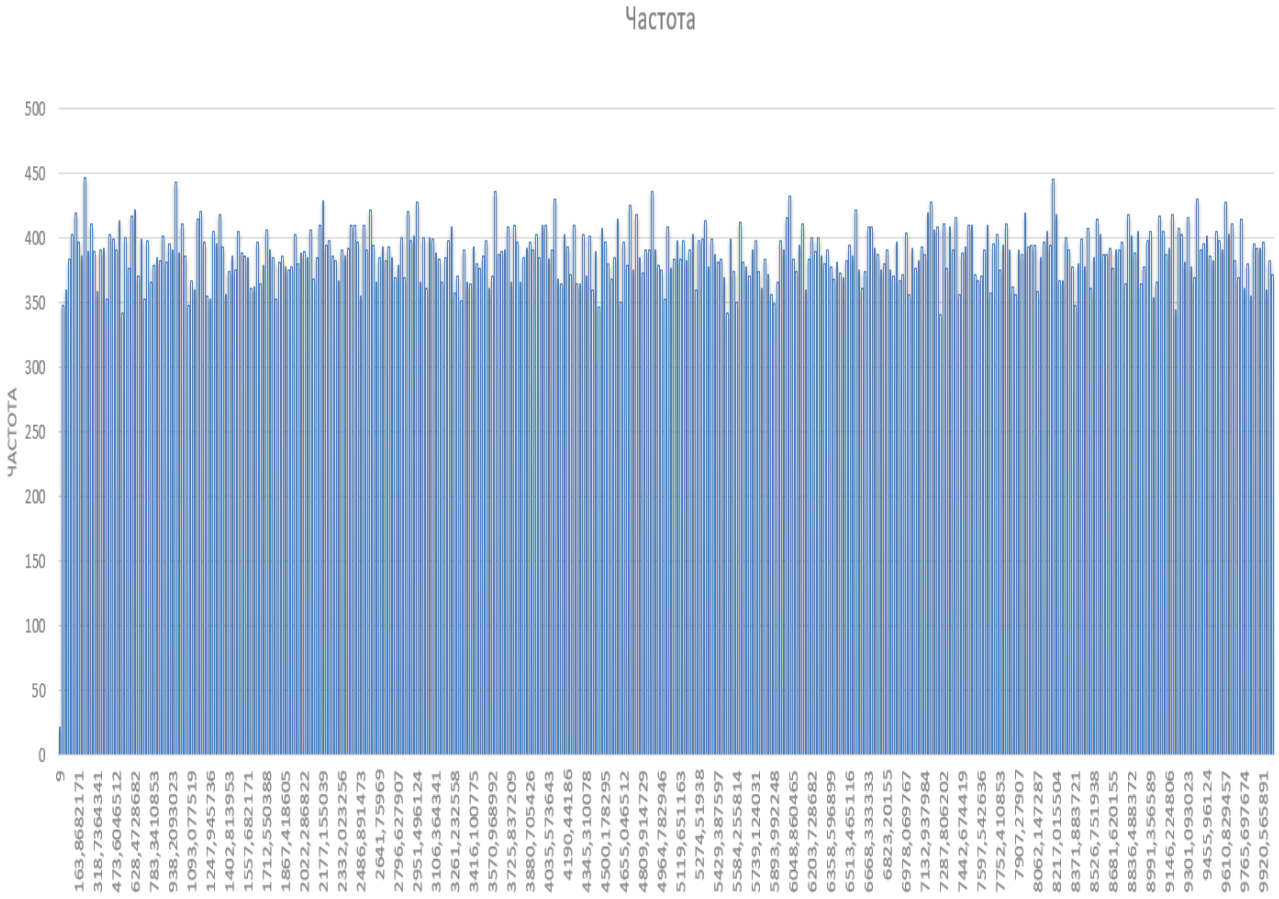


Рис. 4. Частотное распределение для полученной последовательности

Дополнительно произведем суммирование по 100 значений, согласно центрально предельной теореме, новое распределение суммы случайных величин, распределенных по равномерному закону, должно быть близко к нормальному.

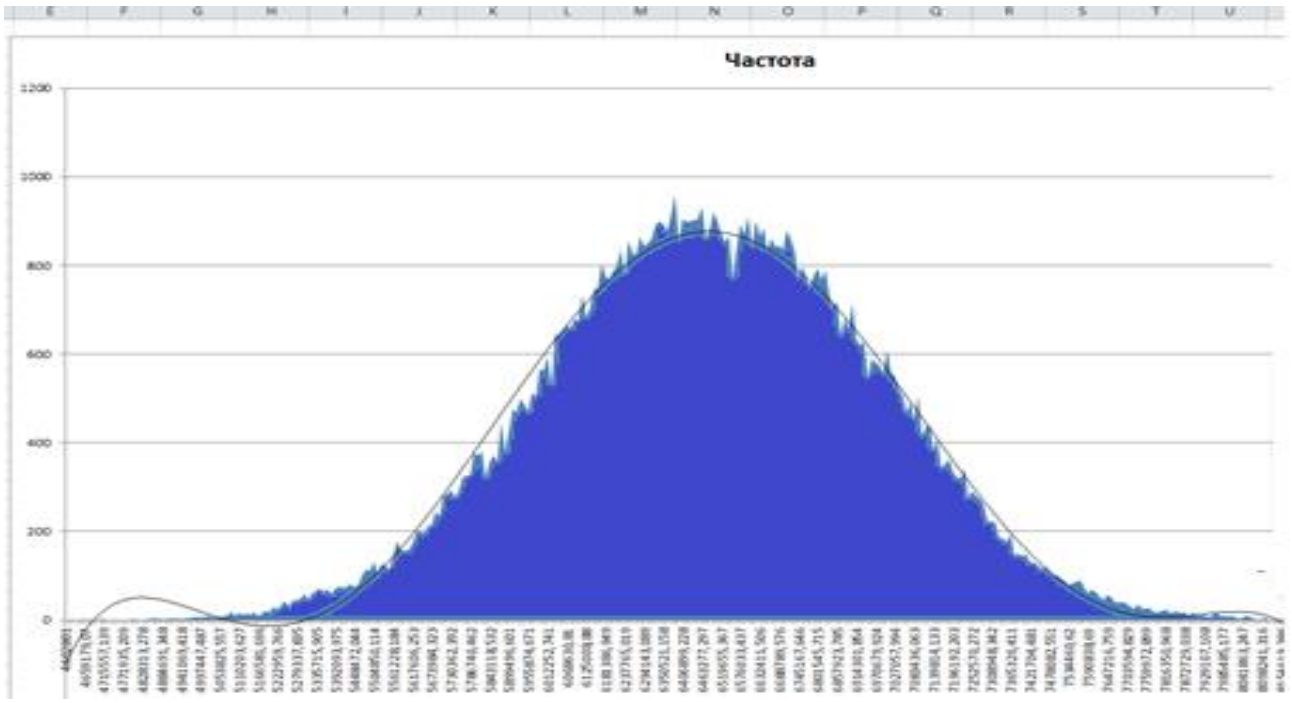


Рис. 5. Распределение, полученное с ЦПТ

Следовательно, можно сделать вывод, что все числа в последовательности встречаются с примерно одинаковой вероятностью.

Заключение. Достоинством рассмотренного ГПСЧ является использование для получения случайных характеристик исключительно цифровых элементов, входящих в состав любых вычислительных систем.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. Мухин О. И. Моделирование систем: Электронный учебник // Пермь: Изд-во ПГТУ, 2001.
2. Кнут Д.Э Искусство программирования: Том 2 // Изд-во Вильямс, 778 с.
3. Лопатин Д.В., Чиркин Е.С. Центр компьютерной безопасности – точка профессионального роста выпускников вуза // Вестник Тамбовского университета. Сер.: Гуманитарные науки. Тамбов, 2008. Вып. 11. С. 45-50.
4. Вентцель Е. С. Теория вероятностей: учебник для вузов — 10-е изд., стер. // М. : Высшая школа, 2006.

УДК 004.94

ОСОБЕННОСТИ РЕАЛИЗАЦИИ ВЕРОЯТНОСТНО-СОБЫТИЙНОГО ПОВЕДЕНИЯ АГЕНТОВ-ЭВАКУИРУЮЩИХСЯ В КОМБИНИРОВАННОЙ МОДЕЛИ РАЗВИТИЯ СОБЫТИЙ ПРИ ПОЖАРЕ

Куклин Е.В.
(г. Иркутск, Иркутский национальный исследовательский технический университет)
e-mail: brainer1995@gmail.com

Аннотация. В статье описывается процесс и результат разработки вероятностно-событийной модели поведения агентов в условиях чрезвычайной ситуации (эвакуации при пожаре) – принцип отбора формирующих поведение параметров, их реализация в модели и взаимодействие с моделью распространения пожара.

Ключевые слова. Событийное моделирование, пожарно-тактическая методика расчета ОФП, Unity 3D.

IMPLEMENTATION FEATURES OF PROBABILISTIC-EVENT BEHAVIOR OF EVACUATING AGENTS IN THE COMBINED MODEL OF THE COURSE OF FIRE

Kuklin E.V.
(Irkutsk, Irkutsk National Research Technical University)
e-mail: brainer1995@gmail.com

Abstract. The article describes the process and the result of developing a probabilistic-event model of behavior of agents in an emergency (fire evacuation). The principle of selection of parameters that form behavior, their implementation in the model and interaction with the fire propagation model are described.

Keywords. Event modeling, fire-tactical calculation methodology of fire hazards, Unity 3D.

Введение Моделирование распространения пожара и оценка пожарного риска для сооружений – важный компонент современного подхода к обеспечению пожарной безопасности.

Существует большой набор программных комплексов [1-3], ориентированных на данную задачу, но, как правило, их функциональность ограничивается воссозданием непосредственно пожара, когда как такой аспект, как процесс эвакуации из здания и особенности поведения попавших в чрезвычайную ситуацию людей в них либо опускается совсем, либо ему уделено минимальное внимание – расставляемые по зданию агенты способны только найти и определить ближайший выход.

В свою очередь, те программные продукты, которые ориентированы на моделирование эвакуации [3, 4], не учитывают пожар и то влияние, которое распространение огня, дыма, отравляющих веществ и испытываемая эвакуирующимися стрессовая нагрузка оказывают на ход эвакуации. Такие факторы, как перекрытие маршрута к эвакуационному выходу вследствие задымленности или движения фронта горения, паника, как отдельных людей, так и групп, результирующая в образовании толпы и потенциальное незнание структуры здания в реальных условиях способны сильно затруднить эвакуацию или вовсе сделать ее невозможной.

Таким образом, ориентиром при разработке комбинированной модели, описанной в данной статье, было объединение моделирования пожара и моделирования поведения агентов, с привязкой поведенческой модели к изменяющейся среде, в которой агенты находятся.

Краткое описание комбинированной модели. Для сочетания двух моделей и возможности наблюдения за происходящим в реальном времени используется упрощенная модель пожара, созданная на основе теории пожарно-тактической методики. Это позволяет проводить расчеты одновременно (более сложные модели пожара позволяют получить только впоследствии воспроизводимый, но не интерактивный результат) и сохранить приемлемое быстродействие, при этом имея на модели распространяющиеся огонь и задымление.

Для визуализации работы модели с использованием движка Unity 3D и скриптовой среды Microsoft Visual Studio 2017 создан специализированный программный продукт. Возможности Unity по быстрой привязке скриптов к объектам через систему компонентов и встроенная система навигации, реализующая перемещение объектов по поверхностям, позволили успешно получить работоспособных и при необходимости редактируемых «на ходу» агентов.

Единицей модельного времени взята секунда, и ежесекундное обновление значения верно для всех параметров агентов и других объектов, кроме координат при движении.

На рисунке 1 приведено изображение, иллюстрирующее основной экран программы – окно показа симуляции.

Параметры агентов и их назначение. Параметры агентов отбирались как согласно имеющимся теоретическим сведениям об особенностях поведения людей в условиях ЧС [5-7], так и, отталкиваясь от собственной концепции того, как должна работать разработанная модель. В качестве примера аналога можно привести модель, используемую программой ZET Evacuation Tool [4] (рис. 2).

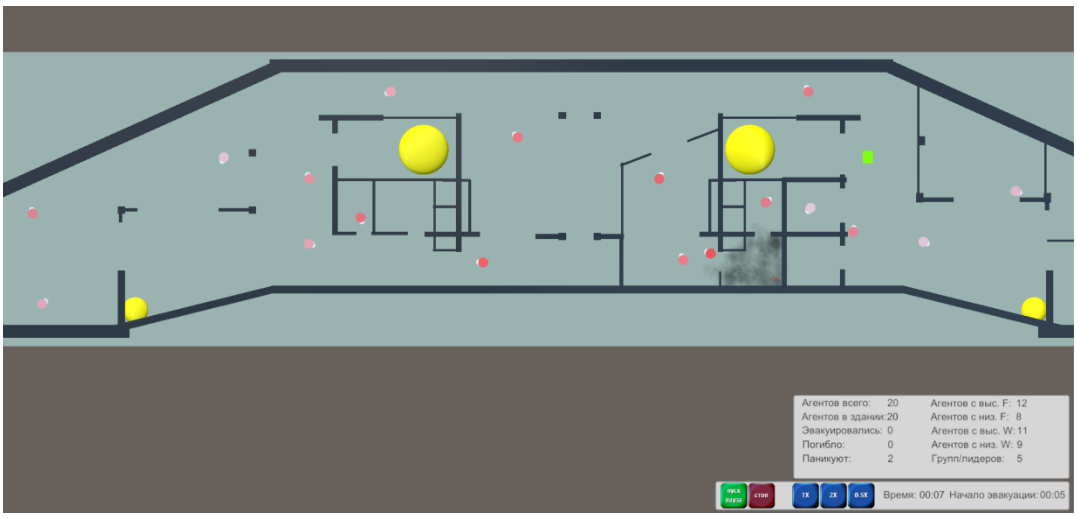


Рис.1. Основной экран программы

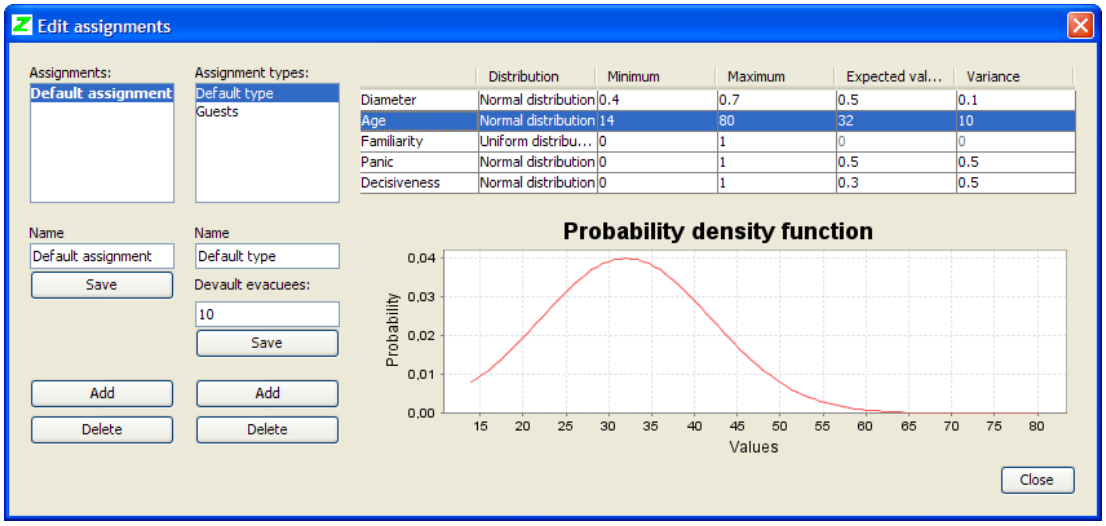


Рис.2. Окно настройки параметров агентов в ZET Evacuation Tool

Для эвакуирующихся в ZET настраиваемы следующие параметры:

– возраст (Age);
– степень осведомленности о структуре сооружения (Familiarity);
– склонность к панике (Panic) и волевых качеств (Decisiveness).

Рассмотрим подробнее два из них – осведомленность и волевые качества, так как они также присутствуют и в собственной модели.

Назначение параметра осведомленности (далее F) в модели, ориентированной на имитацию поиска эвакуационного выхода в потенциально незнакомом здании, очевидно. Вполне естественно предположить, что знакомые со зданием люди смогут без дополнительной информации (указатели направления движения или схемы эвакуации) добраться до ближайшего к их позиции в здании эвакуационного выхода, когда как эвакуирующимся с низкой F, напротив, необходимы подобные сведения или возможность следовать за другими агентами [5].

В комбинированной модели F задается дискретно, со значениями от 0 (здание незнакомо) до 3 (здание хорошо знакомо). С точки зрения реализации на движке F определяет, будет ли поиск агентом пути до выхода просто отбором ближайшего выхода из имеющихся (построение навигационной системой Unity набора путей, отбор наименьшего

расстояния и выбор соответствующего пути) или же «блужданием» по зданию, полуслучайным (при промежуточных значениях поиск эвакуационного выхода ведется в некотором радиусе от его действительной позиции), случайным или, при наличии навигационных объектов – вышеупомянутых указателей и схем – недостаток осведомленности агента компенсируется за счет возможности получить верное направление движения или узнать планировку здания.

Волевые качества, W (от англ. Willpower) в данной модели определяют подверженность агента панике и его способность сохранять спокойное состояние в чрезвычайной ситуации. От волевых качеств зависит ежесекундное изменение показателя эмоционального состояния.

Уровень эмоционального состояния, S – обозначает степень нервного напряжения агента и близость к панике. Представляет собой действительное число, которое варьируется от 0 до 1 с точностью до сотых.

Значение от 0 до 0.45 – комфортное состояние, 0.45 – 0.68 – спокойное, 0.68 – 0.7 – активное, больше 0.7 – повышенной активности [6]. Агенты окрашиваются в цвета от белого (0) до ярко-красного (1) для наглядного представления их S. При значении 0.7 и выше каждую единицу модельного времени агент проходит проверку на впадение в панику:

Если Случайное_Число (0..1) < S, то:

Если Случайное_Число (0..1) > W – впасть в панику

Эффект паники проявляется в потере упорядоченности и логики действий у агента – вместо стандартного шаблона поведения (перемещаться по зданию согласно осведомленности или искать выход на основе навигационных объектов или в процессе «блуждания» по помещениям в одиночку или группой, избегая фронта горения и дымовой завесы) действия выбираются случайно, и среди них появляются заведомо опасные для агента (бегство в произвольном направлении, «застывание» на месте, эффекты толпы при множестве агентов) [7]. На рисунках 3-4 приведены снимки экрана, иллюстрирующие результаты паники:

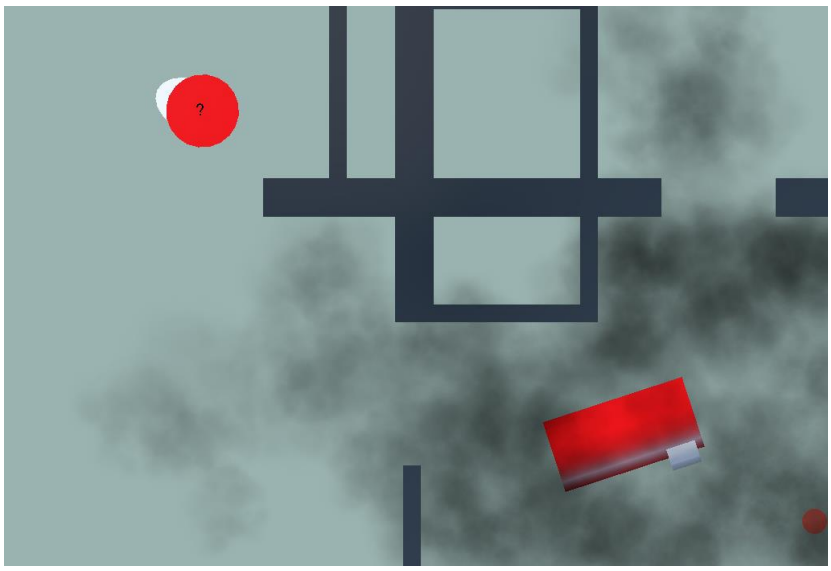


Рис.3. Гибель в задымленном помещении и потеря ориентации у паникующих агентов

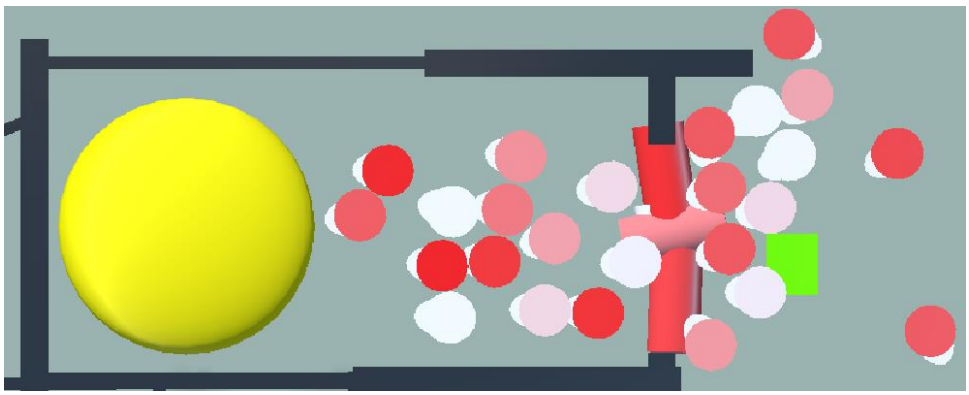


Рис.4. Давка и рассредоточение агентов в толпе при неорганизованном движении к выходу

Возвращаясь к W - чем его значение выше, тем меньше ежесекундная прибавка к S, имитирующая растущее напряжение агентов. Зависимость схожа с проверкой на панику и выглядит следующим образом:

Прибавка_S = 0.02/сек, если Случайное_Число (0..1) > W

W, таким образом, оценивает обратную вероятность ухудшения эмоционального состояния агента. При высоких его значениях (от 0.75 и выше) данный параметр также приобретает «успокаивающий» эффект – эмоциональное состояние агента при значении в 0,68 и выше (предел для «спокойного» состояния) ежесекундно уменьшается на 0,01 до достижения значения в 0,5.

Другим немаловажным (если и вовсе не ключевым) фактором, влияющим на качество эвакуации отдельного агента, является его способность к визуальному восприятию – от того, видит ли эвакуирующийся путь перед ним, выходы, других людей или его видимость ограничена вследствие задымленности или паники, зависит его способность к перемещению по зданию. В данной модели видимость реализована как Поле зрения *(V)* – область трапециевидной формы, реализуемая с использованием объекта типа «камера», в пределах которой агент способен визуально воспринимать окружающее его пространство (рис. 5).

На угол и длину поля зрения влияет как эмоциональное состояние агента (близость к паническому состоянию приводит к «туннельному эффекту» и уменьшению дальности обзора), так и степень задымленности помещения.

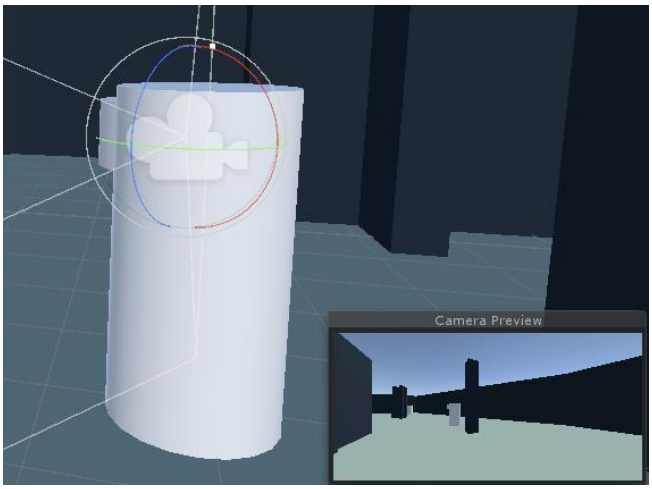


Рис.5. Поле зрения агента

Видимые агентом объекты получаются путем построения ограничивающей трапеции высотой в длину поля зрения и взятия всех объектов, вершины трехмерных форм которых

находятся в пределах данной трапеции, а подсчет числа частиц дыма, попадающих в нее, позволяет определить степень ухудшения обзора. Падение видимости – основной, вместе с недостатком осведомленности, эффект, на опасность которого для эвакуирующихся делается акцент в данной модели.

Последним параметром агента являются его Лидерские качества, L, определяющие способность объединять и вести за собой группу эвакуирующихся. Объединение агентов в контингенты симулирует эвакуацию, например, работников из цеха или учащихся из класса [7]. Групповое движение является основой организованной эвакуации – наличие лидера, в особенности – хорошо знающего планировку помещений (высокое значение F) и не подверженного панике (высокое значение W) – способствует успешному спасению из здания агентов, потенциально не способных на самостоятельную эвакуацию.

После определения групп при настройке симуляции и ее запуска в каждой группе выбирается агент с наибольшим L и назначается ее ведущим. Остальные агенты той же группы становятся ведомыми. В группе только лидер действует согласно вышеописанным шаблонам, когда как целью ведомых становится сохранение близости дистанции до лидера.

В случае потери лидера из вида на 5 секунд и более отдельными членами группы они откалываются от данной группы и продолжают эвакуацию самостоятельно. Откол от группы увеличивает S и при низком значении W (< 0.33) результирует в проверке на панику (см. выше).

Преимущества, даваемые группой – успокаивающее воздействие ведущего (эффект от W лидера распространяется на группу) и возможность обмена знаниями о структуре здания (F лидера приравнивается к наибольшей в группе). Однако, если лидер сам подвержен напряжению, и среди членов группы нет хорошо знающих здание агентов, то это может привести к массовой панике (рост S лидера способствует росту S у ведомых), неэффективному «блужданию» по зданию и затруднению эвакуации для всей группы, а не просто отдельных агентов.

Заключение. Для воссоздания поведения эвакуирующихся в комбинированной модели развития событий при пожаре был отобран набор параметров, на основе которого были созданы и протестированы в программном продукте – оболочке агенты. Путем отладки и добавления зависимостей параметров от различных условий в процессе последующего усовершенствования модели предполагается сделать их поведение максимально близко имитирующим реальные сценарии эвакуации.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. MST – Инженерное программное обеспечение: комплексное программное решение для оценки рисков пожарной безопасности Fenix+ [Электронный ресурс] // URL: https://mst.su/fenix/ (дата обращения: 21.05.2019)

2. Официальный сайт команды-разработчика программы для расчета пожарных рисков Urban [Электронный ресурс] // URL: http://www.urbanpo.ru/ (дата обращения: 15.05.2019)

3. Сопроводительный сайт к программному комплексу для расчета пожарного риска FireCat – описания и документации к отдельным программам в комплексе [Электронный ресурс] // URL: https://pyrosim.ru/programmy-dlya-rascheta-pozharnogo-riska (дата обращения: 15.05.2019)

4. Сайт команды-разработчика ZET Development Team [Электронный ресурс] // URL: http://zet-evakuierung.de/ (дата обращения: 15.05.2019)

5. Теребнев В.В. Особенности математического моделирования опасных факторов пожара // Интернет-журнал «НАУКОВЕДЕНИЕ» Том 9, №3 (2017)

6. Холщевников В.В. Людские потоки в зданиях, сооружениях и на территории их комплексов //Диссертация на соискание ученой степени доктора технических наук. МИСИ, 1983.

7. Кирик Е. С., Дектерев А. А., Литвинцев К. Ю., Харламов Е. Б., Малышев А. В. Математическое моделирование эвакуации при пожаре // Матем. моделирование, 2014, том 26, номер 1, 3–16

УДК 004

РАЗРАБОТКА ВИРТУАЛЬНОГО МУЗЕЯ ВЫЧИСЛИТЕЛЬНОЙ ТЕХНИКИ ИРНИТУ НА ОСНОВЕ ТЕХНОЛОГИИ ДОПОЛНЕННОЙ РЕАЛЬНОСТИ

Худякова Ю.Н., Куликова Л.Л.
(г. Иркутск, Иркутский национальный исследовательский технический университет)
e-mail: Yulya_2010.2010@mail.ru

Аннотация. Статья посвящена вопросам создания виртуального музея вычислительной техники ИРНИТУ при помощи браузеров дополненной реальности. Основная задача исследования – создание контента, который будет хранить информацию о виртуальном музее и экспонатах музея. Статья раскрывает основные цели создания музея, раскрывает структуру дополненной реальности, ее виды, описывает браузеры и их работу. Построена структурная схема взаимодействия пользователя и виртуального контента. Разработаны скрипты и база данных, созданы платформы взаимодействия между пользователем и контентом. Предложенный виртуальный музей будет интересным и, в тоже время, познавательным, у студентов появиться возможность не только узнать историю экспонатов музея, но и познакомиться поближе с возможностями технологий дополненной реальности.

Ключевые слова. Интерактивные мультимедийные технологии, экспонаты музея, дополненная реальность, JuniperResearch, Digi-Capital, музейная работа, вычислительная техника, браузеры дополненной реальности, геолокационная технология, GPS-координаты, аурная технология, триггер, оверлей, LayAr, слои, база данных, PHP скрипт, точки интереса, Blippar, распознавание изображение, считывание информации.

DEVELOPMENT OF A COMPUTER TECHNOLOGY VIRTUAL MUSEUM INRTU BASED ON AUGMENTED REALITY TECHNOLOGY

Hudyakova U.N., Kulikova L.L
(Irkutsk, Irkutsk National Research Technical University)
e-mail: Yulya_2010.2010@mail.ru

Abstract. The article is devoted to the creation of a virtual museum of computer technology at INRTU using augmented reality browsers. The main objective of the study is to create content that will store information about the virtual museum and museum exhibits. The article reveals the main goals of creating a museum, reveals the structure of augmented reality, its types, describes

browsers and their work. A structure diagram of user interaction and virtual content has been built. Scripts and a database were developed, and platforms for interaction between the user and the content were created. The proposed virtual museum will be interesting and, at the same time, informative. Students will have the opportunity to not only learn the history of museum exhibits, but also to get to know better the capabilities of augmented reality technologies.

Keywords. Interactive multimedia technology, augmented reality, JuniperResearch, Digi-Capital, augmented reality browsers.

Введение. Стремительное развитие интерактивных мультимедийных технологий нуждается в появлении новых интерфейсов взаимодействия. Эти интерфейсы отличаются от привычного графического меню; форма и панель инструментов опираются на методы взаимодействия, учитывающие особенности человеческого общения, такие, как жесты и речь. Сегодня направление дополненной реальности, является одним из самых перспективных в сфере IT-разработок. Мобильные приложения дополненной реальности могут помочь дальтоникам видеть цвета, туристам – изучить новые места, водителям – найти припаркованный автомобиль, они также позволяют примерять одежду при покупке или проверить, как будет выглядеть выбранный телевизор в интерьере покупателя.

Исследование виртуальной и дополненной реальности осуществляется несколько десятилетий, но из-за крайней дороговизны оборудования долгое время данные технологии не могли выйти к массовому потребителю. С широким распространением мобильных устройств со встроенными датчиками изменили ситуацию коренным образом. Исследовательская компания «JuniperResearch» в своем докладе, который был посвящен дополненной реальности, прогнозировала, что к 2017 году, ежегодно, на мобильные устройства будут скачиваться приблизительно 2,5 миллиарда приложений с дополненной реальностью. Компания Digi-Capital прогнозирует, что к 2020 году рынок дополненной реальности составит 120 млрд. долларов.

В настоящее время дополненная реальность используется в сфере образования, в маркетинге и рекламном бизнесе, в медицине, архитектуре и строительстве, в искусстве и многих других сферах [1].

Целью данной работы является изучение геолокационной технологии и технологий дополненной реальности и их использование в экскурсионной и музейной работе. Практическое значение работы заключается в прокладке маршрутов до нужных музейных объектов, а также создание музейного контента для существующего в ИРНИТУ музея вычислительной техники.

Дополненная реальность (Augmentedreality, AR) – это технология наложения информации в форме текста, графики, аудио- и других виртуальных объектов на реальные объекты в режиме реального времени. Взаимодействие вычислительных устройств с картинкой реального мира отличает дополненную реальность от виртуальной. С помощью данной технологии реальные объекты приобретают новые свойства и открываются людям с новой стороны. Другими словами, задача дополненной реальности – расширить взаимодействие пользователя с окружением, а не отделить его от реальности, и поместить в виртуальную среду. Слои, накладываемые компьютерным устройством, с контентными объектами на изображение реальной среды носят вспомогательно-информативный характер.

Таким образом, информация, с помощью дополненной реальности, становится доступна пользователю в режиме реального времени [2].

Аурная технология распознавания образов была создана компанией Autonomy, эта технология позволяет «вносить» в окружающий нас мир видео- и другие медиа-контенты в реальном времени. Объекты или изображения, на которые накладывается дополнительный контент, называются триггерами, это могут быть фотоснимок, страничка книги или журнала, постер, а дополнительная виртуальная информация – оверлеями. При направлении камеры мобильного устройства на изображение, оно распознается и поверх него возникает аура. Аура – это совмещение триггера и оверлея, отсюда и название данного вида.

Геолокационная технология использует GPS, компас и гироскоп, которые, на сегодняшний день, входят в современные гаджеты. Геолокация действует по принципу определения местоположения устройства и помогает быстро найти людей и объекты, которые находятся вблизи от пользователя. Виртуальный контент накладывается на реальные объекты, используя их GPS-координаты. Приложение связывается со спутниками, либо сервисами, например, Google-карты, система распознает свое местонахождение, подключается к базе данных и выдает на экране ответ на запрос, который вводил пользователь.

На сегодняшний день большинство разработчиков, как правило, не создают с нуля приложения дополненной реальности, а используют уже существующие браузеры. Компании, которые создают AR-браузеры, позволяют другим создавать свои собственные проекты на базе своей платформы и занимаются распространением их контента. Такие браузеры становятся удивительными сборниками самых разных примеров с использование дополненной реальности, которые являются интересными и полезными

В ходе изучения и анализа браузеров дополненной реальности были выбраны браузер LayAr и Blippar.

LayAr представляет собой браузер дополненной реальности, разработанный в Голландии в 2009 под платформы Android, iOS и Symbian [3].

Архитектура всей системы построена таким образом, что контент для браузера извлекается из внешних источников и предоставляется сторонними разработчиками. Различный контент от различных разработчиков предоставлен в виде слоев. Вся информация привязана к так называемым точкам интереса (POI), которые изображены на экране в виде кругов разного размера и связаны с конкретным географическим положением. Массив данных точек определяется географическими координатами, которые система получает при помощи GPS устройства и компаса. Размер круга зависит от расстояния пользователя и точки интереса, чем ближе находится пользователь к точке, тем больше круг в диаметре. Саму точку интереса и действия, связанные с этой точкой, настраивает сам разработчик слоя. Отсюда следует, что слой можно определить, как совокупность всех точек интереса этого слоя и некоторой статической метаинформации. Статическая метаинформация хранится на серверах LayAr, а динамическая информация о массиве точек интереса извлекается от разработчика в реальном времени [3].

Blippar – английская компания, основанная в 2011 году, которая использует технологию распознавания изображения. Сейчас технология Blippar имеет три функции: считывание информации с маркера, «живой поисковик», функция распознавания лица. У

компании Blippar есть сервис Blippbuilder, который позволяет быстро преобразовывать любые материалы в интерактивные объекты с дополненной реальностью [4].

В ходе работы над виртуальным музеем при помощи LayAr были разработаны геоокационные точки для ИРНИТУ. Для каждого объекта была созданы база данных, в которой хранится название объекта, принадлежащее ИРНИТУ, его координаты, адрес, кроме того, есть возможность перехода на сайт (если он есть) и возможность проложить маршрут от места нахождения устройства до нужного объекта (рис. 1).

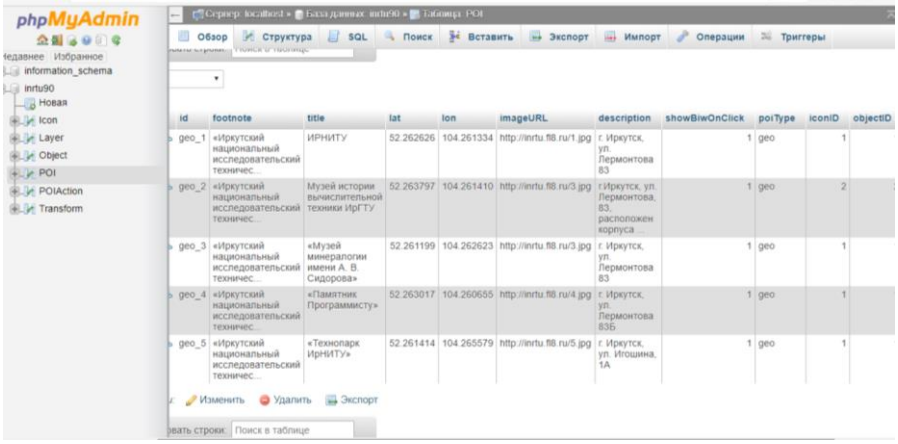


Рис. 1. Данные одной из таблиц БД

PHP скрипт состоит из двух файлов, один из которых содержит подключение к самой базе данных, и другой, который служит основным скриптом во взаимодействии слоя с базой данных, и также хранит подключение к первому файлу. Функции, описанные в основном скрипте, позволяют:

- осуществлять распределение полученных данных;
- прописывать запрос к таблице, для определения действия объекта, т.е. переход на сайт;
- прописывать запрос к таблице, для определения иконки объекта и ее отображения;
- создавать запрос для получения информации о самих объектах и также хранить функцию, которая присваивает полученные данные в переменную и отправляют ее пользователю на телефон (рис. 2).

```
function connectDb() {
  try {
    $dbconn = 'mysql:host=' . DBHOST . ';dbname=' . DBDATA ;
    $db = new PDO($dbconn , DBUSER , DBPASS , array(PDO::MYSQL_ATTR_INIT_COMMAND => 'SET NAMES utf8'));
    // set the error mode to exceptions
    $db->setAttribute(PDO::ATTR_ERRMODE , PDO::ERRMODE_EXCEPTION);
     return $db;
  }
  catch(PDOException $e) {
    error_log('message:' . $e->getMessage());
  }
}
```

Рис. 2. Фрагмент скрипта

Проведено тестирование на сайте LayAr, которое показало, что все объекты, которые были занесены в базу данных, успешно отражаются на карте, а также проведено тестирование на мобильном устройстве, где каждый объект отображался определенной иконкой, и все кнопки функционировали (рис. 3).

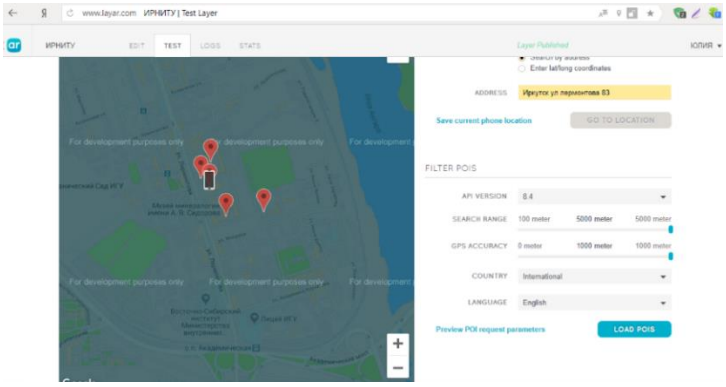


Рис. 3. Тестирование геолокационных точек

Объекты ИРНИТУ, внесенные в базу геолокации, представлены на рисунке 4:

– ИРНИТУ;
– «Памятник программисту»;
– Музей истории вычислительной техники ИРНИТУ;
– Музей минералогии имени А.В. Сидорова.

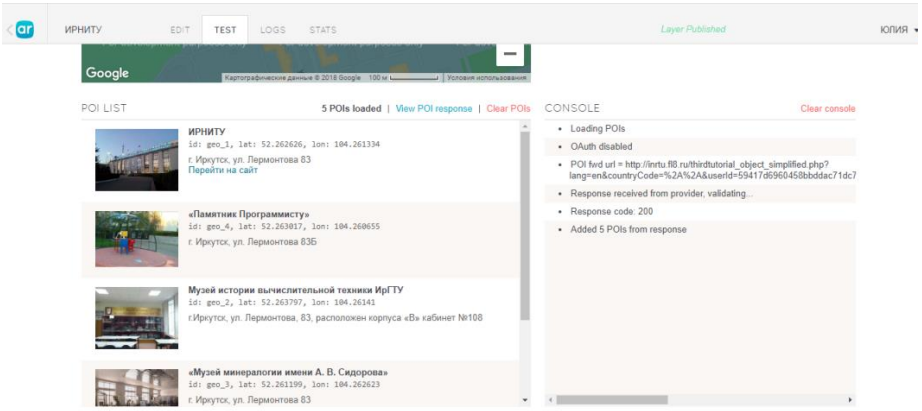


Рис. 4. Объекты ИРНИТУ

В рамках использования дополненной реальности для музея вычислительной техники при помощи Blippar были созданы проект «Техника», где будут храниться фото, видео, информация о технических экспонатах музея, и проект «Выпуски», в котором будут представлены выпуски студентов университета, их фотографии и различные сведения о них (рис. 5).

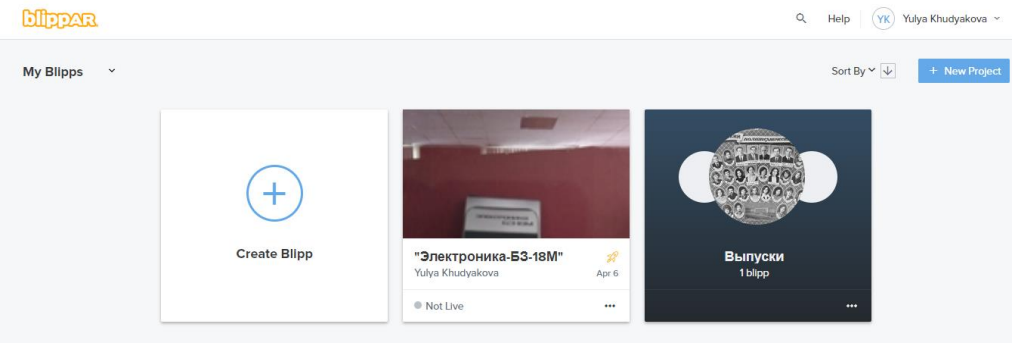


Рис. 5. Проекты

В проекте «Техника» будут храниться объекты музея вычислительной техники. При открытии слоя с определенным объектом через мобильное приложение будет запускаться информирующее видео, где рассказывается о модели объекта, его функциональных возможностях и предыстория, кроме того, есть возможность перехода на другую сцену, через кнопки, где можно прочитать более детальную информацию о музейном экспонате и просмотреть фото (рис.6).

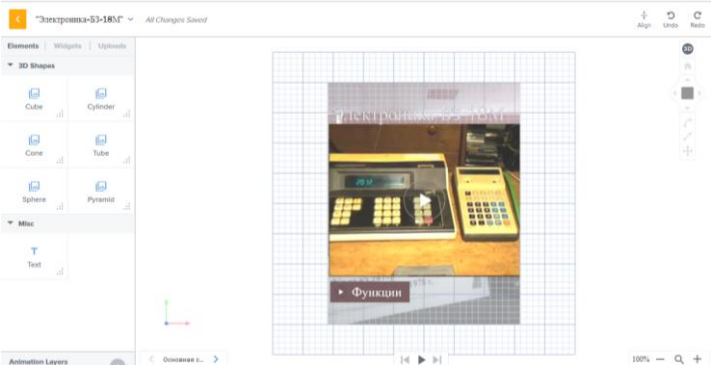


Рис. 6. Объект проекта "Техника"

Проект «Выпуски» будет являться мини-альбомом, где будут представлены выпуски университета, а каждый выпуск будет хранить список студентов и их фотографии (рис. 7).

Рис. 7. Проект "Выпуски"

Выводы. Проведенное исследование возможностей дополненной реальности позволило наметить пути развития музейной работы, связанное с освещением истории вычислительной техники в стенах ИРНИТУ. Эта история началась с 60-х годов прошлого столетия и продолжается по настоящее время. Имеющиеся в музее вычислительной техники экспонаты в наши дни являются раритетными, и информация, связанная с ними, должна быть донесена до сегодняшних студентов современными и интересными для них способами. Наполнение контентом может стать увлекательным занятием для первокурсников при изучении таких дисциплин, как Информатика, Введение в специальность и др. Посещение музея будет интересным и познавательным для абитуриентов: такая форма знакомства даст им возможность ознакомиться как с историей вычислительной техники, так и с ее новыми возможностями.

Дальнейшее развитие представленной работы связано с включением в разработку новых проектов, таких как « Предприятия», «Олимпиады», «Внедренные проекты» и др.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. Развитие дополненной реальности. [Электронный ресурс] //URL: https://habr.com/ru/post/378865/ (дата обращения: 20.05.2019)
2. Яковлев Б.С., Пустов С. И., «История, Особенности и перспективы технологии дополненной реальности». [Электронный ресурс] //URL: https://cyberleninka.ru/article/v/istoriya-osobennosti-i-perspektivy-tehnologii-dopolnennoy-realnosti (дата обращения: 20.05.2019)
3. Баулин И.Н., Слои для Браузера Дополненной Реальности //Издательство: Лаборатория Открытых Информационных Технологий факультета ВМК МГУ им. М.В. – Москва, 2013.-16-23с.
4. Дополненная реальность-новый способ достучаться до потребителя. [Электронный ресурс] //URL: http://marketing.by/mnenie/dopolnennaya-realnost-novyy-sposob-dostuchatsya-do-potrebitelya-kak-eto-rabotaet/ (дата обращения: 20.05.2019)
5. LesterMadden, Professional Augmented Reality Browsers for Smartphones. // ProgrammingforJunaio, LayarandWikitude. 2011.

УДК 004.92; 004.94; 004.42

МОДЕЛИРОВАНИЕ ОТРАЖАТЕЛЬНЫХ ХАРАКТЕРИСТИК КОСМИЧЕСКОГО АППАРАТА EGP В ОПТИЧЕСКОМ ДИАПАЗОНЕ

Шишкин П.И.
(г. Иркутск, Иркутский национальный исследовательский технический университет)
e-mail: syoshi.1995@gmail.com

Аннотация. Статья посвящена вопросам создания программного инструментария, позволяющего интерпретировать полученные данные для исследования поведения космических объектов на орбите. Для создания программного инструментария, был выполнен сравнительный анализ по выбору среды разработки, подходящий для решения основных задач моделирования. Проведена калибровка, которая позволяет достаточно точно определить кривую блеска исследуемых объектов. Представляемая работа нацелена на получение отражательных характеристик космического аппарата в оптическом диапазоне, разбирается на примере разработанной трёхмерной модели EGP.

Ключевые слова. Космический аппарат, космический объект, моделирование, EGP (AJISAI), отражательные характеристики, кривая блеска, динамика блеска, яркость, фотометрия, оптический сигнал.

SIMULATION OF THE REFLECTIVE CHARACTERISTICS OF EGP SATELLITE IN THE OPTICAL RANGE

Shishkin P.I.
(Irkutsk, Irkutsk National Research Technical University)
e-mail: syoshi.1995@gmail.com

Abstract. The article is devoted to the creation of software tools to interpret the data obtained to study the behavior of space objects on the orbit. To create software tools, a comparative analysis was performed on the choice of a development environment suitable for solving the main modeling problems. Calibration which allows to accurately determine the light curve of the studied

objects has been carried out. Presented work is aimed at obtaining the reflective characteristics of a satellite in the optical range, it is analyzed using the developed three-dimensional model of EGP as an example.

Keywords. Satellite, space object, modeling, EGP (AJISAI), reflection characteristics, light curve, light dynamics, brightness, photometry, optical signal.

Введение. На радиооптическом комплексе Института Солнечно-Земной Физики СО РАН с 2003 года накоплен большой объём данных наблюдений за низкоорбитальными космическими объектами. В состав комплекса входят: астрономический комплекс Саянской обсерватории и высокопотенциальный радар некогерентного рассеивания. Важной задачей является разработка программного инструментария, позволяющего интерпретировать полученные данные для исследования поведения космических объектов на орбите, очень актуальными характеристиками являются скорость и ориентации оси вращения космических объектов. Представляемая работа нацелена на решение данной задачи.

Наиболее интересные результаты были получены при анализе амплитуд радиооптических сигналов, отраженных от космического аппарата (КА) EGP AJISAI – японский спутник, был запущен в 1986 году. После запуска спутнику было дано новое название – AJISAI (Гортензия), но этот спутник наиболее известен под аббревиатурой EGP. Некоторые источники называют аппарат экспериментальным геодезическим спутником, или EG. Представляет собой сферу диаметром 215 см., покрытую алюминиевыми пластинками 318 шт. и уголковыми отражателями 1436 шт. в 120 наборах (рис. 1).

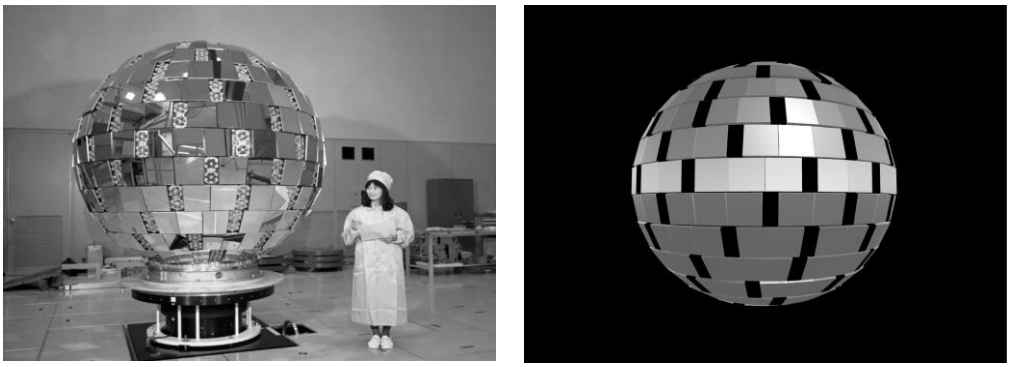


Рис. 1. Спутник AJISAI (EGP) и его 3D-модель в Unity

Трёхмерная модель спутника построена на основе доступной литературы и цифровых изображений, размещённых в Internet. Расположение панелей 120 угловых отражателей взяты из научной статьи «A method to calculate zero-signature satellite laser ranging normal points for millimeter geodesy – a case study with Ajisai» [1] и представлено на рисунке 2. Каждая панель содержит 12 угловых отражателей, за исключением одной, установленной на центральном кольце только с 8. Угловые отражатели сделаны из плавленого кварца для зелёного света 532 нм, а задние поверхности отражателей не имеют покрытия. Специальная конструкция сферического тела (ламинированные листы алюминиевой фольги) сводит к минимуму взаимодействие с магнитным полем Земли и предотвращает потерю энергии вращения.

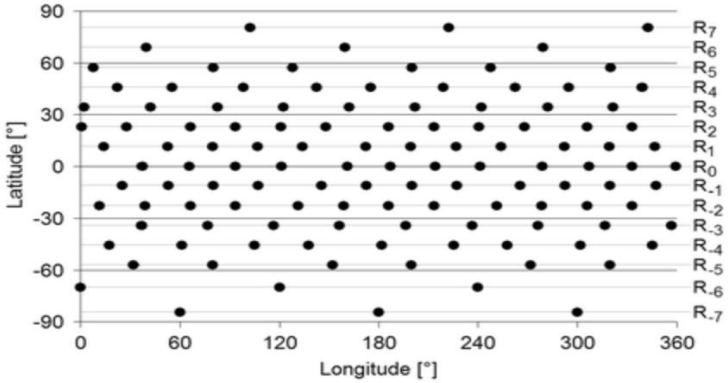


Рис. 2. Расположение панелей угловых отражателей на поверхности. Матрица состоит из 15 колец R(-7)… R(7); R(0) – экваторное кольцо.

Для моделирования оптических сигналов отражённых исследуемыми объектом использован инструмент для разработки двух и трёхмерных приложений UNITY3D, позволяющий достаточно гибко визуализировать систему Солнце-КА-Телескоп, а также позволяющий использовать встроенные возможности моделирования физических свойств исследуемых объектов.

Unity3D позволяет получить отражательные характеристик КА AJISAI, посредством использования метрики евклидова расстояния в трёхмерном пространстве RGB по формуле 1. Данный метод учитывает воспринимаемую степень интенсивности трёх основных цветов, давая каждому коэффициент, соответствующий тому, как свет или тьму человеческий глаз воспринимает. Степень интенсивности света обычно упоминается как яркость (в англ. brightness).

$$\text{brightness} = \sqrt{r^2} * 0.299 + g^2 * 0.587 + b^2 * 0.114. \quad (1)$$

Относительная яркость изменяется от 0 до 1, где 0 – тёмный и 1 – светлый. Путём сложения яркости пикселей видимого объекта наблюдателем, можно найти среднее значение отражательных характеристик КА, что позволяет следить за динамикой изменения блеска (отражающих свойств) КА. В качестве наблюдателя в Unity выступает камера, она является устройством, которое захватывает и отображает моделируемую среду.

Относительная яркость следует фотометрическому определению яркости – понимается под набором хорошо изученных спектральных полос (фильтров) с известной чувствительностью к падающему излучению. Чувствительность обычно зависит от используемых оптических систем, детекторов и фильтров. Для каждой фотометрической системы определён набор первичных фотометрических стандартов – звезды с (точно) известной звёздной величиной в каждой полосе [7]. Значения нормализованы к 1 для эталонного белого цвета. Подобно определением фотометрического, это связанно со светящейся плотностью потока в определённом направлении, что плотность потока излучения взвешивается по функции светимости.

В измерениях отражательных характеристик света, яркость неразрешённого объекта измеряется с течением времени, наблюдаемое в определённом направлении просмотра. Свет, полученный наблюдателем, зависит от геометрии объекта, материалов из которых состоит объект, ориентации всех освещённых граней, освещённости и направления наблюдения. Пример возможной геометрии просмотра показан на рисунке 3. Определена локальная

система координат, вдоль её главной оси. Классический фазовый угол, Земля-объект-Солнце, разбивается на два угла, в плоскости xy и в плоскости xz.

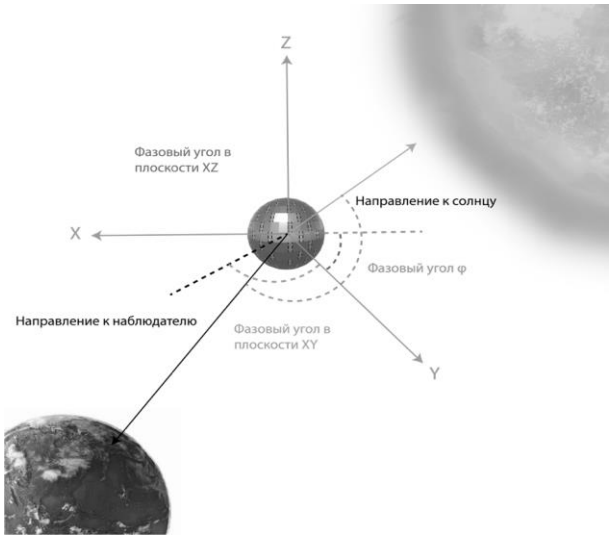


Рис. 3. Система координат возможной геометрии просмотра

Для анализа кривых блеска были смоделированы различные объекты (пластина, куб, цилиндр, КА EGP). Источник света был откалиброван для представления реалистичного освещения от Солнца. Освещённые и видимые грани были интегрированы с дискретными временными шагами. Объекты имеют металлическую поверхность, калибровка проводились при различной степени гладкости и интенсивности падающего света.

Введена система координат для моделируемой металлической пластины. Ось Y совпадает с нормалью, угол между нормалью и вектором наблюдателя составляет 45 градусов. Объект является не подвижным, меняется высота солнца от 0 до 180 градусов, при азимуте равным 270 к азимуту 90 (рис. 4).

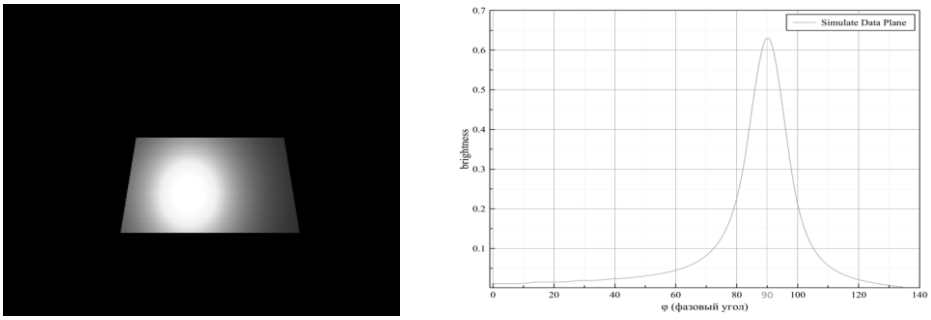


Рис. 4. Моделирование кривой блеска для металлической пластины

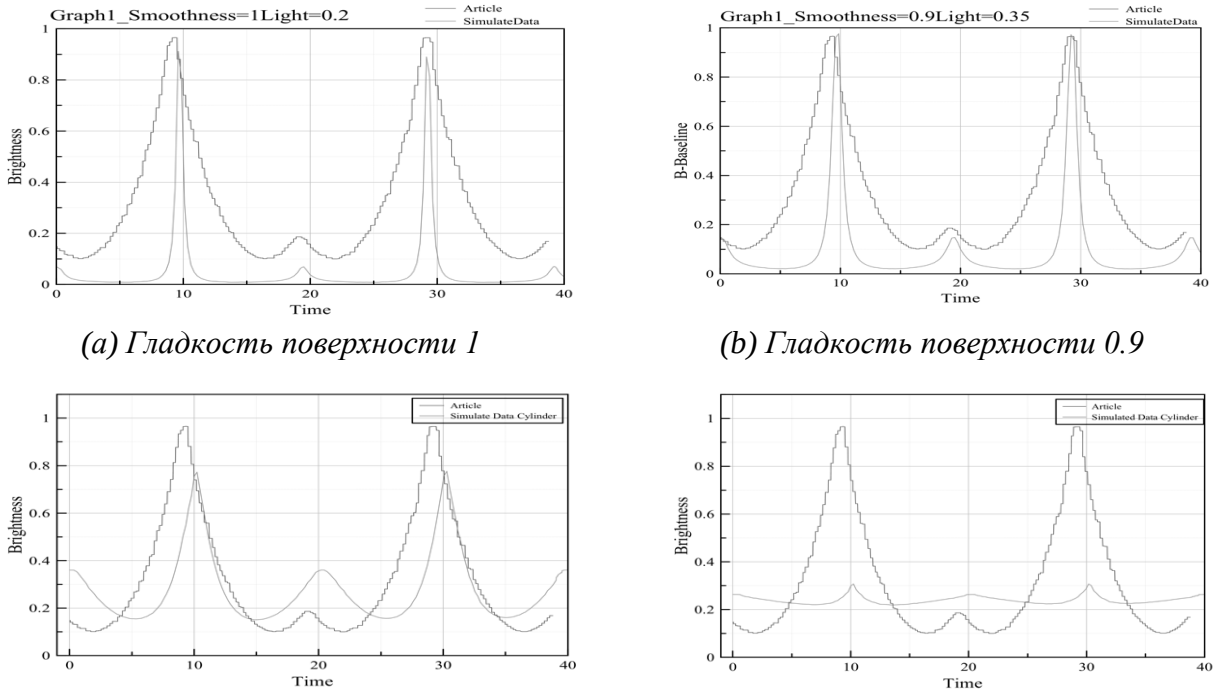
При достижении высоты солнца 90 градусов, нормаль и вектор солнца по отношению к объекту становится равным 45 градусам. В этот момент на графике моделирования кривой блеска заметен самый высокий пик – это говорит о том, что при смоделированном методе получения отражательных характеристик, получаемые данные являются достоверными.

Проведён сравнительный анализ полученной динамики блеска, данные сравнивались с научной работой «Анализ наблюдений моделируемых лёгких кривых в пространстве» [2]. В данной статье автор моделировал различные типы объектов таких как цилиндр (рис. 5.b), полученные данные динамики блеска сравнивались с динамикой блеска смоделированного цилиндра в Unity (рис. 5.a).

(a) Моделируемая фигура *(b) Фигура из статьи*

Рис. 5. Исследуемые фигуры

Моделируется однократное вращение вокруг оси z в системе координат с периодом 40 секунд под фазовым углом 0 градусов, при интенсивности света равным от 0 до 8, гладкость поверхности изменяется в пределе от 0 до 1 (рис. 6). Используемая временная шкала произвольная – предполагалось, что каждый сделанный шаг времени соответствует выборке в одну секунду.



(a) Гладкость поверхности 1 *(b) Гладкость поверхности 0.9*

(c) Гладкость поверхности 0.5 *(d) Гладкость поверхности 0.1*

Рис. 6. Графики кривых блеска при различной степени шероховатости поверхности

Сравнительный анализ данных показывает, что методика получения отражательных характеристик позволяет достаточно точно определить кривую блеска, т.к. амплитуда моделируемых кривых блеска и кривых блеска из статьи очень схожа по времени и высоте пиков. В дальнейшем будет выполнен сравнительный анализ моделируемых данных КА EGP с реальными данными спутника.

Выводы. В результате проделанной работы, с помощью игрового движка Unity3D удалось разработать методику определения кривой блеска, при заданном изменении фазового угла. Предложенный метод имитационного моделирования позволяет оценить отражательные характеристики различных космических объектов, изменяя интенсивность и вращение объекта, можно наблюдать за изменением кривой блеска в реальном времени.

Решение прямой задачи моделирования отражательных характеристик, позволит перейти к обратной задаче – определению скорости и ориентации оси вращения по зарегистрированной динамике блеска.

Проведённая калибровка показала, что данная методика получения отражательных характеристик позволяет достаточно точно определить кривую блеска исследуемых объектов.

В настоящее время решены следующие задачи: заложена динамика движения Земли вокруг солнца (учитывается прецессия и нутация), на основе алгоритма SGP-4 рассчитывается движение EGP вокруг Земли, разработана модель вращения EGP (в собственной системе координат), создана математическая модель структуры поверхности EGP. Разработанный программный комплекс позволяет исследовать динамику отражательных характеристик EGP в оптическом диапазоне, в настоящее время проводиться сопоставление модельных и экспериментальных данных.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. Daniel K., Georg K., Toshimichi O., Franz K., A method to calculate zero-signature satellite laser ranging normal points for millimeter geodesy - a case study with AJISAI. // Режим доступа: Calculate_space.pdf – Дата доступа: 20.01.2017.

2. Thomas S., Analysis of observed simulated light curves of space debris // Режим доступа: Analysis_of_observed.pdf – Дата доступа: 27.04.2017

3. Xian-Zong Bai, Ke-Bo Li, Yan-Gang Liang, Lei Chen, Orbital Data Applications for Space Objects // Режим доступа: Orbital_Data.pdf – Дата доступа: 24.01.2019

4. Свет в компьютерной графике // Render.ru [Электронный-ресурс] URL: http://render.ru/books/show_book.php?book_id=154 (дата обращения: 27.01.2019)

5. Анатомия мешей // Документация Unity3D [Электронный ресурс] // URL: https://docs.unity3d.com/ru/530/Manual/AnatomyofaMesh.html (дата обращения: 08.02.2019).

6. Хабрахабр – новостной сайт [Электронный-ресурс] // URL: https://habrahabr.ru/post/120638/ (дата обращения: 01.03.2019)

7. Stackoverflow [Электронный-ресурс] //

URL: https://stackoverflow.com/questions/596216/formula-to-determine-brightness-of-rgb-color (дата обращения: 11.03.2019)

8. Википедия — свободная энциклопедия [Электронный-ресурс] // URL: https://ru.wikipedia.org/wiki/Фотометрическая_величина (дата обращения: 20.03.2019)

УДК 004.415.2

ПОИСК МАТРИЦЫ РАССТОЯНИЙ С ИСПОЛЬЗОВАНИЕМ ГИС СИСТЕМ

Калугин М. В.
(г. Иркутск, Иркутский национальный исследовательский технический университет)
e-mail: ultus2017@yandex.ru

Аннотация. Проведен обзор наиболее популярных картографических сервисов, позволяющих определить расстояние между объектами. Предложен способ поиска матрицы

расстояний с использованием инструментов Grapphopper, Nominatim на данных, предоставленных проектом OpenStreetMap. Данное решение применимо практически для всех населенных пунктов и ограничивается только покрытием карты OSM.

Ключевые слова. Матрица расстояний, картографические сервисы, поиск расстояния между объектами.

DETERMINING DISTANCE MATRIX USING GIS SYSTEMS

Kalugin M.V.
(Irkutsk, Irkutsk National Research Technical University, Irkutsk)
e-mail: ultus2017@yandex.ru

Abstract. A review of the most popular cartographic services for that allow the determination of the distance between objects was conducted. A method for determining distance matrix using such tools as Graphhopper and Nominatim with the data provided by the OpenStreetMap project is proposed. This method is applicable practically to any inhabited locality. The only restriction is OSM coverage.

Keywords. Distance matrix, cartographic services, distance determination between objects.

Введение. При решении прикладных задач, таких как оптимизация плана перевозки грузов между объектами, должна быть определена матрица расстояний. В настоящей работе предлагается использовать современные картографические системы. Тогда нахождение такой матрицы сводится к нахождению всех значений расстояний между адресами объектов маршрута. Так как количество адресов может быть достаточно велико, разумно автоматизировать процесс нахождения значений матрицы расстояний. В статье представлен подход для автоматизированного формирования матрицы расстояний для объектов, расположенных на карте по соответствующим им адресам.

Постановка задачи. Входными данными является документ – маршрутный лист, содержащий n адресов объектов a_i, где $i = 1, n$ (рисунок 1). Ставится задача: найти расстояния d_{ij} между a_i и a_j, где $i = 1, n;\ j = 1, n;\ i \neq j$ (рисунок 2).

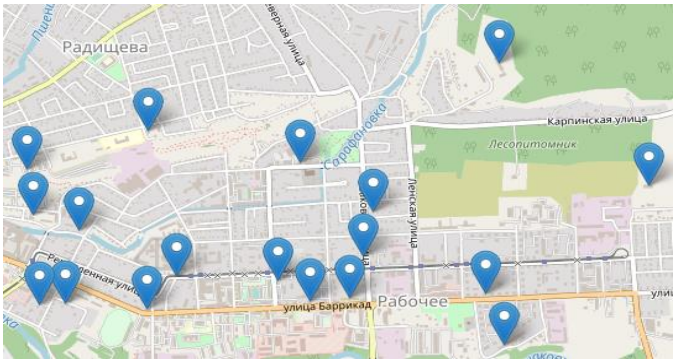


Рис. 1. Пункты из маршрутного листа, отмеченные на карте

Маршрут между двумя адресами обычно не пролегает напрямую, ввиду различных ограничений, таких как: дома, отсутствие дороги, одностороннее движение по дороге, реки, и прочие препятствия (рис. 2).

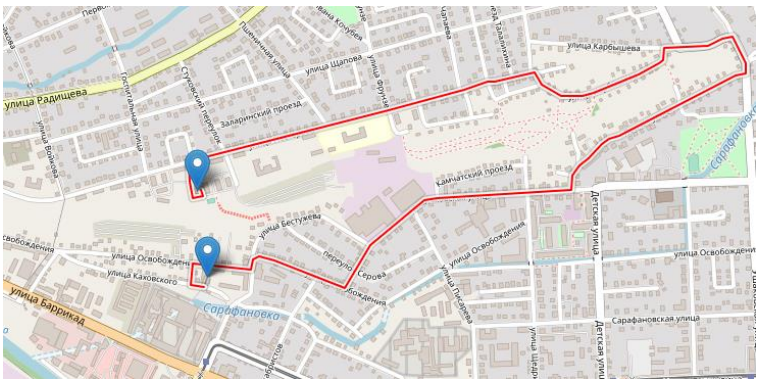


Рис. 2. Пример маршрута между двумя адресами

Формирование матрицы расстояний. ГИС-автоматизированная информационная система, предназначенная для обработки пространственно-временных данных, основой интеграции которых служит географическая информация [1]. Веб-ГИС, являющиеся разновидностью ГИС, предоставляют доступ к таким данным средствами программного интерфейса приложения, основанного на веб-технологиях (Web API).

Основные возможности веб-ГИС:

– отображение географической карты;
– поиск объектов на карте;
– построение маршрута между объектами.

Существует множество сервисов в сети Интернет, позволяющих определить расстояние между двумя адресами. Некоторые из наиболее популярных сервисов приведены в таблице 1.

Таблица 1. Обзор картографических сервисов

№	Название сервиса	Условия использования	Обязательна регистрация и получение ключа
1.	Яндекс карты	25000 запросов в сутки при использовании в некоммерческих целях без права сохранения данных. В любом другом случае предлагается оформление договора на использование API от 120 тыс. руб в год.	Да
2.	Google maps	2500 запросов в сутки при использовании в некоммерческих проектах. Для других целей каждые 1000 запросов обойдутся в 5$	Да
3.	2GIS	Бесплатно для использования в некоммерческих проектах, не направленных на построение маршрутов.	Да
4.	Bing maps	125,000 запросов в год при использовании в некоммерческих целях.	Да
5.	Here maps	От 1$ за каждые 1000 запросов при достижении 250 тыс. запросов за все время пользования.	Да
6.	Mapline	Бесплатный пробный период для индивидуального некоммерческого использования.	Да
7.	OpenStreet Maps(OSM)	Распространяется свободно для любых целей при указании ссылки на источник данных.	Нет

Для построения матрицы расстояний предлагается использовать сервис OpenStreetMaps, учитывая следующие факторы:

– бесплатное использование;

– подключение к сети Интернет необходимо только для скачивания и обновления данных карт;
– сохранение конфиденциальности данных, так как все технологии и инструменты запускаются локально и никакие запросы за пределы локальной сети не отправляются;
– свободное редактирование данных карт.

Карты OpenStreetMaps представляют собой набор данных в формате XML с описанием таких элементов, как Node, Way, Relation, Tag, обладающих набором параметров, которые характеризуют положение на карте, тип объекта, отображение на карте [2]. Например, данные, описывающие автобусную остановку, на карте выглядят следующим образом:

<node id="626311408" lat="52.2642324" lon="104.2220967" version="6" timestamp="2013-02-11T14:18:26Z" changeset="14994520" uid="1227135" user="muklem">
<tag k="bus" v="yes"/>
<tag k="name" v="Мельниковский рынок"/>
<tag k="public_transport" v="stop_position"/>
</node>.

Для обработки данных карты использованы следующие инструменты:

1. Nominatim[1] – инструмент для поиска данных OSM по имени и адресу, а также для создания полных адресов точек OSM (обратное геокодирование);

2. GraphHopper[2] – библиотека с открытым исходным кодом и сервер, написанные на языке программирования Java, для поиска оптимального проезда между двумя точками на карте OSM.

Матрицу расстояний предлагается определить по следующему алгоритму (рис. 3).

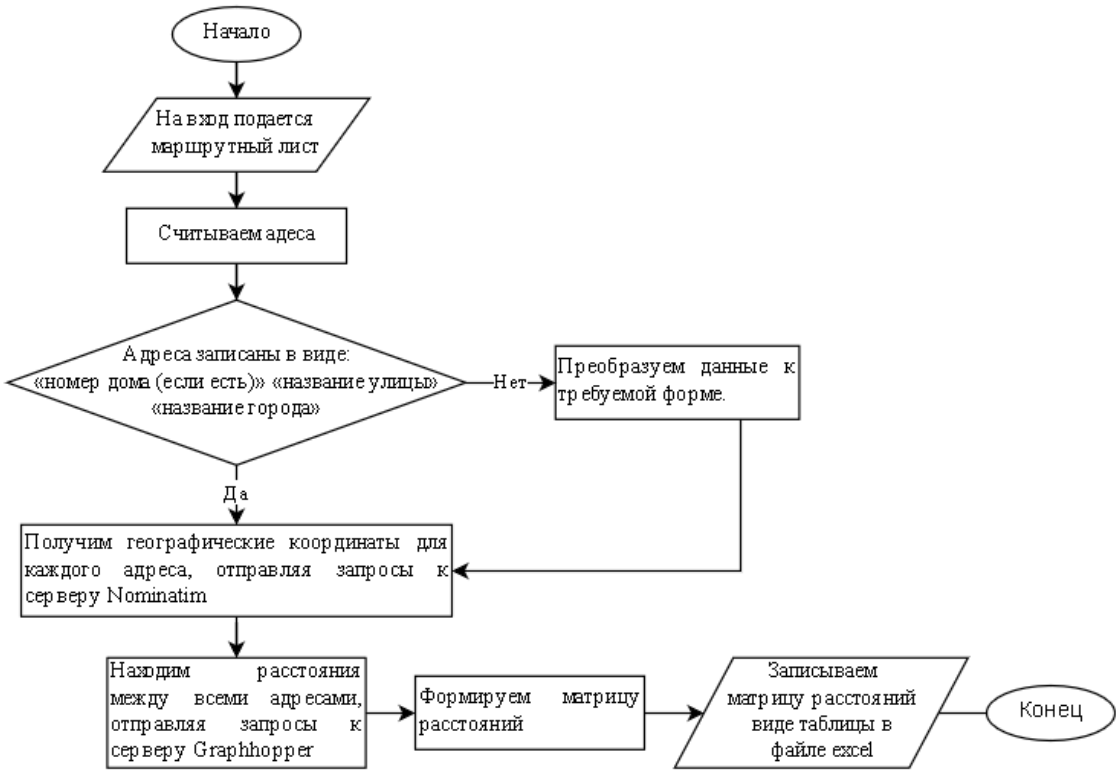


Рис. 3. Блок-схема алгоритма нахождения матрицы расстояний

[1] Nominatim : [сайт]. URL: https://www.nominatim.org
[2] GraphHopper : [сайт]. URL: https://www.graphhopper.com

Запрос к веб-сервису Nominatim выглядит так:

https://localhost:8990/search?q=103+Вилюйская+Иркутск&format=geojson

Получив адрес, сервер запустит процесс геокодирования, который заключается в поиске элемента Node среди данных карты OSM:

<way id="647079356" version="1" timestamp="2018-11-21T00:29:01Z" changeset="64718018" uid="4988915" user="Sasha_SH"><nd ref="6077798173"/> <tag k="addr:housenumber" v="103"/> <tag k="addr:street" v="Вилюйская улица"/> <tag k="building" v="detached"/></way>

<node id="6077798173" lat="52.2839755" lon="104.3761923" version="1" timestamp="2018-11-21T00:28:57Z" changeset="64718018" uid="4988915" user="Sasha_SH"/>

Ответ сервера будет содержать массив данных в формате JSON, содержащий координаты объекта и другие данные:

{Features:[{“properties”:{…},”bbox”:[…],geometry:{“type”:"Point",
”coordinates”:[104.376239, 52.28390875], {…}]}

Следующим шагом после получения координат объектов является построение маршрутов между парами адресов. Запрос к веб-сервису Graphhopper содержит две пары координат и выглядит так:

https://localhost:8989/route?point=52.284752331532694%2C104.25021171569824&point=52.284673572676155%2C104.25488948822023

Получив данный запрос, сервер Graphhopper выполняет алгоритм A* для переданных адресов. Алгоритм A* пошагово просматривает все пути, ведущие от начальной вершины в конечную, пока не найдёт минимальный [3]. Его важным отличием является то, что он просматривает сначала те маршруты, которые «кажутся» ведущими к цели. Ответ сервера так же содержит данные о маршруте между точками в формате JSON:

{“hints”:{“visited_nodes.sum”:”18”},”paths”:[{“distance”:496.506,"weight":32.498556, ”time”:32496, … }]}

Получив значения расстояний для каждой пары адресов, сформируем матрицу расстояний (таб. 2).

Таблица 2. Матрица расстояний для десяти адресов

	103 Вилюйская Иркутск	2 Вилюйская Иркутск	63 Вилюйская Иркутск	34 Нестерова Иркутск	22 Нестерова Иркутск	33 Каховского Иркутск	1 Лаптевых Иркутск	11 Мало-Якутская Иркутск	Скушникова Иркутск	31 Фабричная Иркутск
103 Вилюйская Иркутск	X	824.563	569.108	6108.569	7639.589	5822.393	8783.797	8311.099	9386.812	8498.556
2 Вилюйская Иркутск	824.563	X	343.186	5268.901	6799.921	4982.726	7944.129	7471.431	8547.144	7658.889
63 Вилюйская Иркутск	569.108	343.186	X	5612.131	7143.152	5325.956	8287.36	7814.662	8890.375	8002.119
34 Нестерова Иркутск	5857.053	5017.385	5360.615	X	692.045	3978.19	3874.589	3401.891	4417.535	3589.349
22 Нестерова Иркутск	6487.403	5647.735	5990.966	692.045	X	3485.771	3382.171	2909.473	4148.863	3096.93
33 Каховского Иркутск	5833.884	4994.217	5337.447	2962.45	2470.031	X	3614.239	3141.541	4217.254	3328.998
1 Лаптевых Иркутск	9093.148	8253.48	8596.711	3874.589	3382.171	3923.59	X	515.826	685.088	634.763
11 Мало-Якутская Иркутск	8620.45	7780.782	8124.013	3401.891	2909.473	3450.892	515.826	X	1181.586	187.457
Скушникова Иркутск	10020.053	9180.385	9523.616	4641.282	4148.863	4850.495	685.088	1181.586	X	1300.523
31 Фабричная Иркутск	8807.907	7968.24	8311.47	3589.349	3096.93	3638.349	634.763	187.457	1300.523	X

В результате получили несимметричную матрицу расстояний, так как маршрут в одну сторону может проходить через одни узлы, а в обратную сторону – через другие узлы, следовательно, расстояния будут разниться (например, из-за дорог с односторонним движением).

Вывод. В статье предложена методика поиска матрицы расстояний с использованием инструментов Grapphopper, Nominatim на данных предоставленных проектом OpenStreetMap.

Данное решение применимо практически для всех населенных пунктов и ограничивается только покрытием карты OSM.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. Соловьёв И.В., Цветков В.Я. Геоинформационные системы: краткий словарь-справочник // М.: МГТУ МИРЭА, 2013. - 82 с.
2. OpenStreetMap как источник геоданных. [Электронный ресурс] // URL: https://habr.com/ru/post/270513/ (дата обращения: 25.05.2019).
3. Нильсон Н. Принципы искусственного интеллекта // М.: Книга по Требованию, 2012. - 369 с.

УДК 004.75

АРХИТЕКТУРА ВЕБ-ПРИЛОЖЕНИЯ ДЛЯ ИССЛЕДОВАНИЯ ИНТЕГРИРОВАННЫХ СИСТЕМ ЭНЕРГЕТИКИ

Орехов Д.О.[1], Барахтенко Е. А.[2]
[1](г. Иркутск, Иркутский национальный исследовательский технический университет)
[2] (г. Иркутск, Институт систем энергетики им. Мелентьева Сибирского отделения Российской академии наук)
e-mail: Dimao37@mail.ru , evgeny.barakhtenko@mail.ru

Аннотация. В статье представлена архитектура современного веб-приложения для исследования интегрированных систем энергетики. Рассмотрены основные компоненты этого веб-приложения. Показано использование современных веб-технологий при реализации прототипа веб-приложения для исследования интегрированных систем энергетики.

Ключевые слова. Веб-приложение, клиент-серверная архитектура, распределенная архитектура, веб-технологии, интегрированные системы энергетики.

ARCHITECTURE OF WEB APPLICATION FOR RESEARCH OF INTEGRATED ENERGY SYSTEMS

Orehov D.O.[1], Barakhtenko E.A.[2]
[1](Irkutsk, Irkutsk National Research Technical University, Irkutsk Aviation Plant)
[2](Irkutsk, Melentiev Energy Systems Institute, Siberian Branch of the Russian Academy of Sciences)
e-mail: Dimao37@mail.ru , evgeny.barakhtenko@mail.ru

Abstract. The article presents the architecture of a modern web application for integrated energy systems research. The main components of the web application are considered. The use of modern web technologies in the implementation of a prototype web application for integrated energy systems research is demonstrated.

Keywords. Web application, client-server architecture, distributed architecture, web technologies, integrated energy systems.

Введение. В настоящее время глобальная компьютерная сеть Интернет имеет большое значение для людей. В этой сети ежедневно происходят миллионы операций, которые охватывают социальные сети, платформы для обучения, сайты, порталы компаний и многое другое. Глобальная сеть Интернет охватывает многие аспекты нашей жизни. Благодаря ей и технологиям, которые используются в сети Интернет, становится возможным размещение приложений удаленно и получение доступа к ним не только с рабочего места, но и любой другой точки Земли.

Персональные компьютеры имеют ограничения при работе с приложениями. Для выполнения работы в конкретной программе, необходимо установить ее на компьютер. Веб-приложение же устанавливается на сервер, и пользователь может работать с ним без установки, находясь при этом не на рабочем месте, а в любой точке Земли. Это позволяет мобильно работать с приложением, не тратя время на его установку.

В данной статье рассматривается вопрос разработки архитектуры веб-приложения для проведения исследований интегрированных энергетических систем (ИЭС). Под ИЭС понимаются системы, ориентированные на использование нескольких видов энергоносителей с комплексным применением информационных технологий и телекоммуникаций, в совокупности обеспечивающих возможность построения более эффективной системы энергопроизводства, энергоснабжения и энергопотребления [1]. Это приложение позволит создавать модели ИЭС и проводить расчеты при исследовании ИЭС.

Задача создания модели ИЭС представляет собой процесс создания моделей подсистем этой ИЭС, представляющих разнотипные энергетические системы (электро-, тепло-, хладо-, топливоснабжения), в пределах одной рабочей области и определение различного типа связей между ними. Модель ИЭС используется для проведения расчетов и анализа ИЭС, а также в интеллектуальных системах принятия решений для управления развитием и функционированием ИЭС. Целью работы является разработка архитектуры веб-приложения для исследования ИЭС.

Архитектура веб-приложения для исследования интегрированных энергетических систем. Веб-приложение построено на клиент-серверной архитектуре, где можно выделить две основные части приложения, это клиентская и серверная части. Архитектура веб-приложения представлена на рисунке 1.

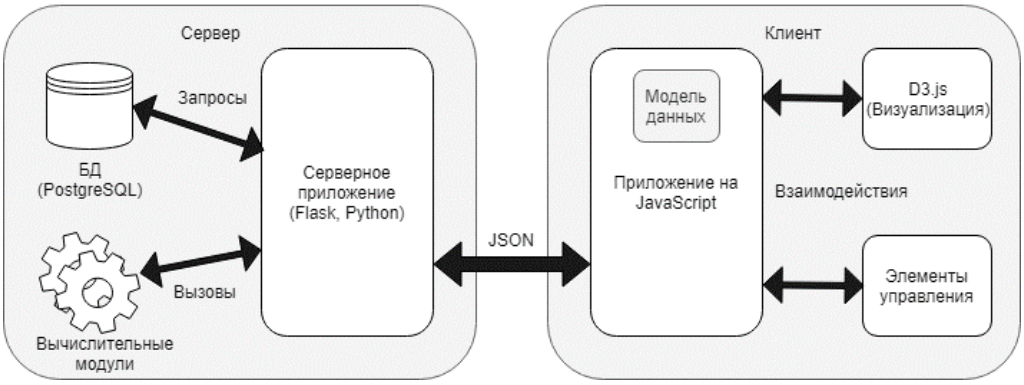


Рис. 1. Архитектура веб-приложения

Клиентская часть веб-приложения выполняется в браузере и состоит из трех основных компонентов. Первый из компонентов – это библиотека D3.js [2], отвечающая за визуальное отображение данных (модели ИЭС) и интерактивную работу с элементами систем.

Второй компонент – это пользовательский интерфейс, содержащий элементы управления в виде различных меню, кнопок, полей и т.д., и строку состояния, содержащую необходимую информацию при работе с приложением.

Третий компонент - библиотека на JavaScript [3], которая устанавливает взаимосвязь с D3.js и предоставляет возможность управлять данными и конфигурациями клиента. Этот компонент содержит в себе модель данных, которая загружается с серверной части и предоставляется первому компоненту клиентской части веб-приложения для выполнения визуализации.

Серверная часть приложения содержит также три компонента, которые могут размещаться на одном сервере или же иметь распределенную архитектуру. Серверное приложение является первым компонентом. Оно реализуется на базе технологии Flask [4], языка программирования Python [5]. Серверное приложение получает от клиента запросы и реагирует на них соответствующим образом.

Вторым компонентом является база данных (PostgreSQL [6]), которая хранит в себе информацию о модели ИЭС, а также служебную информацию (сведения о пользователях, сетях, элементах и их атрибутах, и т.д.). При запросе данных от клиента серверное приложение обращается к БД и получает необходимую информацию, которую в последующем «упаковывает» в формат JSON [7] и передает клиенту.

Третьим компонентом являются вычислительные модули, обеспечивающие решение прикладных задач. Этот компонент содержит в себе выполняемые файлы, которые способны решать различные расчетные задачи. При получении запроса от клиента на выполнение расчетов, серверному приложению передаются необходимые данные. Эти данные отправляются серверным приложением на обработку к соответствующим вычислительным модулям. После выполнения расчетов модули передают их результат серверной части. Далее, серверная часть выполняет следующие действия:

- сохраняет результат расчетов в БД;
- преобразует результат расчетов в формат JSON;
- посылает клиенту результат расчетов в формате JSON.

Модель данных. Модель данных представлена в формате JSON. Этот формат является универсальной структурой данных, которая содержит в себе анонимные объекты: записи, массивы, числа, строки и литералы. JSON обеспечивает создание гибкой структуры данных.

При работе с приложением пользователь создает элементы ИЭС и заполняет информацию об этих элементах. В результате формируется модель ИЭС, содержащая элементы разного типа. Каждый тип элемента обладает своими уникальными параметрами. Если придерживаться строгой (фиксированной) структуры модели, то при создании и записи информации об элементе необходимо записывать элементы в качестве универсальных объектов, которые содержат все параметры. Так как каждый тип элемента не имеет всех параметров, то возникает ситуация, когда в модели накапливается излишняя информация, которая не используется и которую необходимо передавать на сервер, что значительно повышает нагрузку при запросах к серверу.

Гибкость предлагаемого подхода с использование формата JSON не только исключает излишнюю информацию в модели, но также позволяет при необходимости гибко добавлять новые параметры к элементам, не затрагивая при этом другие типы элементов. Так как модели содержат только необходимые параметры, то это упрощает отображение параметров для пользователя. На экран выводится вся информация об объекте, и нет необходимости скрывать неиспользуемые параметры.

Элементы управления. Компонент «элементы управления» имеет окно редактирования параметров, строку состояния, меню управления и интерактивную область для работы (рисунок 2).

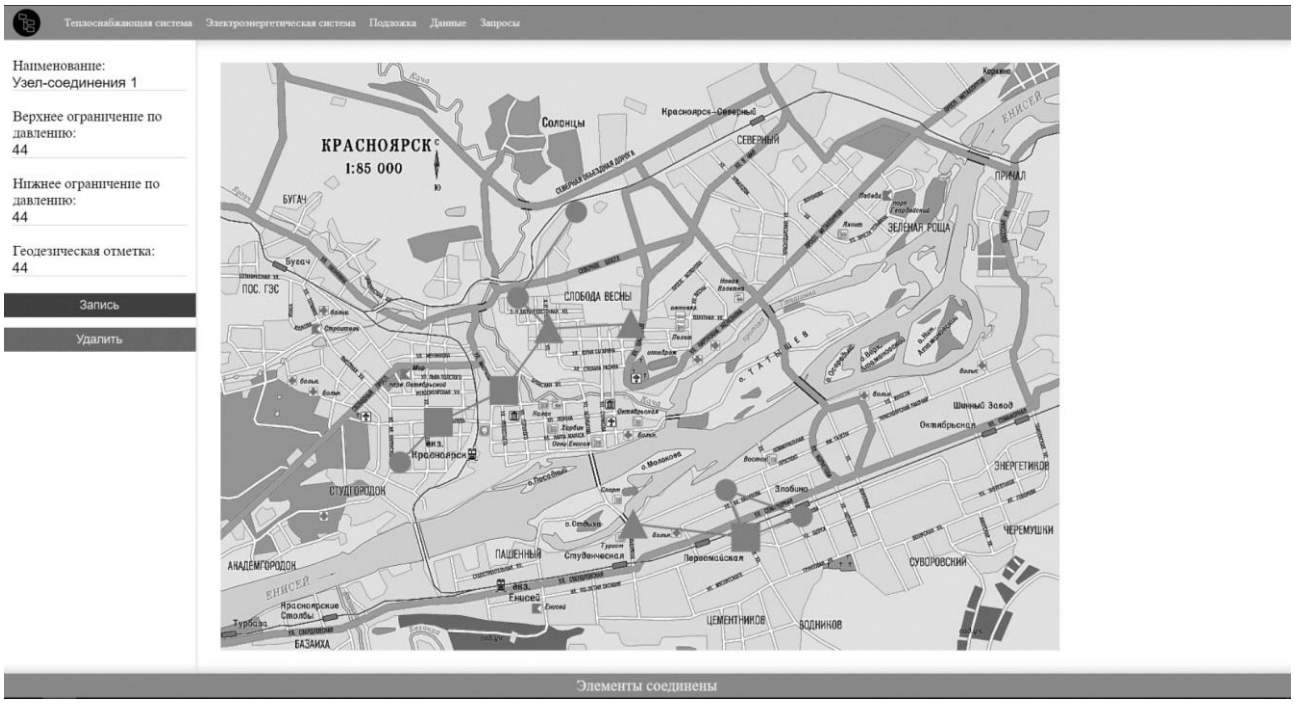


Рис. 2. Пользовательский интерфейс

Основная работа пользователя осуществляется с компонентом «элементы управления». При работе пользователь может изменять параметры элементов ИЭС, создавать ее новые элементы, отправлять запросы серверу и выполнять другие действия. Эти действия выполняются средствами пользовательского интерфейса.

При совершении определенного действия пользователем запускается соответствующая функция, которая выполняет свою задачу. В качестве примера представим последовательность действий приложения при создании нового элемента пользователем. Пользователь выбирает нужную ему энергетическую систему, входящую в состав ИЭС, и тип ее элемента из предложенного списка, указывает то место, в котором нужно поместить новый объект. После этого «элементы управления» обращаются к необходимой функции, которая располагается в главном приложении клиента. При создании нового объекта формируется новый «пустой» объект соответствующего типа с пустыми параметрами. Этот объект добавляется в модель данных. Далее, по цепочке запускается следующая функция, которая при помощи компонента «D3» выполняет визуализацию добавленного объекта. Далее, пользователь может изменить параметры созданного элемента.

Компонент D3. Компонент «D3» отвечает за визуализацию и привязку данных модели к отрисованным элементам при помощи библиотеки d3.js. Средствами этой

библиотеки создается интерактивная работа пользователя. К отображаемым графически элементам привязываются различные обработчики, которые позволяют работать с системой интерактивно. Примерами интерактивной работы являются: увеличение и уменьшение подложки (карты), масштабирование всей модели, перетягивание элементов (drag-and-drop) и прочие действия в интерактивной области приложения.

Приложение на JavaScript. Этот компонент является «ядром» клиентской части. Помимо того, что он содержит в себе модель данных, он также включает все необходимые для работы приложения функции. Этот компонент можно рассматривать как библиотеку функций, которая подключается к веб-странице и организует работу приложения. Без этого компонента невозможно накапливать информацию в модели данных и графически отображать элементы ИЭС средствами компонента D3, а компонент «элементы управления» не сможет функционировать без функций, включенных в «ядро».

Одной из главных функций рассматриваемого компонента является «общение» с сервером. Клиентская часть приложения содержит функции, которые формируют запросы на сервер, при этом данные «упаковываются» для передачи в формат JSON. Главными запросами являются следующие: сохранение данных на сервере, загрузка данных с сервера, выполнение расчетов по данным.

Серверное приложение. Этот компонент является «ядром» сервера. Серверное «ядро» содержит меньше функций, чем клиентское «ядро». Это связано с тем, что основная работа пользователя осуществляется на клиентской части. Функциями рассматриваемого компонента являются: обработка запросов от клиента, получение данных от клиента, работа с БД, отправка данных клиенту и запуск вычислительных модулей.

Вычислительные модули. Один из главных компонентов приложения – это «вычислительные модули». Это набор сторонних выполняемых файлов, которые могут быть написаны на разных языках программирования. Каждый модуль независим от приложения и может работать без него, как и само приложение может работать без этих модулей.

Главная задача этого компонента – это проведение расчетов. Когда от клиента поступает запрос на расчет, происходит поиск необходимого модуля. Если модуль найден, то он запускается для выполнения расчета с параметрами, значения которых сформированы на основе данных, полученных от клиента. На выходе формируется набор данных, который отправляется на серверное приложение. Этот набор данных преобразуется в формат JSON и отправляется клиенту в качестве ответа на запрос о расчете.

Приведем пример запроса на подсчет количества элементов каждого типа. Сервер получает запрос и выполняет поиск нужного модуля. Если модуль не найден, то клиенту отправляется соответствующий ответ об ошибке. В другом случае, модуль определен, сервер запускает его на исполнение и ждет получения результата расчета. При получении результата расчета сервер формирует ответ и оправляет его клиенту.

База данных. Компонент «База данных» является главных компонентом, отвечающим за хранение и накопление данных. В выбранной нами СУБД (PostgreSQL) имеется возможность работы с форматом данных JSON в ячейках таблицы. Это позволяет хранить определенный массив данных в одной ячейке, и этот массив может состоять из разнородных данных. Появляется возможность рассматривать все сущности типа «узел» и «линия связи» в качестве одной абстрактной сущности, представляющей из себя массив анонимных объектов. При работе с БД будет формироваться массив в формате JSON,

который содержит информацию о разнотипных узлах и линиях связи, и этот массив будет записываться в БД.

Такой подход обеспечивает гибкость добавления и редактирования сущностей. Применяя этот подход, отпадает необходимость в создании большого количества таблиц. Средства PostgreSQL позволяют создавать специальные запросы для работы с данными в формате JSON. Все это обеспечивает эффективную работу с данными из БД.

Заключение. В данной статье рассматривается вопрос разработки архитектуры веб-приложения для проведения исследований ИЭС. Клиент-серверная архитектура позволяет построить распределенное веб-приложение. Пользователь, при наличии доступа к сети

Интернет, может пользоваться веб-приложением удаленно для проведения необходимых расчетов без загрузки и установки приложения, что обеспечивает возможность эффективной работы пользователя.

Современные веб-технологии позволяют создавать интерактивные приложения с использованием технологий Drag-and-drop. Пользователь размещает элементы в любой точке рабочей области, а наличие подложки в виде карты местности обеспечивает удобство в моделировании ИЭС.

Распределенные и независимые вычислительные модули выполняют расчеты при решении прикладных задач исследования ИЭС. Существует возможность проводить ресурсоемкие расчеты с использованием технологий параллельных вычислений.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. Voropai N.I., Stennikov V.A., Barakhtenko E.A. Methodological principles of constructing the integrated energy supply systems and their technological architecture // Journal of Physics: Conference Series. 2018. Vol. 1111, N1. P. 012001.

2. Data-Driven Documents: [Электронный ресурс] // URL: https://d3js.org/ (дата обращения: 10.02.2019).

3. Современный учебник JavaScript: [Электронный ресурс] // URL: https://learn.javascript.ru/ (дата обращения: 08.01.2019).

4. Flask is a microframework: [Электронный ресурс] //URL: http://flask.pocoo.org/ (дата обращения: 21.02.2019).

5. Python is a programming language: [Электронный ресурс] // URL: https://www.python.org (дата обращения: 15.02.2019).

6. PostgreSQL: The World's Most Advanced Open Source Relational Database: [Электронный ресурс] //URL: https://www.postgresql.org/ (дата обращения: 25.02.2019).

7. Introducing JSON: [Электронный ресурс] // URL: https://www.json.org/ (дата обращения: 15.02.2019).

УДК 005.591.6

«ПАТТЕРН АВТОМАТИЗАЦИИ» ПОДДЕРЖКИ РЕШЕНИЯ ОРГАНИЗАЦИОННО-РАСПОРЯДИТЕЛЬНЫХ ЗАДАЧ

Засядко А.А., Лукьянов Н.Д.
(г. Иркутск, Иркутский национальный исследовательский технический университет)
e-mail: zasyadko-aa@rambler.ru, lukyanovnd@istu.edu

Аннотация. Излагаются результаты разработки подхода к решению проблемы распределения обязанностей между сотрудниками небольших компаний посредством создания нового инструмента – «паттернов автоматизации» для проектной поддержки внедрения информационных систем и технологий. Приводятся примеры решения организационно – распорядительных задач с помощью формирования ряда паттернов автоматизации на примере реализации функций конкретного учреждения.

Ключевые слова. Организационно-распорядительные задачи, автоматизация, регламенты, бизнес-процессы, проекты, внедрение информационных систем и технологий.

«AUTOMATION PATTERN» OF THE SUPPORT OF SOLVING ORGANIZATIONAL AND ADMINISTRATIVE TASKS

Zasyadko A.A., Lukyanov N.D.
(Irkutsk, Irkutsk National Research Technical University)
e-mail: zasyadko-aa@rambler.ru, lukyanovnd@istu.edu

Abstract. The paper describes the results of developing an approach to solving the problem of the distribution of responsibilities between employees of small companies by creating a new tool – «patterns of automation» for project support for the implementation of information systems and technologies. Examples of solving organizational and administrative tasks using a number of automation patterns on the example of the implementation of the functions of a particular institution are given.

Keywords. Organizational and administrative tasks, automation, regulations, business processes, projects, deployment of information systems and technologies.

Введение. Многие компании и предприятия малого бизнеса, муниципальные и казенные учреждения, другие организации с небольшой численностью персонала сталкиваются со сложной проблемой – осуществление автоматизации своей деятельности, которую вынужденно должны выполнять специалисты этих организаций. Сотрудники, да и руководители данных организаций, в том числе муниципальных служб нередко, не обладая достаточными компетенциями или опытом в области информационных технологий, как правило, не имея возможности содержать в своем составе необходимых специалистов по информационным технологиям (ИТ), часто, не получая ресурсов для обращения к услугам сторонних организаций по внедрению ИТ и пр. испытывают затруднения в решении многих задач автоматизации бизнес-процессов, особенно организационно-распорядительного характера, связанных с управлением людьми, в частности посредством административных методов (с помощью приказов, распоряжений, указаний и т.п.). Занимаясь организационно-распорядительными задачами путем традиционного менеджмента, руководство и персонал исполнителей понапрасну теряет свое время и силы, не полностью реализует свой потенциал без привлечения богатого арсенала средств автоматизации путем применения, в частности, информационных систем и технологий. Проблема проведения успешной автоматизации

решения этих задач усугубляется тем, что используемые при решении организационно-распорядительных задач компьютерные информационные системы, не только популярного класса Task - и Project Management (менеджмента задач и проектов), но классов с более богатыми функциями CRM (управления взаимоотношениями клиентов) и XRM (управления взаимоотношениями всех со всеми), встречают при внедрении определенные (условно говоря) «информационно-знаниевые барьеры».

Постановка задачи исследования. В связи с этим актуальной проблемой является возможность создания некоторого инструмента, позволяющего обеспечивать согласование процедур, с одной стороны, формализации действий пользователя и, с другой стороны, аналитикой программного обеспечения в целях облегчения внедрения информационных систем (ИС).

Сегодня разворачивающаяся в стране «цифровая трансформация экономики», хоть и содержит потенциал для роста организаций, выдвигает серьезные угрозы. Социологи прогнозируют, что массовые рутинные операции и процедуры в ближайшее время начнут окончательно вытесняться из бизнеса, а сотрудники, не способные выполнять ничего, кроме подобных действий, будут увольняться. Однако существует множество организаций, в которых цифровые компетенции хоть и важны, но приоритетами являются особые функции ряда сотрудников, которые не подлежат автоматизации; их функции не могут быть принципиально вытесненными цифровыми технологиями. Не умеющие создавать цифровые ценности компании окажутся в уязвимом положении перед конкурентами, умеющими делать это. Поэтому, для обеспечения организациям цифровых ценностей, располагающих, как «ручными» (вытесняемыми), так и не автоматизируемыми (сохраняющимися) компетенциями, приходится обращаться к обновлению известного инструментария обеспечения автоматизации процессов их управленческой деятельности.

В подобной ситуации оказываются многие организации, в частности, муниципальное казенное учреждение «Информационно-туристская служба г. Иркутска» (далее МКУ ИТС), которое столкнулось с проблемой организационного характера в части распределения обязанностей между сотрудниками при использовании современных средств автоматизации и на примере которого рассматривается подход к разработке интересующего нас инструмента поддержки решения организационно-распорядительных задач.

Основная деятельность учреждения – предоставление информационно-консультационных услуг гостям города по вопросам пребывания в г. Иркутске и на озере Байкал, в т.ч. на иностранных языках. Помимо этого, на сотрудников, возлагаются различные служебные обязанности по организации и оказанию многочисленных услуг, выдаются задания и поручения по разработке и реализации разнообразных проектов, связанных с задачами учреждения. Выполняемые услуги и работы (в проектах) составляют различного рода организационные задачи, подлежащие распределению между сотрудниками с учетом их обязанностей (компетенций) и загруженности. Однако, сотрудники имеют, как правило, образование в области лингвистики или получили подготовку в сфере бизнеса, экономики и менеджмента, что достаточно далеко от компетенций в сфере информационных систем и технологий не только внедрения, но и использования.

Обсуждение результатов. Был проанализирован комплекс наиболее распространенных из числа основных функциональных задач, решаемых рассматриваемой службой, который представлен в виде блок-схемы на рис. 1.

В комплекс организационно-распорядительных задач, порождающих проекты, в которые привлекаются сотрудники на различных ролях и с различными поручениями, входят:

- Задача 1: Ведение проектов системы уличной навигации, мониторинг и её содержание;
- Задача 2: Продвижение работ и услуг в социальных сетях и на сайте учреждения;
- Задача 3: Подготовка и выпуск печатной продукции (путеводители, карты, каталоги и т.п.);
- Задача 4: Организация, проведение и участие в выставочных мероприятиях;
- Задача 5: Представление информационно-консультационных услуг гостям по пребыванию;
- Задача 6: Организация, обустройство и обеспечение выносных офисов в гостевой сезон;
- Задача 7: Подготовка муниципальных контрактов, конкурсной и аукционной документации;
- Задача N: Другие проекты учреждения и их задачи.

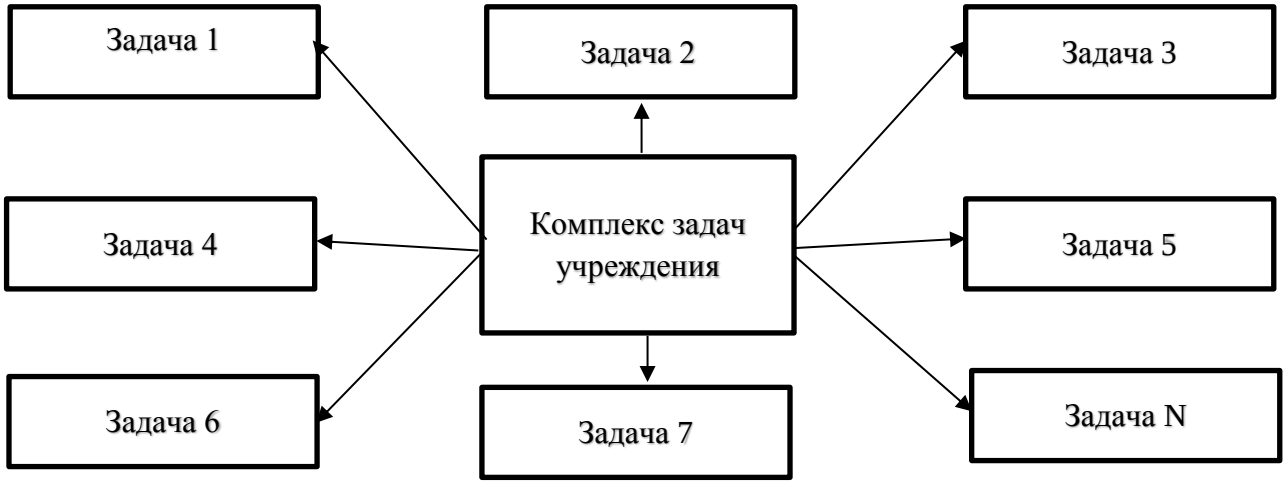


Рис. 1. Блок-схема организационно-распорядительных задач учреждения

Рассматриваемое учреждение представляет собой типичный образец компании, где имеется проблема организационного характера в части распределения обязанностей по оказанию услуг и выполнению проектов, которая обусловлена дисбалансом между числом и компетенциями сотрудников, объемом и спецификой задач, а также сроком и временем выполнения поручений, составляющих эти задачи. При этом решаемые задачи являются не только весьма многочисленными и разнообразными, но и выполняются ограниченным персоналом, которые должны нести самые различные обязанности: от переводчиков иностранного языка (на китайском, английском, немецком и т.д.) до менеджеров проектов, не обладающих, вместе с тем, требуемой компьютерной грамотностью (т.е. в качестве опытных пользователей ПК).

На эти сложные условия деятельности накладывается сезонные всплески трудозатрат, так как в летний период поток туристов резко нарастает и не может быть обслужен без

автоматизации ряда своих бизнес-процессов и функций. Такого рода сложные управленческие задачи являются сегодня не редкостью на рынке, и со временем будут нарастать их количество и сложность, что требует формирования и привлечения специального инструментария, порождающего возможности комплексной автоматизации для поддержки деятельности учреждений и компаний.

Особенностью развиваемого подхода по построению необходимого инструмента для обеспечения автоматизации является системное рассмотрение бизнес-процедур функциональных процессов, которые выстраиваются в общей совокупности или в едином сопряжении ручных и компьютеризированных действий, разрешающих рассматриваемые задачи в интересующих проектах [1]. Данные процедуры могут выступать в качестве своеобразных шаблонов, поддерживающих внедрение программных продуктов автоматизации организационно-распорядительных задач и облегчающих построение современных информационных систем управления задачами и проектами в учреждениях неинформационного характера.

В основу построения рассматриваемого шаблона взято понятие «регламента» [2]. «Регламент» - это документ, который перечисляет и описывает по порядку этапы (шаги), необходимые их участниками для выполнения бизнес-процессов. Несмотря на то, что регламентом является достаточно известная совокупность правил, устанавливающих порядок проведения работ или осуществления деятельности (ГОСТ 52294-2004), пусть даже определяемый в электронной форме, однако при автоматизации и прежде всего при внедрении информационных систем про него почему-то забывают.

Как следствие автоматизации подвергаются сплошным образом все бизнес-процессы учреждения или каждый из них в целом, что затрудняет, как минимум, внедрение информационных систем, а, как максимум, приводит к пренебрежению возможностями автоматизации ввиду неудобства использования ею со стороны пользователей из числа обычных (неквалифицированных) сотрудников учреждения.

Решением рассматриваемого инструмента является следующее представление. Есть простое правило: любое повторяемое действие в компании должно быть регламентировано. Конечно, регламенты для описания бизнес-процессов широко используются на практике, их зачастую применяют для моделирования и построения информационных систем (например, продукты класса BMP, методология SADT, язык UML и пр.). Но, вместе с тем, для определения содержания и порядка необходимых действий, которые должны предпринимать исполнители бизнес-процессов одновременно с учетом процессов поддержки их действий на основе средств автоматизации, такого рода регламенты отсутствуют. По существу, такие комплексные регламенты выступают связывающим компонентом (в виде некоего «моста») между действиями человека-исполнителя, выполняющего «вручную» ряд этапов бизнес-процесса, и автоматизированными действиями с помощью ЭВМ с соответствующим программным обеспечением по остальным этапам бизнес-процесса.

Эти повторяющиеся регламенты действий с учетом сценария распределения обязанностей будут выступать в качестве инструмента поддержки внедрения информационных систем, и поэтому были названы «паттерном автоматизации». Как известно, «Паттерн» (*pattern*) – повторяющийся шаблон, образец, например, способа решения задачи, стереотипа действий процесса и т.п. Таким образом, предлагается

разрабатывать, так называемые «паттерны автоматизации», призванные обеспечивать поддержку решения организационных проектов компаний, а именно, помогая распределять обязанности между исполнителями по возникающим в проектах задачам и подключать к получению результатов на целесообразных этапах бизнес-процессов автоматизированные информационные системы.

Вследствие подготовки и применения «паттернов автоматизации» облегчается и ускоряется разработка и исполнение типовых проектов компании, а также обеспечивается, вместе с тем, не только взвешенное распределение работ между сотрудниками, но и рациональное подключение автоматизации к выполнению определенной части работ (избранных этапов) бизнес-процессов. Вместе с тем, «паттерны автоматизации» позволяют снять или снизить неблагоприятное влияние упомянутых выше «информационно-знаниевых барьеров», облегчить внедрение информационных систем, сэкономить силы и ресурсы.

Ниже представляются для формирования интересующих «паттернов автоматизации» разработки определенных регламентов в обеспечение проектов для первых трех задач организационно-распорядительного характера.

А) «Проект установки указателя системы уличной навигации» (в рамках решения указанной выше задачи № 1) включает следующие процедуры:

1. Определение места, где необходим указатель и его направление
    1.1. Привлечение внештатного сотрудника
    1.2. Подготовка постановочного эскиза
2. Создание дизайн проекта – указателя
3. Планирование закупок:
    3.1. Покупка по счету
    3.2. Объявление аукциона
4. Определение поставщиков
5. Заключение и исполнение муниципального контракта:
    5.1. Контроль за выполнением заказа
    5.2. Прием результатов.

Соответствующий «паттерн автоматизации» для приведенного проекта показан в виде диаграммы Ганта на рис.2 (даты и продолжительность действий отдельных процедур установлены условно для демонстрации результатов решения задачи).

ИД	Название задачи	Начало	Конец	Длительность
1	Определение места где необходим указатель и его направление	18.12.2018	21.12.2018	4д
2	Привлечение внештатного сотрудника	24.12.2018	26.12.2018	3д
3	Подготовка постановочного эскиза	27.12.2018	27.12.2018	1д
4	Создание дизайн проекта - указателя	28.12.2018	03.01.2019	5д
5	Планирование закупок	04.01.2019	07.01.2019	2д
6	Оплата по счёту/Объявление аукциона	08.01.2019	10.01.2019	3д
7	Заключение муниципального контракта	11.01.2019	14.01.2019	2д
8	Приём результатов	15.01.2019	15.01.2019	1д

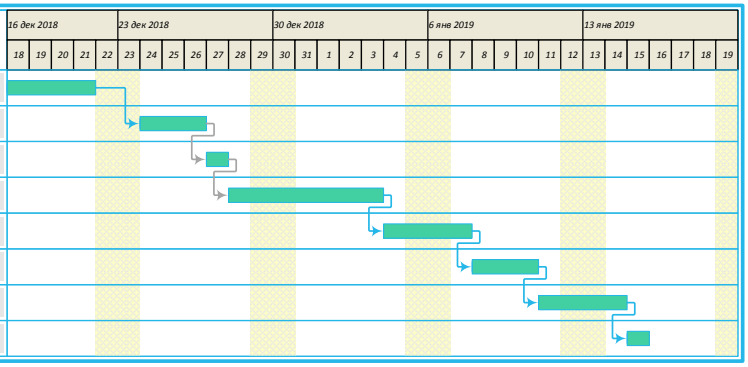
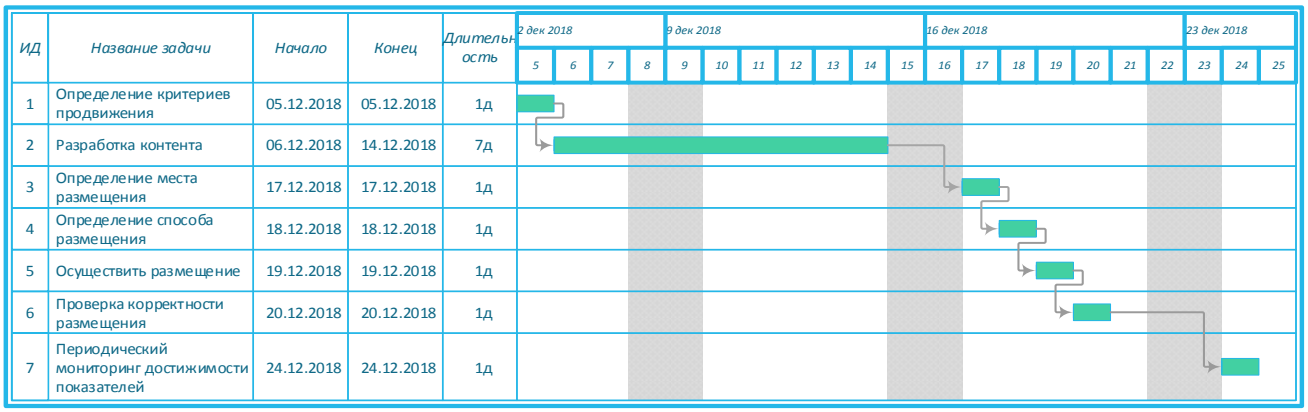


Рис. 2. Диаграмма Ганта «паттерна автоматизации» по задаче №1

Б) «Проект продвижения работ и услуг в социальных сетях и на сайте учреждения» (в рамках решения указанной выше задачи № 2) включает следующие процедуры:

1. Определение продвигаемых работ и услуг, критериев их продвижения в сетях (на сайте)
2. Разработка (сбор материалов) для контента, размещаемого в сети (на сайте)
3. Определение места размещения контента
4. Определение способа размещения контента (поддержка SEO, SMM)
5. Осуществить размещение контента в сети (на сайте)
6. Проверка корректности размещения контента
7. Периодический мониторинг достижения критериев.

Соответствующий «паттерн автоматизации» для приведенного проекта показан в виде диаграммы Ганта на рис.3 (даты и продолжительность действий отдельных процедур установлены условно для демонстрации результатов решения задачи).



ИД	Название задачи	Начало	Конец	Длительность
1	Определение критериев продвижения	05.12.2018	05.12.2018	1д
2	Разработка контента	06.12.2018	14.12.2018	7д
3	Определение места размещения	17.12.2018	17.12.2018	1д
4	Определение способа размещения	18.12.2018	18.12.2018	1д
5	Осуществить размещение	19.12.2018	19.12.2018	1д
6	Проверка корректности размещения	20.12.2018	20.12.2018	1д
7	Периодический мониторинг достижимости показателей	24.12.2018	24.12.2018	1д

Рис. 3. Диаграмма Ганта «паттерна автоматизации» по задаче №2

В) «Проект подготовки и выпуска печатной продукции учреждения» (в рамках решения указанной выше задачи № 2) включает следующие процедуры:

1. Определение перечня печатной продукции
2. Разработка дизайн – макета издания
3. Поиск поставщика перевода на иностранные языки
4. Покупка по счёту или объявление аукциона
5. Определение издателя
6. Покупка по счёту или объявление аукциона
7. Заключение муниципального контракта
8. Планирование закупок
9. Приём результатов.

Соответствующий «паттерн автоматизации» для приведенного проекта показан в виде диаграммы Ганта на рис.4.

ИД	Название задачи	Начало	Конец	Длительность
1	Определение перечня продукции	05.12.2018	10.12.2018	4д
2	Разработка дизайн макета	17.12.2018	19.12.2018	3д
3	Поиск поставщика перевода	20.12.2018	20.12.2018	1д
4	Покупка по счёту/Объявление аукциона	21.12.2018	27.12.2018	5д
5	Определение издателя	31.12.2018	01.01.2019	2д
6	Покупка по счёту/Объявление аукциона	02.01.2019	15.01.2019	10д
7	Заключение муниципального контракта	16.01.2019	18.01.2019	3д
8	Планирование закупок	21.01.2019	25.01.2019	5д
9	Приём результатов	28.01.2019	28.01.2019	1д

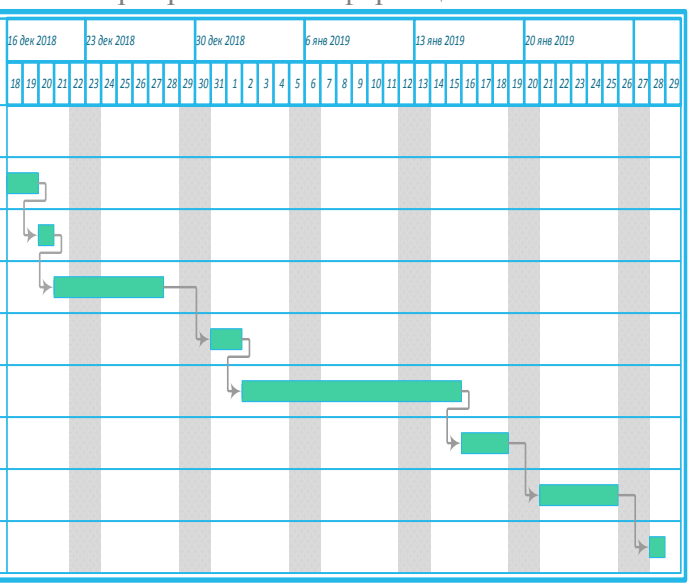


Рис. 4. Диаграмма Ганта «паттерна автоматизации» по задаче № 3 «Проект подготовки и выпуска печатной продукции»

В рассмотренном учреждении планируемые проекты и возникающие в их составе процедуры решения задач, предусматриваются к реализации путем задействования одного из популярных программных продуктов на базе использования информационной автоматизированной системы (АИС) «Trello» для распределения отдельных действий (поручений) между всеми сотрудниками в рамках организации работ проектных команд [3].

Trello – АИС, разработанная Fog Creek Software, позиционируемая для управления проектами, которая построена на парадигме, известной как «канбан» - методология инноваций, впервые опробованная на фирме Toyota в 1980-х г. для управления цепочками поставок продукции, и ныне популяризированная во всем мире. «Канбан» - интуитивно понятный способ организации потока работ, который помогает, как считают его создатели: «делать больше, работать быстрее», а главное, выполнять задачи «точно в срок».

В переводе с японского языка «канбан» означает «бирка», «значок», которые в концепции построения Trello превратились в понятие «карточек» для донесения пользователям информации о конкретных работах и состоянии их выполнения. «Карточки» служат основой решения задач организационно-распорядительного плана при управлении проектами.

Разработка «паттернов автоматизации» тесно связана с требуемыми для задач формализуемыми алгоритмами управления бизнес-процессов (путем адаптации типовых регламентов к конкретным наборам необходимых действий при решении задач). В работе в качестве алгоритмов управления рассматриваются методы решения задач, прежде всего, в виде линейных структур, достаточно легко отображаемых простыми диаграммами (графиками) Ганта.

В перспективе алгоритмы управления могут представляться в виде более сложных структур – «дерева», где предполагаемые действия определяются в качестве как «узлов», так и «листьев» (структур графов). Наконец, целесообразным представляется создание библиотек «паттернов автоматизации», коллекции которых могут удовлетворять потребителей наборами решений задач.

Выводы. Разрабатываемый подход к решению проблемы распределения обязанностей между сотрудниками небольших компаний может быть поддержан посредством создания эффективного инструмента – «паттернов автоматизации». Этот подход в состоянии обеспечивать рациональным образом проектную поддержку и облегчать решение организационно-распорядительных задач, тем самым, должен содействовать внедрению информационных систем и технологий, что весьма актуально для развития идей цифровой экономики для большой группы заинтересованных организаций.

Авторы благодарят студентов Исаченко А.А., Михеева А.В., Первушина Е.А., Петроченко Л.Н., Суюнову М.Ю. за участие в работе, представленной в рамках Корпоративного акселератора ИРНИТУ-2108.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. Рыбаков М. Бизнес-процессы: как их описать, отладить и внедрить // М.: Издательство Михаила Рыбакова, 2016. – 392 с.
2. Система регламентов в компании, как краеугольный камень эффективности руководителя [Электронный ресурс]// URL: http://openstud.ru/blog/for-businessmen/system-of-reglaments/ (дата обращения: 15.05.2019)
3. «Начало работы с Трелло» [Электронный ресурс]// URL: https://trello.com/guide (дата обращения: 15.05.2019)

УДК 004.4'236

РАЗРАБОТКА ИНСТРУМЕНТАЛЬНОГО СРЕДСТВА ДЛЯ АНАЛИЗА КОНФИГУРАЦИЙ 1С: ПРЕДПРИЯТИЕ

Копылова В.В.[1], Черкашин Е. А.[2]
[1] (г. Иркутск, Иркутский национальный исследовательский технический университет)
[2] (г. Иркутск, Институт динамики систем и теории управления им. В.М. Матросова)
e-mail: kopylovika@mail.ru ,eugeneai@icc.ru

Аннотация. В данной статье рассматривается вопрос анализа структуры конфигураций 1С:Предприятие. Представлена общая схема сравнения объектов конфигурации и визуализации результатов этого сравнения в виде дендрограммы. Показаны примеры приложения и интерпретации результатов.

Ключевые слова. Иерархический кластерный анализ, 1С: Предприятие, структура конфигурации, средство анализа.

DEVELOPMENT OF A CONFIGURATION TOOL FOR ANALYSIS OF 1C:ENTERPRISE

Kopylova V.V.[1], Cherkashin E.A.[2]
[1] (Irkutsk, Irkutsk National Research Technical University)
[2] (Irkutsk, Matrosov Institute for System Dynamics and Control Theory of Siberian Branch of Russian Academy of Sciences)
e-mail: kopylovika@mail.ru ,eugeneai@icc.ru

Abstract. This article discusses the analysis of the structure of 1C:Enterprise configurations. The outline of the comparison of configuration objects is presented as well as visualization of the

result of this comparison in the form of dendrogram. Application examples and the interpretation of results are demonstrated.

Keywords. Hierarchical cluster analysis, 1C:Enterprise, configuration structure, analysis tool.

Введение. Компании, осуществляющие разработку программного обеспечения для автоматизации бизнес-процессов предприятий с использованием системы 1С, наработали огромный объем программного кода, реализующего различные сферы деятельности. Компания 1С разрабатывает и выпускает множество прикладных решений не только для автоматизированного бухгалтерского, оперативного и управленческого учета на предприятиях, но и для автоматизации складского, кадрового учета, производства, и т.п. Прикладные решения позволяют оперативно формировать отчеты, получать данные, обрабатывать большой объем информации. Самым популярным прикладным решением в России в настоящее является 1С:Бухгалтерия. Внедрение и эксплуатация этого прикладного решения на предприятии позволяет существенно экономить время на ведение бухгалтерского учета, оптимизировать численность, а также повышать производительность сотрудников, занятых ведением учета.

Структура любого прикладного решения 1С включает платформу, конфигурацию и базу данных. Рассмотрим подробнее, в чем состоит их отличие [1].

Платформа – это основа системы и среда разработки конфигураций. Платформа получила название 1С:Предприятие. Платформу разрабатывает отдельный от разработки конфигурации штат программистов.

Конфигурация – это, программа, разработанная на платформе 1С Предприятие, выполняющая конкретные учетные задачи предприятии. Самая популярная конфигурация фирмы 1С – 1С:Бухгалтерия.

Конфигурации имеют релизы или редакции – выпуск обновления действующей конфигурации, который связан с развитием прикладного решения, усовершенствованием различных функций, а также изменением действующего законодательства или требований рынка. Новая версия конфигурации не несет существенных изменений в строении программы с точки зрения кода.

База данных формируется за счет информации, которую пользователи вводят в систему при помощи интерфейса пользователя, предоставляемого прикладным решением, и хранится на определенном носителе – компьютере или сервере. Используя другие программные инструменты информацию из БД можно обрабатывать, корректировать, удалять и пр.

Сама по себе платформа не может выполнять никаких задач пользователя, так как она предназначена для обеспечения работы конфигурации и выполнения ряда других функций. Но также и конфигурация не будет работать без установки платформы. Они работают только в комплексе.

Каждое прикладное решение представляет собой сложную структуру данных, которая интерпретируется в рамках технологической платформы 1С. Однако беглое визуальное сравнение конфигураций показывает, что различные конфигурации разрабатываются независимо друг от друга. Кроме того, в рамках одной конфигурации объектная структура

представляет собой неиерархический список структур, обладающих общими свойствами, что приводит к копированию структурной и алгоритмической информации и, соответственно, затрудняет сопровождение программного кода.

Целью исследования в данной работе является создание методики анализа структуры и свойств объектов конфигурации 1С, разработка программной системы, позволяющей сравнивать структуру и свойства различных объектов конфигураций, а также визуализировать результаты этого сравнения.

В задачи исследования входят:

1. Ознакомление с форматом XML-файла выгрузки конфигурации и создание процедуры импорта данных о структуре объектов в систему;
2. Разработка функции сравнения структур объектов;
3. Построение дендрограммы при помощи иерархических методов кластерного анализа, визуализация этой структуры;
4. Интерпретация и анализ полученных результатов.

Методика анализа структуры и свойств объектов конфигураций 1С. За время своего существования на платформе 1С количество функционирующих копий конфигураций постоянно увеличивается. Изменяется также Российское законодательство, что требует оперативного улучшения каждой копии продукта. Автоматизация предприятий, в общем случае, требует доработок общего решения, делая его более индивидуальным. Всё это способствует увеличению объектов в конфигурации, что может привести к дублированию функций одних объектов в других, вплоть до полных функциональных эквивалентов. Т.е. возможно построение избыточных объектов в конфигурации. Анализ конфигурации на схожесть объектов друг с другом позволяет выявлять объекты-дубликаты.

Другой возможностью использования процедуры анализа является поддержка труда начинающего программиста: знакомые объекты сравниваются с другими, и, в некоторой мере, по результатам сравнения индуктивно производится оценка свойств незнакомых объектов. Такой подход, по нашему мнению, позволит повысить производительность труда начинающих программистов.

Хоть это не свойственно платформе 1С, в большинстве программных систем, поддерживающих объекты и объектно-ориентированное программирование структуры (классы) находятся друг с другом в отношении «наследования» свойств. И эти отношения можно представить в виде иерархической структуры – N-арного дерева. Такая иерархия является естественной, так как профессиональные программисты всегда стараются вводить абстракции, если есть два класса объектов, схожих по структуре и свойствам, при этом создается общий класс-предок, куда помещаются общие поля и методы обоих классов, что позволяет избегать дублирования в реализации этих элементов. С другой стороны, в отлаженном программном коде при создании новых функций у существующих классов можно создавать классы-потомки, где уже и реализуются новые функции.

В данном исследовании будем предполагать, что объектам конфигурации 1С соответствуют одноименные классы, находящиеся в некоторой иерархической структуре наследования, а эти объекты являются единственными экземплярами этих классов. Таким образом, под схожестью двух объектов будем далее понимать свойства месторасположения соответствующих классов в этой иерархии. Объекты схожи друг с другом в большей

степени, если между их классами расположено меньше структурных элементов (вершин и дуг) по пути, соединяющем эти классы.

Аналогичным свойством обладают результаты алгоритмов таксономии (кластеризации). Чтобы построить дендрограмму, необходимо создать трехаргументную функцию, которая возвращает 0, если оба аргумента являются идентичными (в частности, одним и тем же), и некоторое положительное число, если объекты не совпадают, причем, чем больше объекты «не совпадают», тем это число будет больше. Такая функция выполняет роль *меры различия* (dissimilarity) двух объектов.

Иерархическая кластеризация – совокупность алгоритмов упорядочивания данных, направленных на создание дерева вложенных кластеров. Выделяют два класса методов иерархической кластеризации [3]:

- Агломеративные (соединяющие) методы: новые кластеры создаются путем объединения более мелких кластеров и, таким образом, дерево создается от листьев к стволу;
- Дивизивные (разъединяющие) методы: новые кластеры создаются путем деления более крупных кластеров на более мелкие и, таким образом, дерево создается от ствола к листьям.

В данном исследовании в качестве меры различия используются обобщенные характеристики структуры объекта: количество реквизитов (полей или атрибутов), количество табличных частей и количество реквизитов табличных частей (атрибутов в таблицах). Эти параметры составляют три координаты евклидова пространства объектов. Исходные данные об объектах загружаются из XML-файла. Дендрограмма строится агломеративным методом, объединяя сначала наиболее схожие объекты в множества. Корневая вершина дендрограммы – это, собственно, и есть конфигурация.

В качестве примера рассмотрим простую конфигурацию, специально разработанную для тестирования предлагаемой методики. Конфигурация состоит из пяти справочников: Валюты, Страны, Контрагенты, Номенклатура и Сотрудники (рисунок 1).

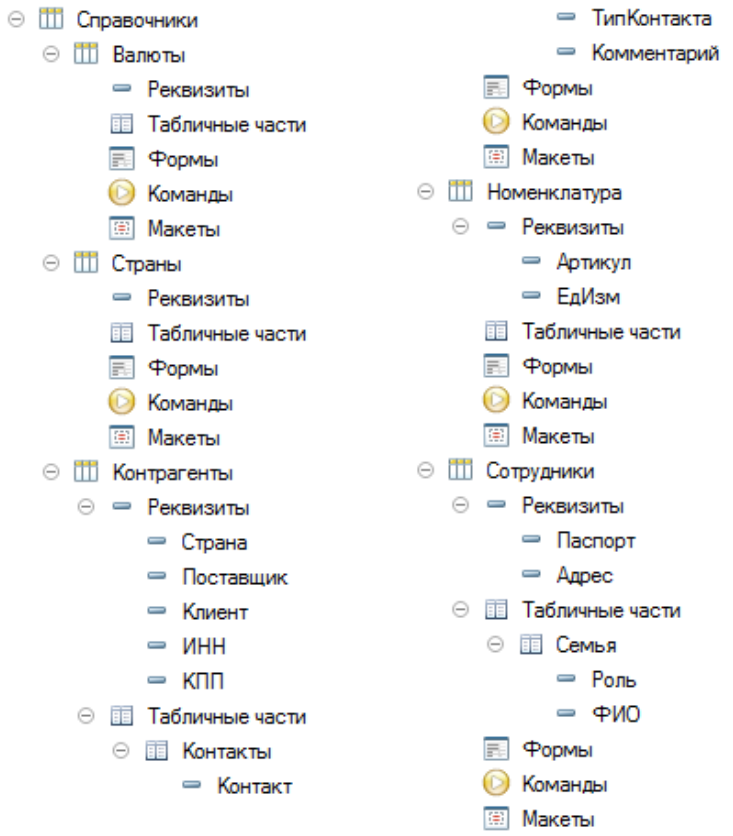


Рис.1. Состав тестовой конфигурации

Справочники Валюты и Страны состоят из реквизитов «Пометка Удаления», «Код» и «Наименование». Реквизитов у справочника Номенклатура больше: «Пометка Удаления», «Код», «Наименование», «Артикул» и «Единицы измерения». Структуры справочников Контрагенты и Сотрудники включают ещё и табличную часть. Контрагенты – реквизиты «Пометка Удаления», «Код» «Наименование», «Страна», «Поставщик», «Клиент», «ИНН», «КПП»; табличная часть – Контакты, состоящая из реквизитов «Контакт», «Тип Контакта», «Комментарий». Сотрудники – реквизиты «Пометка Удаления», «Код» «Наименование», «Паспорт» и «Адрес»; табличная часть – Семья, состоящая из реквизитов «Роль», «ФИО».

Процесс чтения структур объектов в выгруженной в XML-файл конфигурации заключается в последовательном обходе дерева в глубину. Существует два основных способа реализации чтения XML – это DOM и SAX [2]. Первый сначала загружает полностью исходный файл и строит дерево DOM в памяти компьютера, затем программа делает запросы к дереву и получает необходимую информацию. Второй способ позволяет реализовывать обработку данных XML в процессе чтения файла. В данной работе использован способ обработки исходного файла выгрузки, интегрирующий DOM с объектными свойствами языка Python – ElementTree [4]. Библиотека ElementTree представляет DOM в виде объектов языка, а не объектов абстрактной библиотеки низкого уровня, что значительно упрощает обработку исходных XML-данных.

Так как целью чтения XML-документа конфигурации является сбор обобщенных данных о структуре объектов, при нахождении нужного элемента структуры, данные о нем заносятся в соответствующую таблицу локальной базы данных, реализованной при помощи библиотеки SQLite3 [5].

Напомним, что каждый объект конфигурации характеризуется количеством реквизитов, количеством табличных частей, а также количеством реквизитов в табличных частях, что дает евклидово пространство с тремя координатами. Для представленных в рассматриваемом примере объектов координаты представлены в таблице 1.

Таблица 1 – Координаты объектов

Справочник	**Координаты**
Валюты	[3,0,0]
Страны	[3,0,0]
Контрагенты	[8,1,3]
Номенклатура	[5,0,0]
Сотрудники	[5,1,2]

Рассчитаем расстояния (таблица 2) между объектами и построим дендрограмму (рисунок 2).

Таблица 2 – Расстояние между объектами (матрица различий)

	Валюты	Страны	Контрагенты	Номенклатура	Сотрудники
Валюты	0,0	0,0	5,916	2,0	3,0
Страны	0,0	0,0	5,916	2,0	3,0
Контрагенты	5,916	5,916	0,0	4,395	3,162
Номенклатура	2,0	2,0	4,395	0,0	2,236
Сотрудники	3,0	3,0	3,162	2,236	0,0

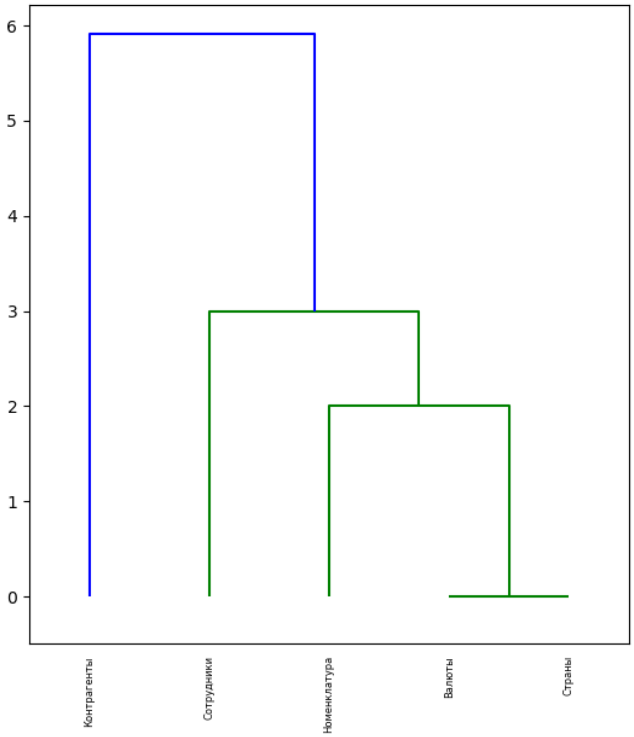


Рис. 2. Дендрограмма

Поскольку все данные о структуре объектов сохранены в базе данных, структура объекта легко восстановима и представима на формальном языке.

```
Справочник                    Реквизит Наименование      Реквизит Наименование
СправочникСсылка.Валюты       Реквизит Страна            Реквизит Артикул
```

```
Реквизит ПометкаУдаления
Реквизит Код
Реквизит Наименование
/Справочник
Справочник
СправочникСсылка.Страны
Реквизит ПометкаУдаления
Реквизит Код
Реквизит Наименование
/Справочник
Справочник
СправочникСсылка.
Контрагенты
Реквизит ПометкаУдаления
Реквизит Код
Реквизит Поставщик
Реквизит Клиент
Реквизит ИНН
Реквизит КПП
ТабличнаяЧасть Контакты
Реквизит Контакт
Реквизит ТипКонтакта
Реквизит Комментарий
/ТабличнаяЧасть
/Справочник
Справочник
СправочникСсылка.
Номенклатура
Реквизит ПометкаУдаления
Реквизит Код
Реквизит ЕдИзм
/Справочник
Справочник
СправочникСсылка.
Сотрудники
Реквизит ПометкаУдаления
Реквизит Код
Реквизит Наименование
Реквизит Паспорт
Реквизит Адрес
ТабличнаяЧасть Семья
Реквизит Роль
Реквизит ФИО
/ТабличнаяЧасть
/Справочник
```

Аналогично можно получить представление структуры данных объекта на другом объектно-ориентированном языке, в качестве которого будем использовать MQL4. MetaQuotes Language 4 (MQL4) – встроенный язык программирования торговых стратегий, разработанный компанией MetaQuotes Software Corp. на основе своего многолетнего опыта в создании торгово-информационных платформ. Этот язык позволяет писать собственные программы-эксперты (Expert Advisors), автоматизирующие управление торговыми процессами и идеально подходящие для реализации собственных торговых стратегий. Кроме того, на MQL4 можно создавать собственные технические индикаторы (Custom Indicators), скрипты (Scripts) и библиотеки функций (Libraries).

Обсуждение результатов. Рассмотрим два одинаковых (согласно заданной нами функции) по структуре справочника «Валюты» и «Страны». Никаких созданных реквизитов или табличных частей они в себе не содержат. Оба справочника имеют стандартные атрибуты «Пометка Удаления», «Код» и «Наименование». Тогда можно создать класс «Базовый» и вынести в него рассматриваемые атрибуты.

Псевдокод	MetaQuotes Language 4 (MQL4)

Справочник СправочникСсылка.Базовый
Реквизит ПометкаУдаления
Реквизит Код
Реквизит Наименование
/Справочник
	Class Base{
	protected:
	bool MarkRemoval
	int Code
	string Name
	}

Справочники будут наследниками созданного класса.

Справочник СправочникСсылка.Валюты наследует СправочникСссылка.Базовый /Справочник	Class Currencies : public Base { }
Справочник СправочникСсылка.Страны	Class Country : public Base {

наследует СправочникСсылка.Базовый /Справочник	}

Продолжая построение классов на основе наследования, получим следующий код.

Справочник СправочникСсылка.Номенклатура
наследует СправочникСсылка.Базовый
Реквизит Артикул
Реквизит ЕдИзм
/Справочник
	Class Nomenclature : public Base
	{
	private:
	string Article
	string EdIzm
	}

Справочник СправочникСсылка.Сотрудники
наследует СправочникСсылка.Базовый
Реквизит Паспорт
Реквизит Адрес
ТабличнаяЧасть Семья
Реквизит Роль
Реквизит ФИО
/ТабличнаяЧасть
/Справочник
	Class Country : public Base {
	private:
	string Passport
	string Address
	table Family {
	string Role
	string name
	}
	}

Справочник СправочникСсылка.Контрагенты
наследует СправочникСсылка.Базовый
Реквизит Страна
Реквизит Поставщик
Реквизит Клиент
Реквизит ИНН
Реквизит КПП
ТабличнаяЧасть Контакты
Реквизит Контакт
Реквизит ТипКонтакта
Реквизит Комментарий
/ТабличнаяЧасть
/Справочник
	Class Country : public Base {
	private:
	Country aCountry
	bool Provider
	bool Customer
	string INN
	string KPP
	table Contacts{
	string Contact
	string TypeContact
	string Comment
	}
	}

Заключение. В статье предложена методика обобщенного анализа структуры объектов конфигурации 1С. Методика основана на построении дендрограммы (агломеративным методом) на основе «схожести» структур объектов. В качестве меры схожести использованы количественные характеристики структуры объектов. Приведен пример анализа тестовой конфигурации пяти объектов-справочников.

Использование разработанной методики и инструментария анализа конфигураций 1С позволяет получить наглядное представление структуры конфигураций 1С:Предприятие в виде иерархий объектов, локализовать дублирование объектов, имеющих одинаковую

структуру, а также получить программу представления объектов на другом объектно-ориентированном языке, поддерживающем наследование свойств.

Благодарности. Работа поддержана Российским фондом фундаментальных исследований, гранты 17-07-01341, 18-07-00758, 17-47-380007.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. 1С:Предприятие 8. Система программ [Электронный ресурс] // URL: http://v8.1c.ru/predpriyatie/questions_licence.htm (дата обращения: 15.05.19)
2. IBM Developer [Электронный ресурс] // URL: https://www.ibm.com/developerworks/ru/library/x-xjavaforum5/index.html (дата обращения: 15.05.19)
3. Национальный открытый университет ИНТУИТ [Электронный ресурс] // URL: https://www.intuit.ru/studies/professional_skill_improvements/1210/courses/6/lecture/182?page=2 (дата обращения: 15.05.19)
4. Сообщество Программистов. Исходный код наших знаний [Электронный ресурс] // URL: https://python-scripts.com/object-oriented-programming-in-python (дата обращения: 15.05.19)
5. Сообщество Программистов. Исходный код наших знаний [Электронный ресурс] // URL: https://python-scripts.com/sqlite (дата обращения 15.05.19)

УДК 004.94

РАЗРАБОТКА МОДУЛЯ ПРОВЕДЕНИЯ ВИРТУАЛЬНЫХ АУКЦИОНОВ ДЛЯ СТЕНДА СТ-ИЭС ООО «ПОЛЮС-НТ»

Чекан М.А.
(г. Иркутск, Иркутский национальный исследовательский технический университет)
e-mail: chekoopa@mail.ru

Аннотация. Статья посвящена разработке и реализации программного модуля для виртуального проведения аукционов в составе тренировочного стенда «СТ-ИЭС» производства ООО «Полюс-НТ». В статье рассматриваются детали анализа предметной области, использованные инструменты и отдельные аспекты касательно реализации, необходимые для получения автоматизированного клиент-серверного игрового сервиса с веб-интерфейсом и интеграцией в общий программно-аппаратный комплекс.

Ключевые слова. Автоматизация, аукционы, умная энергетика.

DEVELOPMENT OF A VIRTUAL AUCTIONS MODULE FOR «ST-IES» STAND OF POLUS-NT

Chekan M.A.
(Irkutsk, Irkutsk National Research Technical University)
e-mail: chekoopa@mail.ru

Abstract. The article is devoted to the development and implementation of a software module for the virtual auctions as part of a training stand «ST-IES» developed by Polus-NT. The article discusses the details of the analysis of the domain, the tools used and particular aspects regarding the implementation necessary to obtain an automated client-server game service with a web interface and integration into the whole computing appliance.

Keywords. Automation, auctions, smart energy systems.

Введение. Стенд-тренажёр «Интеллектуальные энергетические системы» (далее – энергостенд) – автоматизированный аппаратно-программный комплекс, предназначенный для моделирования работы энергосистемы в игровой форме и развития навыков планирования и управления реальной сетью энергоснабжения [1]. Энергостенд выступает одновременно в роли макета энергосистемы и автономной измерительной системы.

Процесс работы на стенде подразделяется на несколько взаимосвязанных этапов (микроигр), одной из которых является розыгрыш объектов потребления и генерации на аукционе. Аукцион проводится до основной игры с целью определить принадлежность объектов командам-участникам, а также тарифы для участвующих в игре объектов. Закрытый формат аукциона добавляет в игру элемент человеческой психологии, но в то же время затрудняет работу ведущего по сбору ставок и подведению итогов. Имеет место человеческий фактор, замедляющий игру и обуславливающий возможные ошибки в процессе проведения. Помимо этого, проведение аукциона ведущим требует использование стороннего реквизита (бумага и ручки, приложения-мессенджеры), что также усложняет процесс дополнительными зависимостями.

Таким образом, обуславливается актуальность и практическая значимость данной работы, целью которой является проектирование и программная реализация программного модуля проведения виртуальных аукционов в рамках программно-аппаратного комплекса «СТ-ИЭС». Цель формализуется через следующие задачи:

- составить техническое задание на проектирование и разработку;
- провести анализ предметной области, функциональное моделирование, выделение автоматизируемых операций;
- проанализировать средства разработки, удовлетворяющие требованиям технического задания;
- спроектировать и разработать составляющие программного модуля, пользовательский интерфейс и серверную часть, а также объектную модель и структуру базы данных, и интерфейс взаимодействия;
- произвести проектирование и реализацию программного обеспечения с помощью выбранных средств и технологий согласно этапам разработки, подготовить к интеграции в программно-аппаратный комплекс.

Анализ моделируемой игры. Аукцион является составной «микроигрой» в процессе эксплуатации энергостенда и проходит в формате закрытого аукциона первой цены [2]. Имеются лоты, представляющие собой игровые объекты-здания с индивидуальным адресом. Участники делают ставки на заявленный лот и передают их ведущему, не раскрывая конкурентам. Ведущий сравнивает ставки и подводит итог текущего круга: закрытие лота с передачей его победителю или повторный аукцион на иных условиях. Сравнение ставок

производится по возрастанию или убыванию в зависимости от типа лота. Повторный аукцион объявляется в случае, если две или более ставки различаются меньше, чем на 0,5 у.е. Владельцы остальных ставок от участия в повторном аукционе отстраняются, для остальных ставок, кроме лидирующей, ставится дополнительное ограничение на минимальную ставку (сдвиг на 0,1 у.е.). Если спорные ставки установлены на предельное значение, разыгрывается all-pay аукцион, ставки на котором списываются со счёта участника вне зависимости от исхода. После первого круга участники могут отказаться от одного из своих лотов, выставив их на второй круг аукциона вместе с теми, что не были разыграны на первом. При этом сам участник на «сброшенный» лот не претендует. По завершении аукциона результаты (владение и тарифы) переносятся в основную игру (сборка и моделирование энергосети). По описанной выше модели игры составлена диаграмма состояний, изображённая на рисунке 1.

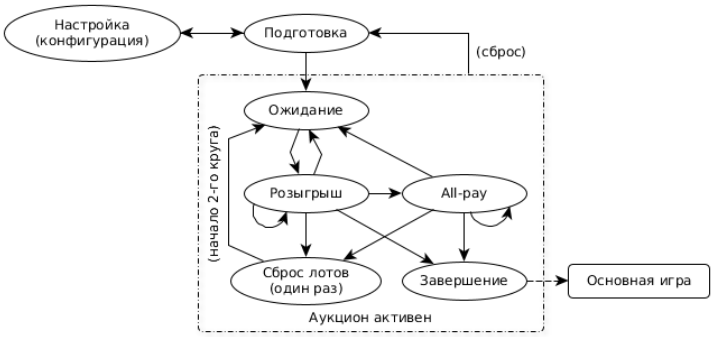


Рис. 1. Модель состояний аукциона

Учитывая правила, можно определить две роли: ведущий и участник (игрок). При этом косвенное влияние на изменение состояния участники имеют только в пределах состояний «Розыгрыш» и «All-pay», в остальных случаях переход закономерен согласно правилам (ограничение по времени, объявление лота, завершение игры) либо выполняется по воле ведущего (старт и сброс игры, досрочное завершение или объявление лота).

Проектирование и программная реализация. При первом приближении можно точно установить, что конечный программный модуль является клиент-серверным приложением. Серверная часть устанавливается на центральном сервере энергостенда и принимает запросы с клиентских машин (терминалов) игроков. Для реализации была выбрана архитектура Web-сервиса с REST-интерфейсом как наиболее удовлетворяющая требованиям технического задания, в том числе по однородности используемых технологий. Согласно архитектуре, приложение подразделяется на серверную часть (backend) и интерфейс пользователя (frontend).

Типовые интеракции (взаимодействия «игрок-ведущий») с учётом правил и предполагаемого пользовательского опыта (UX) были спроецированы на модель состояний аукциона (рисунок 2) с указанием условий изменения состояния игры. Ряд условий является вложенным (зависит от события «Таймер лота закончился…»). При этом помимо взаимодействия с системой пользователь должен получать с серверной части обновление в виде актуального состояния игры, детализированного согласно его роли (например, игрок не знает чужих ставок, в отличие от администратора).

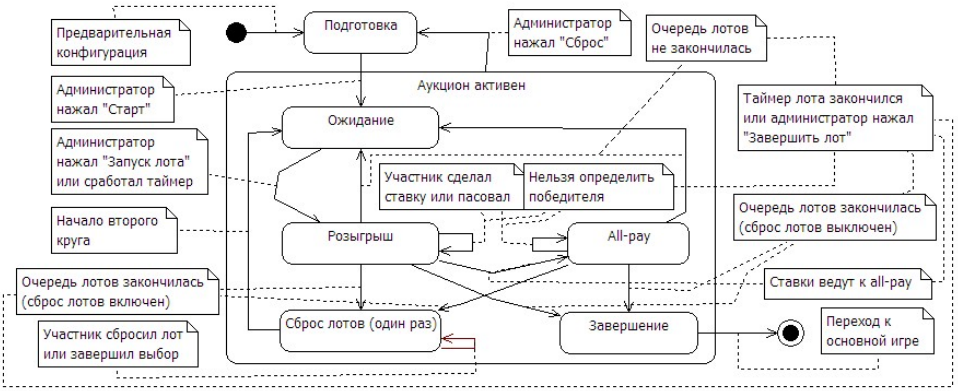


Рис. 2. Диаграмма состояний аукциона с интеракциями

Из этого следует необходимость использования многопоточного сервера с возможность запуска в отдельном потоке управляющей программы для контроля таймера и изменения состояний игры. В качестве инструмента реализации серверной части был выбран язык программирования Python с библиотекой асинхронного веб-сервера aiohttp. Данные инструменты позволили с наименьшими затратами времени реализовать необходимую логику. Необходимое разграничение прав между игроками и администратором достигается реализованной авторизацией через ключи (заданные администратором или случайно сгенерированные) с привязанными к ним ролями и передачей ключа сессии в заголовке HTTP-запроса.

С учётом правил и необходимых для отображения данных построена объектная модель данных в серверной части (рисунок 3). Объект «Состояние» (кроме поля «Настройки») с применёнными ограничениями на детализацию передаётся клиенту на запрос об обновлении.

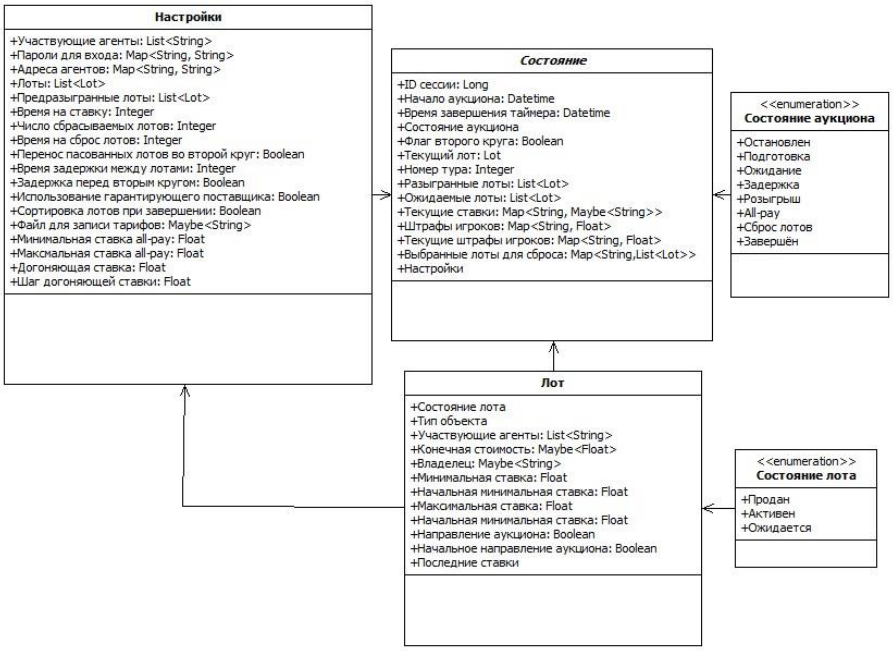


Рис. 3. Объектная модель интерфейса взаимодействия

Для подстройки параметров аукциона (разыгрываемые лоты, время на ставку, величина различия ставок и т.д.) применяется конфигурация в формате YAML, загружаемая

из файла или HTTP-запроса. Её содержимое формирует объект «Настройки». По завершении аукциона формируется файл с тарифами на разыгранные лоты в формате YAML и схемой, согласованной с главным программным модулем энергостенда. Если это указано конфигурацией, тарифы автоматически записываются в соответствующий файл.

Интерфейс пользователя реализован в форме одностраничного веб-приложения, связанного с серверной частью REST-запросами. Доступ к нему осуществляется через веб-браузер или специальную веб-оболочку. Это позволяет разместить веб-сервер с интерфейсом в любой точке сети энергостенда. В качестве языка программирования для реализации интерфейса выбран Elm [3], компилируемый в JavaScript функциональный язык программирования, реализующий архитектуру Model-View-Controller и декларативное описание интерфейсов в функционально-реактивном стиле. Это позволило получить стабильную и легко изменяемую реализацию интерфейса с отображениями, меняющимися в зависимости от состояния. Так, на рисунке 4 представлен интерфейс пользователя для первого игрока (в углу рядом с кнопкой выхода имеется индикация активного пользователя). В верхней части приведён полный перечень лотов, разыгранных и ожидающих розыгрыш. Внизу помещена карточка текущего аукциона с информацией о лоте, статусе заявок других игроков и таймером завершения данного тура. Напротив карточки размещается панель действий, для игрока она представляет собой форму ввода заявки с текстовым полем и кнопкой для пасования, для администратора на панели в зависимости от состояния отображаются блоки с кнопками для управления игрой.

Рис. 4. Скриншот интерфейса пользователя (для первого игрока)

Так как разрешение экрана на терминалах стенда фиксировано, вся отрисовка выполняется SVG-тегами по координатам, что избавляет разработчиков от реализации адаптивной вёрстки и ускоряет разработку и рефакторинг интерфейсной части. Исключение составляют списки (например, перечень лотов), реализованные через встраивание HTML-контейнера (тег div, вложенный в тег foreignObject). Для упрощения отладки предусмотрен специально разработанный модуль для глобального масштабирования.

Заключение. Разработан и реализован программный модуль для виртуального проведения аукционов на рынке электроэнергетики. Модуль автоматизирует часть деятельности администратора комплекса, снимая с него нагрузку по проведению аукциона и переносу его результатов в основную игру, а также оптимизирует процесс эксплуатации

стенда-тренажёра, сокращая время на проведение аукциона. Полученное программное обеспечение успешно применяется в реальных условиях при эксплуатации стенда СТ-ИЭС на различных мероприятиях (образовательный интенсив Остров 10-21, финал Олимпиады НТИ 2019 года), опыт применения показывает большую динамику работы на стенде, лёгкость использования, а также отладки и добавления функциональности благодаря успешному подбору инструментов.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. Полюс-НТ [Электронный ресурс] // URL: http://polyus-nt.ru/ (дата обращения: 01.10.2018)

2. Правила игры на стенде «Интеллектуальные энергетические системы» версии 2.3 [Электронный ресурс] // URL: https://goo.gl/xcT2Ju (дата обращения: 01.10.2018)

3. Elm - A delightful language for reliable webapps. [Электронный ресурс] //URL: https://elm-lang.org/ (дата обращения: 26.09.2018)

Авторский указатель

Андреева А.М. Комплексные сети в стилометрии русских литературных текстов – 32
Аршинский В.Л. Программно-инструментальная среда «ГрЛАО» для оценки эффективности работы организации – 37

Барахтенко Е.А. Архитектура веб-приложения для исследований интегрированных систем энергетики – 79
Бахвалова З.А. Метод формализации требований к входным данным – 44
Белей Е.Г. Применение алгоритмов решения проблемы булевой выполнимости к составлению оптимального расписания работы композиции вычислительных сервисов – 7
Белых А.А. Разработка системы политематической классификации на базе алгоритмов машинного обучения – 12

Засядко А.А. «Паттерн автоматизации» поддержки решения организационно-распорядительных задач – 85

Калугин М.В. Поиск матрицы расстояний с использованием ГИС систем – 74
Камышова Е.А. Метод формализации требований к входным данным – 44
Кондратьев В.С. Применение дубликатов конфликтных ограничений для ускорения работы алгоритмов решения проблемы булевой выполнимости – 19
Копылов М.М. Разработка и исследование цифрового генератора случайных последовательностей – 53
Копылова В.В. Разработка инструментального средства для анализа конфигураций 1С:Предприятие – 92
Куклин Е.В. Особенности реализации вероятностно-событийного поведения агентов-эвакуирующихся в комбинированной модели развития событий при пожаре – 57
Куликова Л.Л. Разработка виртуального музея вычислительной техники ИРНИТУ на основе технологии дополненной реальности – 63

Лукьянов Н.Д. «Паттерн автоматизации» поддержки решения организационно-распорядительных задач – 85

Маланова Т.В. Кластеризация точек евклидовой плоскости с ограничением по весу – 25

Орехов Д.О. Архитектура веб-приложения для исследований интегрированных систем энергетики – 79

Сиротинина Е.А. Кластеризация точек евклидовой плоскости с ограничением по весу – 25

Труфанов А.И. Комплексные сети в стилометрии русских литературных текстов – 32

Худякова Ю.Н. Разработка виртуального музея вычислительной техники ИРНИТУ на основе технологии дополненной реальности – 63
Хуен Ван Лэ Программно-инструментальная среда «ГрЛАО» для оценки эффективности работы организации – 37

Чекан М.А. Разработка модуля проведения виртуальных аукционов для тренировочного стенда СТ-ИЭС ООО «Полюс-НТ» – 100
Черкашин Е.А. Разработка инструментального средства для анализа конфигураций 1С:Предприятие – 92

Шишкин П.И. Моделирование отражательных характеристик космического аппарата EGP в оптическом диапазоне – 69

Научное издание

ВИНЕРОВСКИЕ ЧТЕНИЯ 2019

Материалы Всероссийской молодежной
научно-практической конференции

(Иркутск, 6–7 июня 2019 г.)

Электронная версия
Заказ ЭИ-8, поз. плана д/п
Усл. печ. листов 6,75