

Международный журнал  
информационных технологий  
и энергоэффективности |



Том 4 Номер 3(13)



2019



## СОДЕРЖАНИЕ / CONTENT

### ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ

- 
- |    |  |           |
|----|--|-----------|
| 1. | <b>Якушева К.И.</b> Способ извлечения утверждений из неформализованного текста на основе онтологий | <b>3</b>  |
|    | <b>Yakusheva K.I.</b> A Method of Extracting Statements from Unstructured Text Using Ontology      |           |
| 2. | <b>Папок А.В.</b> Приложение дополненной реальности для иллюстрации детской книги                  | <b>10</b> |
|    | <b>Rapok A.V.</b> An Augmented Reality Application for Illustration of a Baby Book                 |           |
| 3. | <b>Лебедев М.М.</b> Взаимодействие языков программирования   | <b>19</b> |
|    | <b>Lebedev M.M.</b> Programming Languages Interaction  |           |
| 4. | <b>Савостин Н.В.</b> Технология Blockchain, краткий обзор  | <b>26</b> |
|    | <b>Savostin N. V.</b> Blockchain Technology, Short Overview  |           |
| 5. | <b>Дударовская О.Г.</b> Энергоэффективность массопереноса в насадочных аппаратах                   | <b>34</b> |
|    | <b>Dudarovskaya O.G.</b> Energy Efficiency of Mass Transfer in Nozzles                             |           |
-



Международный журнал информационных технологий и энергоэффективности

Сайт журнала:

<http://www.openaccessscience.ru/index.php/ijcse/>



УДК 004.912

## СПОСОБ ИЗВЛЕЧЕНИЯ УТВЕРЖДЕНИЙ ИЗ НЕФОРМАЛИЗОВАННОГО ТЕКСТА НА ОСНОВЕ ОНТОЛОГИЙ

**Якушева К.И.**

*Филиал федерального государственного бюджетного образовательного учреждения высшего образования «Национальный исследовательский университет МЭИ» в г. Смоленске, Россия (214013, г. Смоленск, Энергетический проезд, дом 1); e-mail: [www.roksana@mail.ru](mailto:www.roksana@mail.ru)*

В данной статье предлагается способ извлечения утверждений на основе онтологии для повышения качества извлечения. Суть способа состоит в применении онтологии предметной области на этапе извлечения отношений. Способ извлечения утверждений базируется на подходе по правилам. Для решения задачи извлечения помимо онтологий применяются лингвистические правила и шаблоны. Способ включает в себя такие этапы анализа текста как: графематический, морфологический, синтаксический. Также в статье представлена методика, позволяющая оценить качество извлечения утверждений из неформализованного текста по двум основным показателям: точности и полноте извлечения. Для объединения этих показателей предложено использовать F-меру. Методика оценки эффективности основана на официальных метриках РОМИП (Российский семинар по оценке методов информационного поиска).

Ключевые слова: извлечение утверждений, методика оценки эффективности, оценка точности и полноты, извлечение утверждений на основе онтологий.

## A METHOD OF EXTRACTING STATEMENTS FROM UNSTRUCTURED TEXT USING ONTOLOGY

**Yakusheva K.I.**

*Smolensk Branch of Federal state budgetary educational institution of higher education "National research University Moscow power engineering Institute", Russia (214013, Smolensk, Energeticheski proezd, 1); e-mail: [www.roksana@mail.ru](mailto:www.roksana@mail.ru)*

This article proposes a method of extracting assertions based on ontology to improve the quality of extraction. The essence of the method consists in applying the ontology of the subject area at the stage of extracting relationships. The method of extracting assertions is based on a rule-based approach. In addition to ontologies, linguistic rules and patterns are used to solve the extraction problem. The method includes the following stages of text analysis. The article also presents a method for assessing the quality of extracting statements from a non-formalized text using two main indicators: accuracy and completeness of extraction. To combine these indicators, it is proposed to use the F-measure. The methodology for evaluating the effectiveness is based on the official ROMIP metrics (Russian seminar on the evaluation of information retrieval methods).

Keywords: extracting assertions, methods for evaluating effectiveness, evaluating accuracy and completeness, extracting assertions based on ontologies.

Задачи извлечения информации из текстов на сегодняшний день являются достаточно важными в компьютерной лингвистике. Можно выделить как подзадачу извлечение утверждений. Эта подзадача актуальна в тех случаях, когда есть потребность извлекать информацию, к примеру, для пополнения онтологий. Исходя из существующих подходов к извлечению информации и принимая во внимание необходимость извлечения именно утверждений в определенной тематике, можно сделать вывод о том, что наиболее подходящим является подход, основанный на правилах. Также учитывая все особенности этого подхода, можно утверждать, что для достижения более высокой точности и полноты извлечения оптимально использовать сочетание таких средств как лингвистические шаблоны и онтология предметной области. Это позволит компенсировать невозможность предусмотреть все варианты шаблонов и те случаи, когда шаблоны оказываются неэффективны.

Поэтому ставится цель:

- разработать способ и реализующий его алгоритм извлечения утверждений из неформализованного текста на основе онтологий, который будет более эффективен, чем классический подход, основанный на правилах;
- разработать методику оценки эффективности способа извлечения утверждений.

### **Способ извлечения утверждений из неформализованного текста на основе онтологий**

К разрабатываемому способу предъявляются следующие требования:

- предлагаемый способ должен учитывать все этапы обработки текста: графематический, морфологический, синтаксический;
- разрабатываемый способ должен учитывать тематику предметной области (например, книжный каталог/магазин);
- способ извлечения утверждений должен предназначаться для текстов русского языка;
- разрабатываемый способ должен предполагать использование онтологии предметной области (книжный каталог/магазин);
- лингвистические шаблоны в способе извлечения утверждений должны основываться на заданной тематике и учитывать максимально возможное количество вариантов.

За основу способа берется метод извлечения фактов предложенный в [1]. Изменения относятся к 4 этапу, в который добавлен новый алгоритм, также изменен 5 и 6 этап, так как способы различаются извлекаемой информацией. Основным отличием можно назвать использование онтологии предметной области для извлечения отношений.

Предлагаемый способ извлечения утверждений из неформализованного текста состоит из следующих основных этапов.

*Этап 1.* Разбиение текста на абзацы и предложения.

*Этап 2.* Разбор каждого предложения на простые элементы: слова, пробелы, цифры, кавычки, знаки препинания и т.д.

*Этап 3.* Выявления сущностей, содержащихся в предложении: предикатов, объектов, предложенных групп, численных значений, дат и т.п.

*Этап 4.* Выявления связей между сущностями (по фактам или шаблонам).

*Этап 5.* Отбор отношений для формирования утверждений.

*Этап 6.* Проверка наличия неопределенности в указанных утверждениях.

Этап 7. Вывод полученных утверждений.

Далее более подробно раскрываются основные этапы.

**Этап 1** Графематический анализ текста. На данном этапе с текстом производятся следующие действия.

- разбиение текста на графемы;
- выделение в исходном тексте абзацев, заголовков, примечаний;
- определение границ предложений.

На данном этапе могут возникнуть две связанные задачи:

- 1) определение, терминального знака препинания (под терминальными знаками понимается точка, восклицательный и вопросительный знаки) как границы предложения в данном контексте,
- 2) определение всех границ предложений в документе.

На данном этапе часто используются словари сокращений, которые помогают частично решить задачу определения конца предложения.

Выходные данные этого этапа являются входными для следующего и представляют размеченный текст. Возможно также табличное представление, где каждый элемент таблицы – это отдельное предложение.

**Этап 2** Предполагает разделение текста на еще более мелкие части: слова, пробелы, знаки препинания, цифры, скобки и т.д. Этот этап также, как и предыдущий можно отнести к графематическому анализу.

Выходные данные для этого этапа - таблица с отдельными словами, знаками препинания, цифрами и т.д.

Этот этап заканчивает подготовку текста к извлечению сущностей.

**Этап 3** Осуществляет выявление сущностей. Первоначально необходимо определять имена собственные, числовые значения, названия в кавычках и т.п. После этого удаляем лишние пробелы. После чего происходит определение предикатов объектов (чаще всего глагол (причастие, деепричастие), а также предложных групп и дата/время.

На этом этапе используются дополнительные словари, например, словарь имен, фамилий, чисел и т.п.

Ключевым элементом этапа 3 разработанного метода извлечения утверждений являются правила поиска сущностей.

Предложено описывать каждое правило в виде функции определенного вида.

Функции необходимые к реализации разработанного метода:

1. функция для определения соответствия ли входного элемента правилу отбора сущностей;
2. функция поиска во множестве входных данных хотя бы одного элемента, удовлетворяющего правилу отбора сущностей;
3. функция перебора во множестве элементов входных данных, при этом одновременно все элементы должны соответствовать правилу отбора сущностей.

Для каждой из этих функций должны быть заданы три параметра: последовательность операций над данными для проверки соответствия, минимальное число совпадений и максимальное число совпадений. Для выполнения операций над данными каждая функция может вызывать другие функции.

**Этап 4** Извлечение отношений между сущностями.

На данном этапе помимо правил, как было в изначальном методе, подключается онтологический подход. Система ищет совпадение слов из одного предложения и сравнивает с утверждениями из онтологии.

Важно, чтобы онтология была максимально наполнена и была релевантной наборам текстов, из которых будут извлекаться утверждения.

Шаблоны и правила, также помогают установить связь между сущностями, извлеченными в предыдущем этапе.

Основные сущности языка в обрабатываемых в текстах [2]:

- предикат (predicate) – множество элементов, которое состоит из глагола (причастие, деепричастие) или краткого прилагательного и дополняющих слов (наречий);
- дата/время (date/time) – множество элементов, обозначающих дату и (или) время;
- объект (object) – множество элементов, обозначающее одну сущность (явление, процесс) реального мира;
- предложная группа (prepositionalgroup) – множество элементов, которое состоит из предлога и объекта (объектов).

Основные отношения [3]:

- Объект – Предикат – Объект (ObjectPredicateObject) – множество элементов, обозначающее взаимодействие между несколькими объектами;
- Объект – Предикат – Свойство (ObjectPredicateProperty) – множество элементов, обозначающее свойство объекта.

**Этап 5** Отбор отношений должен учитывать следующие критерии:

- 1) наличие важного ключевого предикативного слова (глагол, причастие, деепричастие, краткое прилагательное);
- 2) наличие важного ключевого слова-существительного;
- 3) наличие трех элементов в составе утверждения.

При соблюдении всех трех условий утверждение можно считать сформированным.

**Этап 6** разработанного способа при наличии одинаковых утверждений, полученных из различных источников информации – проверка наличия неопределенности в указанных утверждениях. Возможен вариант полного совпадения утверждений, в этом случае один из вариантов отбрасывается автоматически.

**Этап 7** Полученное множество утверждений группируется и представляется пользователю в виде отчетного документа. Возможно формирование текстового документа или вывод его на экран в рамках программного интерфейса.

Извлечение отношений происходит с помощью обращения к онтологии по средствам запросов на языке SparQL [4].

### **Методика оценки эффективности**

Оценка эффективности системы извлечения утверждений базируется на официальных метриках РОМИП (Российский семинар по оценке методов информационного поиска) [5].

Для оценки качества работы системы применяются разные оценки, которые основываются на анализе результатов работы системы. При этом "самым лучшим" алгоритмом можно считать тот, для которого выводы, сделанные системой, совпадают с мнением экспертов.

Большинство метрик, применяемых в современной оценке текстового поиска, опирается на отношении релевантности (принадлежности) найденного утверждения эталонному.

Метрики для неупорядоченного множества утверждений основаны на бинарной классификации документов «релевантен/не релевантен» по отношению к выбранному утверждению. Эти метрики основываются на матрице классификации (таблица 1), которая применяется в задачах поиска, извлечения фактов, событий и т.п. Применение метрики к утверждениям приемлемо, так как они сходны с фактами и включают в себя необходимые критерии оценки.

Таблица 1. Основные категории документов ответа системы

	Релевантны	Не релевантны
Найдено системой	a	b
Не найдено системой	c	d

Здесь, a — количество утверждений, найденных системой и релевантных с точки зрения экспертов; b — количество утверждений, найденных системой, но не релевантных с точки зрения экспертов; c — количество релевантных утверждений, не найденных системой; d — количество нерелевантных утверждений, не найденных системой.

Основными критериями, интересующим нас, являются точность и полнота.

#### **Полнота (recall)**

Полнота (recall) вычисляется как отношение найденных релевантных утверждений к общему количеству релевантных утверждений:

$$r = \frac{a}{a+c} \quad (1)$$

Полноту можно охарактеризовать как способность системы находить нужные пользователю утверждения, но она не учитывает количество нерелевантных утверждений, выдаваемых пользователю. Например, если полнота равна 50%, то это значит, что половина релевантных утверждений системой не найдена.

#### **Точность (precision)**

Точность (precision) вычисляется как отношение найденных релевантных утверждений к общему количеству найденных утверждений:

$$p = \frac{a}{a+b} \quad (2)$$

Точность можно охарактеризовать как способность системы выдавать в списке результатов только релевантные утверждения. Например, если точность равна 50%, то это значит, что среди найденных утверждений половина релевантных и половина – нерелевантных.

Иногда имеет смысл совместить эти две характеристики в одну, в таком случае можно использовать F-меру.

#### **F-мера (F-measure)**

F-мера часто используется как единая метрика, объединяющая метрики полноты и точности в одну метрику. F-мера для данного случая вычисляется по формуле:

$$F = \frac{2}{\frac{1}{p} + \frac{1}{r}} \quad (3)$$

Отметим основные свойства метрики F:

- $0 \leq F \leq 1$
- Если  $p = 0$  или  $r = 0$ , то  $F = 0$
- Если  $p = r$ , то  $F = p = r$

$$\bullet \min(p, r) \leq F \leq \frac{p+r}{2}$$

Предлагается использовать данную оценку эффективности таким образом.

- 1) Система должна обработать набор текстов, состоящих не менее чем из десяти текстов, для того чтобы избежать случайных ошибок.
- 2) Тексты предоставляются группе экспертов-оценщиков, они определяют эталонные утверждения, согласно предметной области.
- 3) Затем результаты экспертов-оценщиков сравниваются с результатами системы и делятся на 3 группы, согласно матрице классификации:
  - найдены системой совпадают с эталонными утверждениями;
  - не найдены системой эталонные утверждения;
  - найдены системой ошибочно.
- 4) Подсчитав количественные показатели из предыдущего пункта находятся показатели полноты и точности.
- 5) Вычисляется F-мера, которая будет использоваться для оценки различных подходов.
- 6) Вычисляется относительное приращение показателей качества.

Для тестирования рекомендуется подбирать тексты таким образом, чтобы они не содержали опечаток и грамматических и пунктуационных ошибок, или предварительно проверять их, это повысит точность извлечения.

### Оценка эффективности

Разработанные программные средства, основанные на предлагаемом способе, тестировались на наборе данных состоящих из 10 текстов по тематике предметной области. В качестве входных данных берутся рецензии пользователей книжного сервиса LiveLib [6]. Они представлены в виде текстового файла. Текст рецензий не структурирован, все авторские ошибки и опечатки сохраняются.

В таблице 1 приведены результаты оценки способа извлечения утверждений на основе онтологий со способом без использования онтологического подхода.

Таблица 1 – сравнение оценок эффективности.

	С использованием онтологического подхода	Без использования онтологического подхода
Точность (p)	0,83	0,75
Полнота (r)	0,65	0,44
F-мера	0,73	0,56

Таким образом показатели точности извлечения при использовании онтологии в среднем больше на 8%, а показатель полноты на 21%, если сравнивать общую эффективность извлечения утверждений из неформализованного текста на основе онтологий, то данный способ дают прирост эффективности в 17% по сравнению со способом, построенным на базе правил и лингвистических шаблонов.

В рамках статьи предложен способ извлечения утверждений на основе онтологий. Данный способ повышает качество извлекаемых утверждений из неформализованного текста в сравнении с подходом на правилах без использования онтологического подхода.

Также в статье приводится методика оценки эффективности извлечения утверждений и предлагается для оценки описанного способа и представлены результаты оценки предложенного способа.

### Список литературы

1. Grishman R., Information Extraction. In: The Handbook of Computational Linguistics and Natural Language Processing. A. Clark, C. Fox, and S. Lappin (Eds), Wiley-Blackwell, 2010, pp. 515-530.
2. Большакова Е.И., Воронцов К.В., Ефремова Н.Э., Клышинский Э.С., Лукашевич Н.В., Сапин А.С. Автоматическая обработка текстов на естественном языке и анализ данных. – М.: НИУ ВШЭ, 2017, 296 с.
3. Sekine's Extended Named Entity Hierarchy. URL: <http://nlp.cs.nyu.edu/ene/>. (дата обращения 21.05.2019).
4. Feldman R., Sanger J. (ed.). The text mining handbook: advanced approaches in analyzing unstructured data. — Cambridge University Press, 2007.
5. Gruber, T. R. A translation approach to portable ontology specification. // KnowledgeAcquisition.1993. Vol. 5. № 1. Pp. 199–220.
6. Башмаков А.И., Башмаков И.А. Интеллектуальные информационные технологии. – Москва, Издательство МГТУ имени Н.Э. Баумана, 2005. – 304 с.

### References

1. Grishman R., Information Extraction. In: The Handbook of Computational Linguistics and Natural Language Processing. A. Clark, C. Fox, and S. Lappin (Eds), Wiley-Blackwell, 2010, pp. 515-530.
  2. Bolshakova E.I., Vorontsov K.V., Efremova N.E., Klyshinsky E.S., Lukashevich N.V., Sapin A.S. Automatic processing of natural language texts and data analysis. - M.: HSE, 2017, 296 p.
  3. Sekine's Extended Named Entity Hierarchy. URL: <http://nlp.cs.nyu.edu/ene/>. (referral date of May 21, 2019).
  4. Feldman R., Sanger J. (ed.). The text mining handbook: advanced analysis in analyzing unstructured data. - Cambridge University Press, 2007.
  5. Gruber, T. R. A translation approach to portable ontology specification. // KnowledgeAcquisition.1993. Vol. 5. No. 1. Pp. 199-220.
  6. Bashmakov A.I., Bashmakov I.A. Intellectual information technologies. - Moscow, Publishing House of Moscow State Technical University named after NE Bauman, 2005. - 304 p.
-



Международный журнал информационных технологий и энергоэффективности

Сайт журнала:

<http://www.openaccessscience.ru/index.php/ijcse/>



УДК 004.42

## ПРИЛОЖЕНИЕ ДОПОЛНЕННОЙ РЕАЛЬНОСТИ ДЛЯ ИЛЛЮСТРАЦИИ ДЕТСКОЙ КНИГИ

**Папок А.В.**

*ФГБОУ ВО «НИУ «МЭИ», Москва, Россия (111250, г. Москва, ул. Красноказарменная, д.14), e-mail: Helena.97@mail.ru*

**В статье описывается пример приложения дополненной реальности с множественным выбором для иллюстрации детской книги по основам электроэнергетики. Описывается разработка приложения в редакторе Unity и его связь с Vuforia SDK. Представлена разработка 3D-модели. А также разработка интерфейса.**

Ключевые слова: электроэнергетика, AR-приложение, множественный выбор, Unity, Vuforia, иллюстрация книги.

## AN AUGMENTED REALITY APPLICATION FOR ILLUSTRATION OF A BABY BOOK

**Рарок А.В.**

*NRU «MPEI», Moscow, Russia (111250, Moscow, street Krasnokazarmennaya, 14), e-mail: Helena.97@mail.ru*

**The paper describes an example of an augmented reality application with multiple choices to illustrate a children's book. on the basics of electric power. The book in the form of a game introduces children to the electric power industry. It describes how to develop an application in the Unity editor and its connection to the Vuforia SDK. Presents the development of 3D-models. As well as the development of the interface.**

Keywords: extracting assertions, methods for evaluating effectiveness, evaluating accuracy and completeness, extracting assertions based on ontologies.

Дополненная реальность (англ. augmented reality, AR) [1] — это технология, которая имитирует искусственные объекты в реальной среде. В дополненной реальности используются датчики и алгоритмы для определения положения и ориентации камеры. Затем AR технология визуализирует 3D-графику, звуки или видео так, как видит камера, и накладывает сгенерированное изображения на пользовательский вид реального мира.

В настоящее время большую популярность набирает использование AR технологии в издательской сфере. Широкое применение книг с дополненной реальностью пришло со стороны детских книг и энциклопедий.

Целью разработанного приложения с использованием технологии дополненной реальности является иллюстрация детской книги. Книга рассчитана для детей дошкольного возраста

и направлена на знакомство с электроэнергетикой в игровой форме. Первая ассоциация, возникающая при упоминании данной сферы, — это конечно же электричество. Ярким примером электроприбора является люстра. Задача приложения состоит в том, чтобы правильно посчитать количество перегоревших лампочек.

Приложение должно:

- быть совместимо с устройствами под ОС Android;
- распознать метку и «поднять» 3D-модель люстры, с произвольным количеством перегоревших ламп;
- вывести сообщение, в зависимости от нажатой кнопки;
- позволить, перезапустить приложение;
- позволить поворот люстры для лучшего обзора, при касании экрана.

Первым шагом разработки, было создание 3D-модели люстры в 3D-редакторе Blender [2]. Для достоверности, лампы люстры моделировались по их реальному изображению, что представлено на рисунке 1. Полная же модель люстры представлены на рисунке 2.



Рисунок 1 - Полная 3D-модель лампы

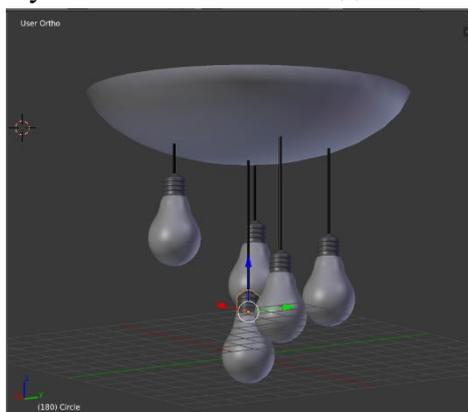


Рисунок 2 - 3D-модель люстры

Следующий этап разработки – создание метки. В качестве изображения была выбрана картинка люстры из интернета, которая была отредактирована под 3D-модель (рисунок 3).



Рисунок 3 – Метка люстры

Для использования этой метки в AR-приложении, создается база данных с метками в Vuforia, туда и загружается наше изображение. После загрузки изображения Vuforia самостоятельно определяет точки распознавания и присваивает рейтинг изображения, который отображается звездами. Чем ниже рейтинг, тем хуже будет распознаваться изображение. Метке люстры «LMP» присвоили 4 звезды, значит приложению будет легко распознать метку.

Для работы с Unity необходимо для начала активизировать и настроить Vuforia и подключить базу данных с метками [3-4].

На сцену добавляем объект ARCamera, у которой будет Vuforia configuration (рисунок 4).

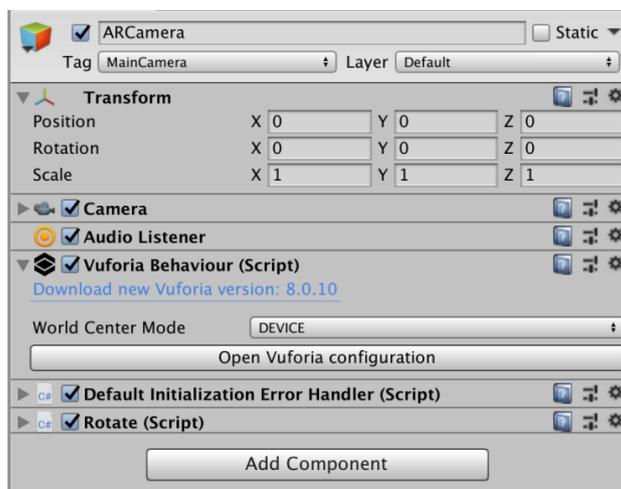


Рисунок 4 - Vuforia configuration

В Vuforia configuration производится подключение БД и вводятся лицензионный ключ (рисунок 5).

После на сцену добавляется изображение (ImageTarget), у которого имеется компонент Image Target Behavior. В этом компоненте необходимо подключить базу данных (рисунок 6).

На сцену добавляется 3D-объект люстры, как дочерний объект ImageTarget. Добавленную модель редактируем для корректного восприятия пользователем этого объекта, при его распознавании (рисунок 7). У 3D-объекта люстры, имеется несколько компонентов, которые представляют из себя коды: скрипт «Kol», обрабатывающий нажатие кнопки и сопоставление нажатой кнопки и правильного ответа, с выводом соответствующего сообщения; скрипт Rotate, осуществляющий поворот люстры относительно оси Oy.

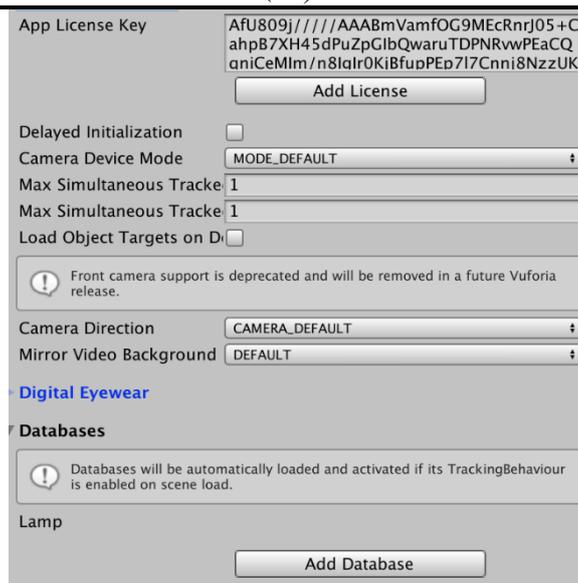


Рисунок 5 - Настройки Vuforia

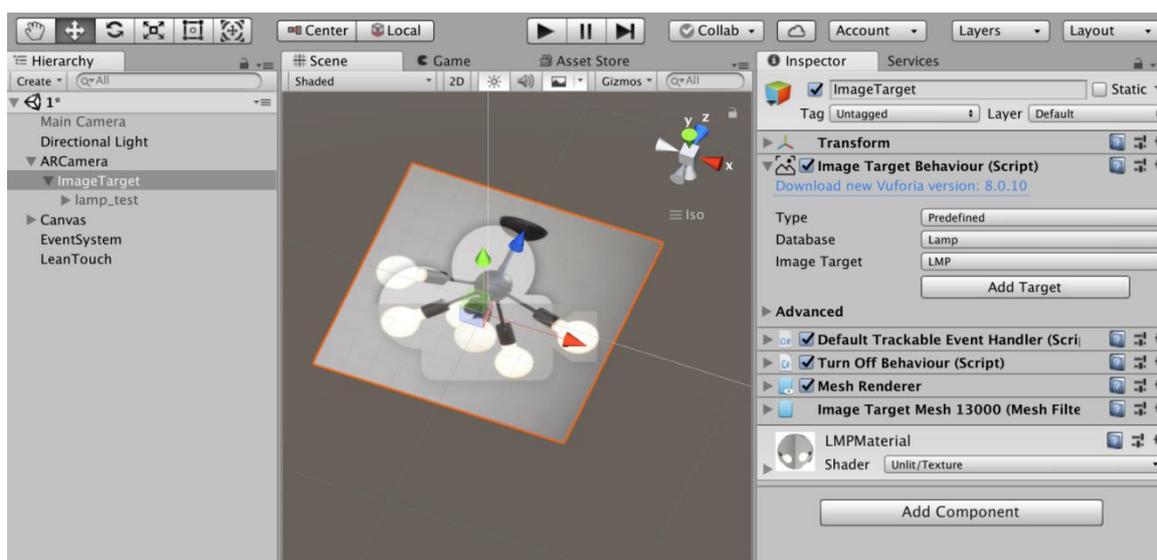


Рисунок 6 - Настройки ImageTarget

Затем добавляется корневой элемент интерфейса Canvas, который соответствует размеру экрана устройства, для которого разрабатывается приложение (рисунок 8).

На объект Canvas добавляются кнопки, которые располагаются в нижней части Canvas, с номерами от 0 до 5. Эти числа соответствуют возможному количеству перегоревших ламп. У кнопок есть компонент Anchor Presets, в нем привязываем кнопки к середине нижней части экрана. Таким образом, при изменении разрешения экрана или ориентации, кнопки сохранят свои позиции (рисунок 9).

В приложении возможны правильные и неправильные ответы, правильно ли посчитал количество перегоревших ламп пользователь или нет. Для этого создаются надписи правильно (correct) и неправильно (incorrect), привязанные к центру интерфейса (рисунок 10).

Так же для пользователя добавляется еще одна кнопка - New Game. Она нужна для того, чтобы пользователь каждый раз не запускал приложение, а просто перезагрузил его. К этой кнопке в компоненте Button в событие On Click() добавляется код перезагрузки сцены (рисунок 11).

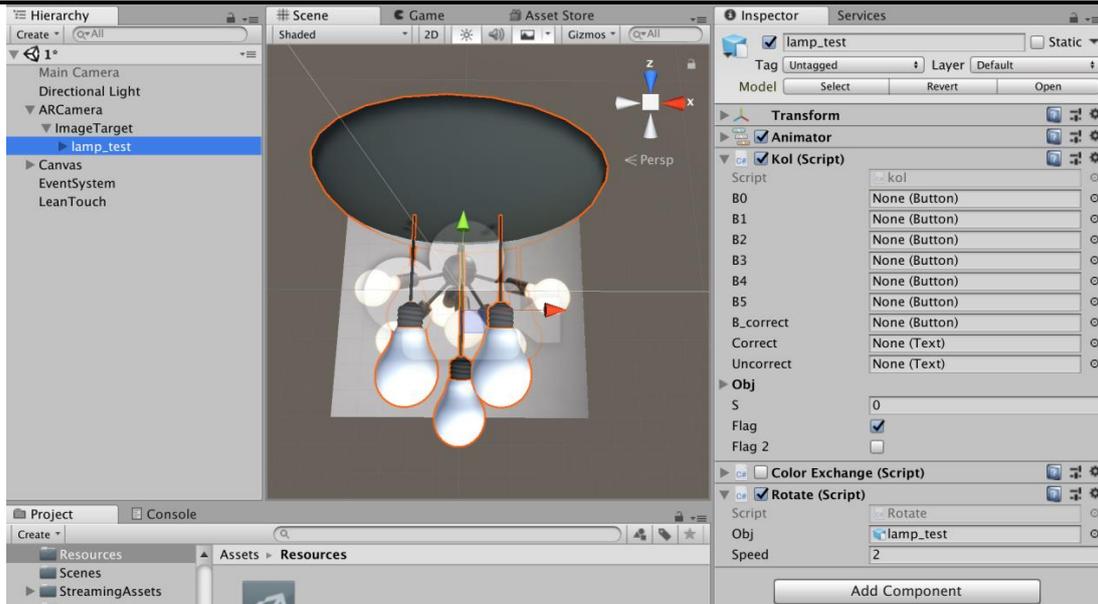


Рисунок 7 - 3D-модель люстры

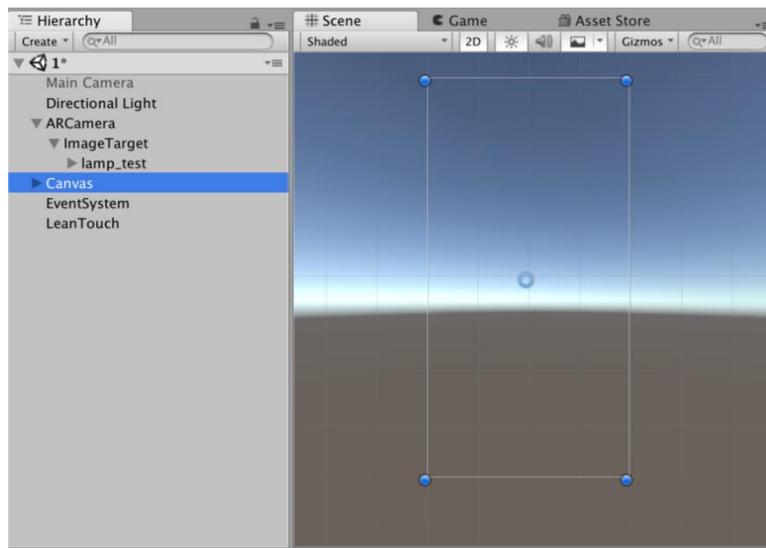


Рисунок 8 - Элемент интерфейса Canvas

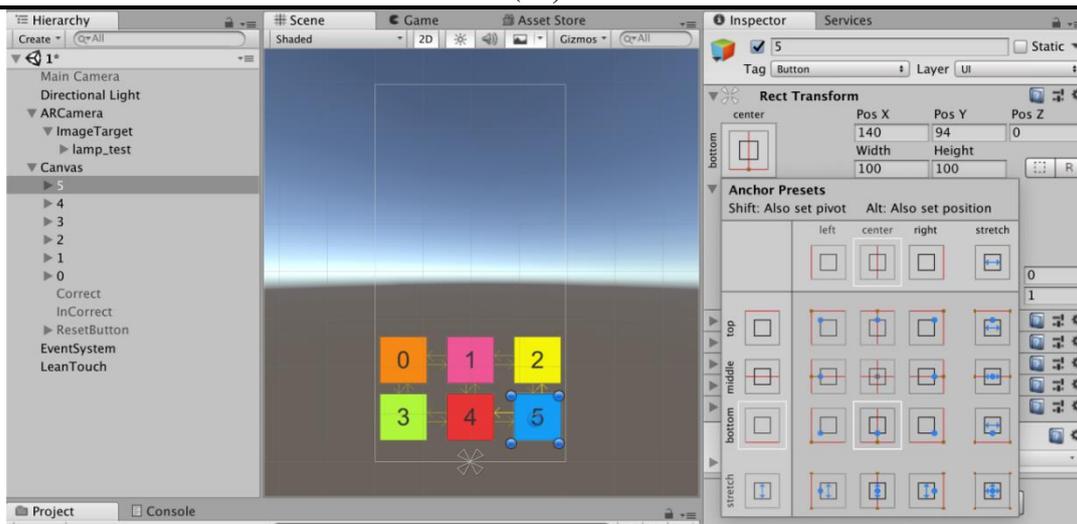


Рисунок 9 - Кнопки выбора варианта ответа

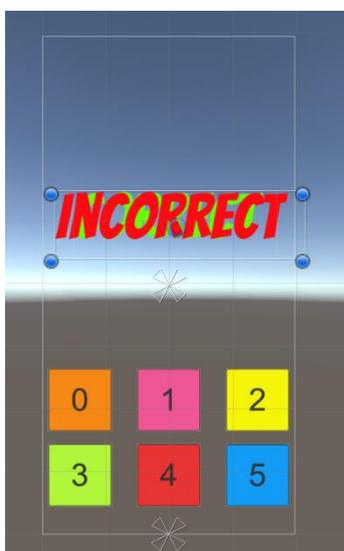


Рисунок 10 - Сообщения о корректности ответа

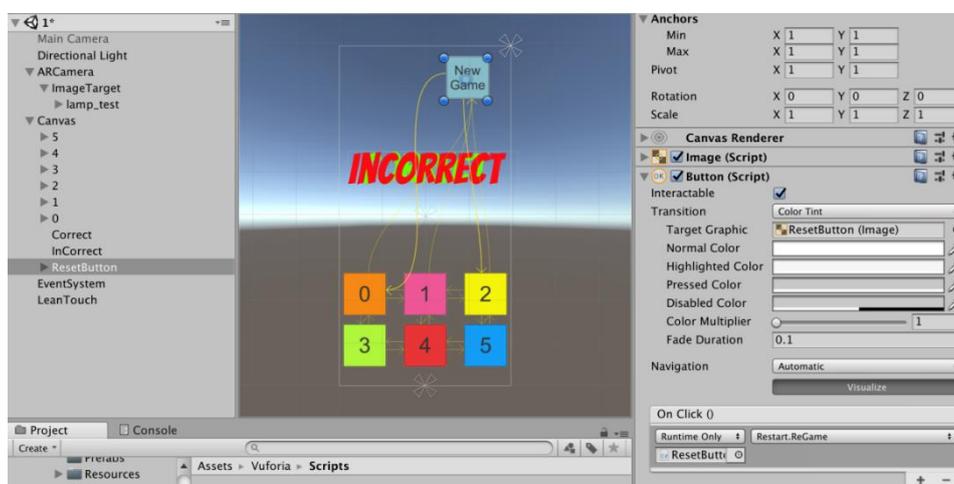


Рисунок 11 - Кнопка перезагрузки приложения

Интерфейс приложения представлен на рисунке 12.

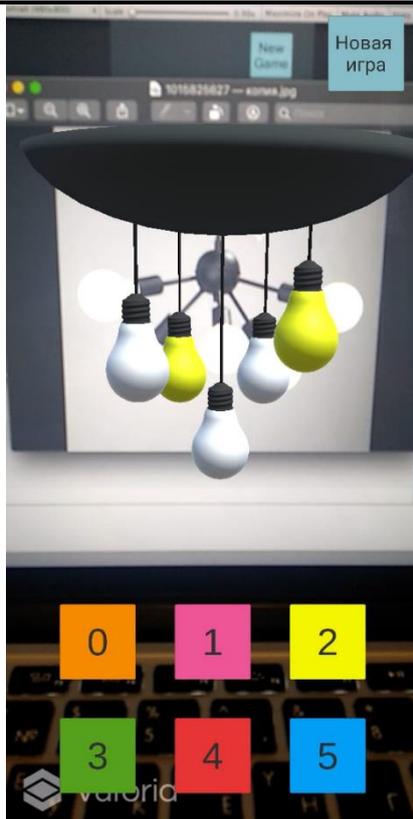


Рисунок 12 – Интерфейс приложения

На рисунке 13 представлена схема алгоритма работы пользователя с приложением.

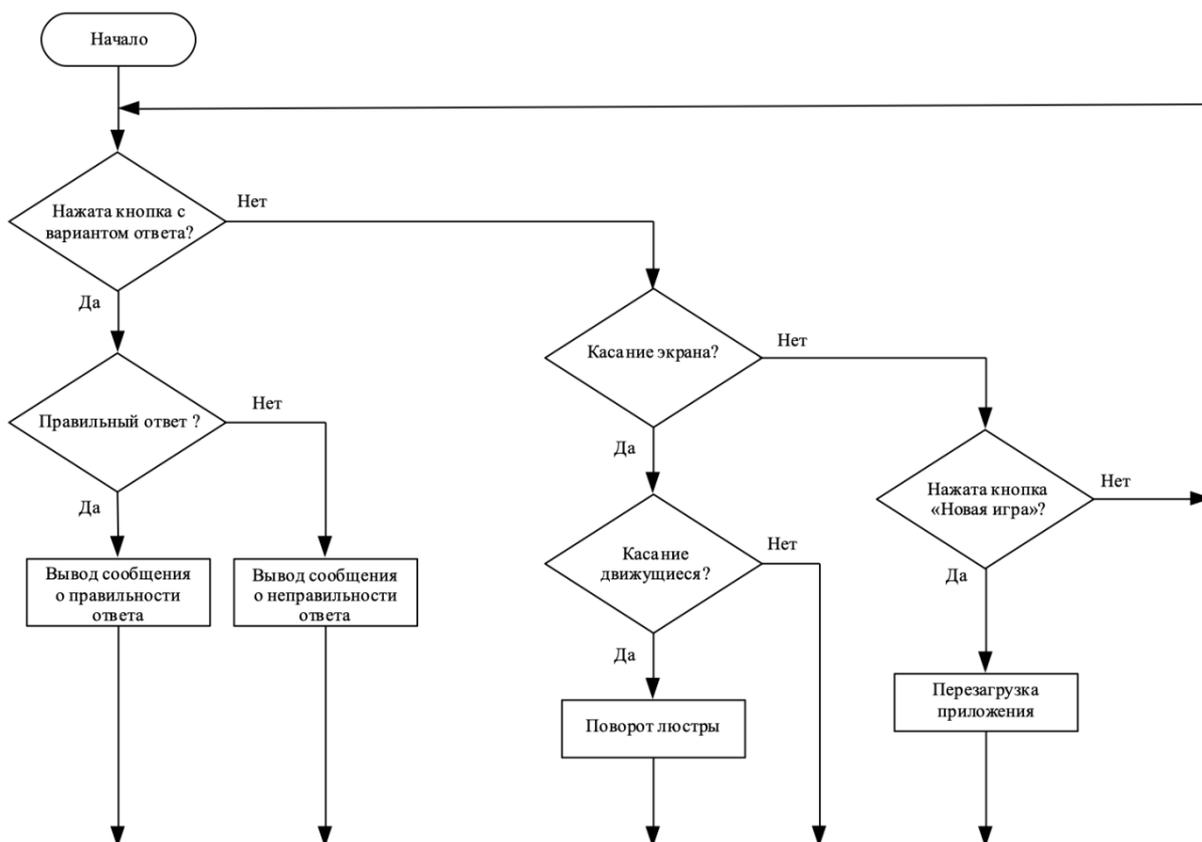


Рисунок 13 – Схема алгоритма работы пользователя с приложением

В ходе работы все скрипты были написаны на C#, по этой причине было необходимо подключать определённые директивы для работы с Unity. В таблице 1 представлены все директивы [3], которые использовались при создании приложения.

Таблица 1

Директивы	Функции
System.Collections	Содержит в себе интерфейсы и классы, которые необходимы для двумерных массивов, списков. Подключается автоматически при создании скрипта.
UnityEngine	Пространство имен (коллекция классов) для получения доступа ко всем методам (функциям) и переменным в классе Unity. Подключается автоматически при создании скрипта.
UnityEngine.UI	Необходимо для получения доступа ко всем компонентам UI. Это использовалось для вывода сообщений и создания кнопок.
UnityEngine.SceneManagement	Необходимо для управления сценами. Использовалось для перезагрузки приложения.

Все созданные скрипты наследуются от базового класса MonoBehaviour [3]. Он содержит в себе такие функции, как Start() и Update(). Первая функция используется для управления всеми объектами сцены, вторая – для модификаций объектов на сцене и самой сцены.

Тестирование приложения производилось функциональным и модульными подходами. Проведенное тестирование продемонстрировало четкую работу приложения, быстрое обнаружение метки (изображение люстры) и появление 3D-модели люстры, корректность множественного выбора с выводом соответствующих сообщений. Тестирование показало, что основное влияние на качество распознавания метки оказывает освещение и характеристики камеры смартфона.

Таким образом, разработанное AR-приложение с использованием современной технологии дополненной реальности позволяет в игровой форме привлечь юное поколение к сфере электроэнергетики, что и является его целью.

Визуализация общеизвестной модели люстры, довольно простого задания и яркого интерфейса позволяет использовать данное приложения для развития детей дошкольного возраста.

Дальнейшее развитие данной работы заключается в интеграции с другими AR-приложениями.

## Список литературы

1. Что такое дополненная реальность и как она работает? [Электронный ресурс] / Pioneer Design Blog – Режим доступа: <https://blog.pioneers.com.ua/2018/09/ar-vr/>, свободный. (Дата обращения: 27.05.2019 г.)
2. Blender [Электронный ресурс] / Blender – Режим доступа: <https://www.blender.org/>, свободный. (Дата обращения: 15.11.2018 г.)
3. Руководство Unity [Электронный ресурс] / Unity Documentation – Режим доступа: <https://docs.unity3d.com>, свободный. (Дата обращения: 28.10.2018 г.)
4. Vuforia [Электронный ресурс] / Vuforia engine developer portal – Режим доступа: <https://developer.vuforia.com/>, свободный. (Дата обращения: 28.10.2018 г.)

## References

1. What is augmented reality and how does it work? , Pioneer Design Blog. [Online]. Available: <https://blog.pioneers.com.ua/2018/09/ar-vr/> [Accessed: 27-May-2019]
  2. Blender, Blender. [Online]. Available: <https://www.blender.org/> [Accessed: 15-Nov-2018]
  3. Unity Manual, Unity Documentation. [Online]. Available: <https://docs.unity3d.com> [Accessed: 28-Oct-2018]
  4. Vuforia, Vuforia engine developer portal. [Online]. Available :<https://developer.vuforia.com/> [Accessed: 28-Oct-2018]
-



Международный журнал информационных технологий и энергоэффективности

Сайт журнала:

<http://www.openaccessscience.ru/index.php/ijcse/>



УДК 004.43

## ВЗАИМОДЕЙСТВИЕ ЯЗЫКОВ ПРОГРАММИРОВАНИЯ

**Лебедев М.М.**

ФГБОУ ВО «НИУ «МЭИ», Москва, Россия (111250, г. Москва, ул. Красноказарменная, д.14),  
e-mail: leb.dev96@gmail.com

---

**В статье описываются возможные варианты взаимодействия нескольких языков программирования, производится их подробный разбор. Рассматриваются положительные стороны их совместного использования. Также говорится о разделении языков на компилируемые и интерпретируемые, и что понимают под промежуточными в этой классификации.**

Ключевые слова: мультиязыковая разработка, взаимодействие языков, JVM, .NET CLR, COM, компиляция и интерпретация.

## PROGRAMMING LANGUAGES INTERACTION

**Lebedev M.M.**

*NRU «MPEI», Moscow, Russia (111250, Moscow, street Krasnokazarmennaya, 14), e-mail: leb.dev96@gmail.com*

---

**The paper describes in details different variants of multiple programming languages interaction. Their advantages and disadvantages are observed. What is more, in this article it is told about the subdivision of programming languages on compliable and interpretable ones and what are those which are between.**

Keywords: Multilanguage programming, programming languages interaction, JVM, .NET CLR, COM, compilation and interpretation.

Для каждого языка существует своя наиболее подходящая область применения. Зачастую для разработчиков это означает, что знания одного конкретного языка программирования недостаточно, однако освоить большое их количество в совершенстве – большая проблема. Многие проекты сейчас не живут каким-то одним языком, то есть, одна из частей существует на одном языке, другая – на другом. Это очень удобно, хотя бы потому, что начинающему программисту легче определиться с выбором языка, а компаниям, участвующим в разработке найти нужных специалистов. Данная статья рассматривает, как различные языки взаимодействуют друг с другом.

Различные языки могут сосуществовать в рамках одного или нескольких процессов. Взаимодействие в рамках нескольких процессов проще: нет никакой разницы, на каком языке данные созданы, и какой их читает. Как пример можно привести пару клиент – сервер [1]. Различные процессы могут легко обмениваться данными через файл или базу данных, для этого им достаточно иметь «договорённость» о том, в каком формате и куда будут предоставляться эти данные. Также межпроцессные взаимодействия могут использовать один из следующих механизмов:

- обмен сообщениями;
- синхронизация;
- разделение памяти;
- удалённые вызовы (Remote Procedure Call).

Самым быстрым средством среди перечисленных является использование разделяемой памяти [2]. В остальных обмен происходит через ядро, что приводит к переключению контекста и снижению производительности. Для использования данного средства выделяется сегмент памяти общий для процессов, и оба процесса получают ссылку на этот участок памяти.

Удалённый вызов процедур (Remote Procedure Call) - это класс технологий, который позволяет компьютерным программам вызывать функции или процедуры в другом адресном пространстве – например, на некотором удалённом хосте. Как правило, для реализации RPC требуется общий сетевой протокол для обмена данными и оговорённый язык сериализации объектов / структур [3]. На рисунке 1 представлена схема взаимодействия процессов через удалённый вызов процедур, под «стабом» понимается фрагмент кода, который занимается конвертацией параметров перед дальнейшей передачей ядру или одному из процессов.

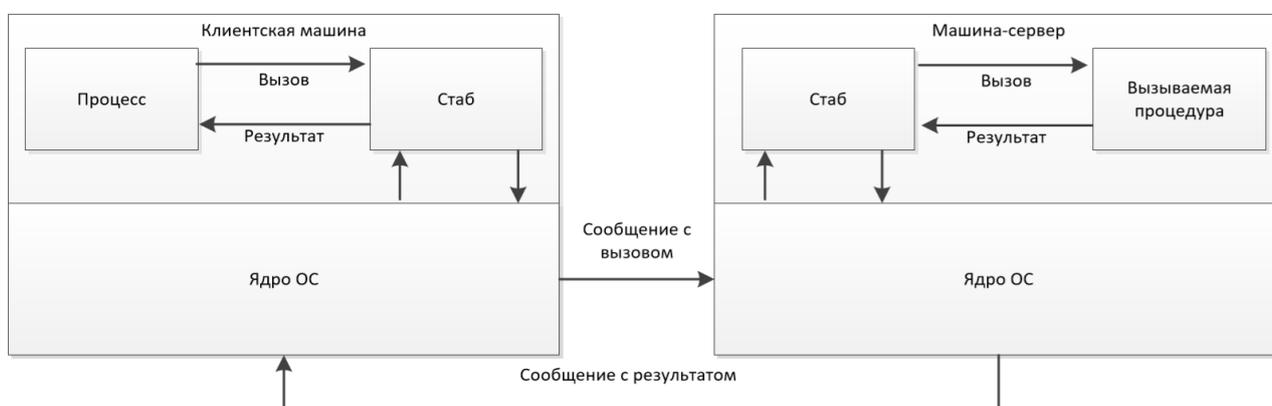


Рисунок 1 – Схема взаимодействия через удаленный вызов процедур

Как пример обмена сообщениями, можно привести технологию обмена данными через почтовые ящики [4], данная технология доступна на компьютерах под операционной системой Windows. Почтовый ящик – это локальный псевдофайл, хранящийся в памяти. Такие ящики пропускают сообщения в одном направлении. Процесс, создавший свой почтовый ящик, является его сервером, а остальные процессы являются, в данном случае, клиентами почтового ящика. Клиенты могут посылать серверу сообщения, записывая их в почтовый ящик. Сообщение может быть послано на почтовый ящик того же компьютера, компьютера в сети или сразу на все ящики с одинаковым именем на некотором домене. Возможность посылки

широковещательных сообщений – одна из главных особенностей именно технологии почтовых ящиков. Пришедшие сообщения дописываются и хранятся до «прочтения» их сервером. Для обеспечения двунаправленности двум процессам необходимо завести два собственных ящика. Другим более известным примером являются каналы – простое средство для взаимодействия процессов систем на базе ОС UNIX. Различают именованные и неименованные каналы. Неименованные представляют собой сущность, в которую можно помещать и извлекать данные через два файловых дескриптора (для чтения и записи соответственно) [5]. Именованные каналы представляют собой файлы отдельного типа, доступ к которым любой из процессов может получить в любое время, используя имя этого канала [6].

Когда речь идёт о взаимодействии в рамках одного процесса - функциям и подпрограммам нужно уметь вызывать друг друга, для чего и вводят стандарты вызова. Для языков семейства C и Pascal обычно применяется стандарт бинарных соглашений. Другим примером стандартов являются СОМ-объекты, которые могут быть написаны на многих языках.

Component Object Model — это технологический стандарт от компании Microsoft, закрепившийся в основном на ОС семейства Windows. Он предназначен для создания ПО на основе взаимодействующих компонентов объекта. Каждый из объектов может быть одновременно использован несколькими процессами [7]. Программы, построенные на стандарте, фактически представляют собой набор взаимодействующих между собой СОМ-компонентов. Взаимодействие происходит через СОМ-интерфейсы, по умолчанию каждый компонент должен, как минимум, переопределять интерфейс «IUnknown».

Как уже говорилось выше, процессы могут вызывать функции на других языках, обмениваясь результатами их выполнения. Соглашение о вызове - это описание технических особенностей вызова подпрограмм, определяющее:

- способы передачи параметров подпрограммам (их порядок помещения в стек);
- способы вызова (передачи управления);
- способы передачи результатов обратно в точку вызова;
- способы возврата (передачи управления).

Рассмотрим особенности некоторых из применяемых соглашений [8]:

- Cdecl – стандарт языков C. Передача аргументов - справа налево. Вызывающая программа должна сама очищать стек, сохранять, а затем восстанавливать значения регистров.
- Pascal – используется компиляторами языка Паскаль. Аргументы передаются через стек, слева направо. Результат возвращается через параметр Result.
- Stdcall или Winapi — применяется в ОС Windows для вызова функций WinAPI. Аргументы передаются через стек, справа налево. Вызываемая подпрограмма должна очищать стек.
- Fastcall — самый быстрый способ, так как аргументы передаются через регистры (если их недостаточно, то используют стек). Вызов не стандартизирован.
- Safecall — соглашение, используемое для вызова методов интерфейсов СОМ. Методы представляют собой функции, возвращающие результат типа HRESULT, который затем должен быть проанализирован.
- Thiscall – используется компиляторами C++ при вызове методов классов. Почти не отличается от Cdecl.

Стоит отдельно отметить ещё один способ взаимодействия языков в рамках нескольких процессов – когда языки компилируются в общий промежуточный код. Например, языки .NET и языки для виртуальной машины Java «собираются» в единый исходный код. Поэтому объекты, созданные на Java доступны, например, на Scala – доступность определяется на уровне метаданных виртуальной машины. Чтобы подробнее рассмотреть данный тип взаимодействия необходимо вспомнить об условном делении языков на компилируемые и интерпретируемые. [9].

Программы на компилируемых языках собираются программой-компилятором в исполняемый модуль, содержащий набор инструкций для некоторого типа процессора. Для выполнения кода на интерпретируемом языке требуется программа-посредник (интерпретатор): она выполняет исходный код программы без перевода. На рисунке 2 представлена схема, отображающая принципиальное отличие компиляции от интерпретации.

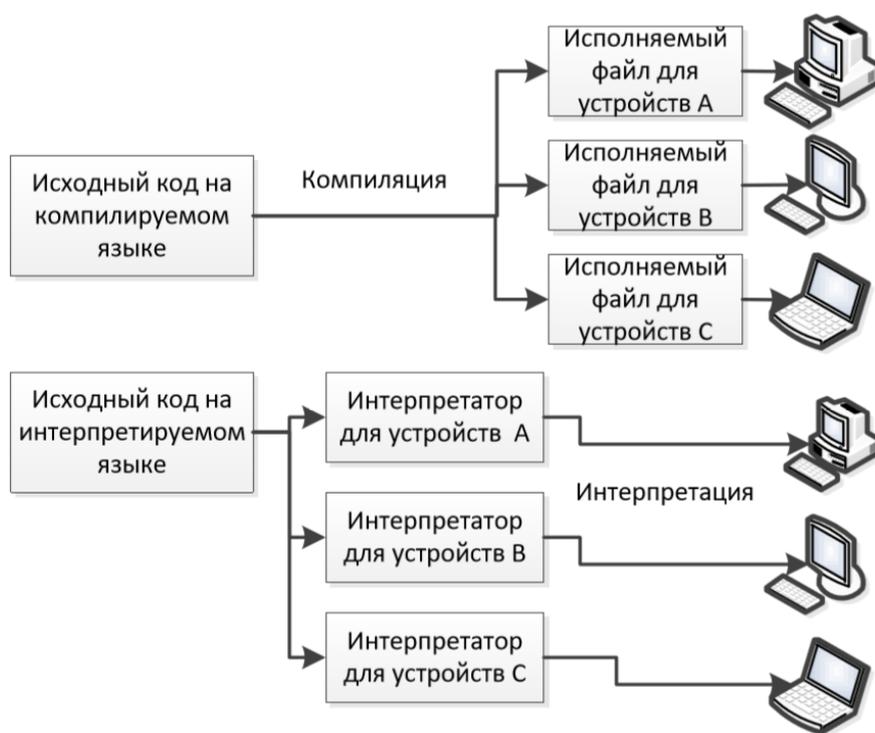


Рисунок 2 – Компиляция и интерпретация программ

JVM (Java Virtual Machine) – или виртуальная машина Java, исполняет собственный байт код, который собирается Java компиляторами. В Java код могут быть собраны программы на Scala, Ada и т. д. JVM можно установить практически на любое устройство. То есть программа, созданная и скомпилированная на одном устройстве на языке семейства Java, может быть выполнена практически где угодно. [10]

Подобно JVM, .NET CLR представляет собой виртуальную машину, интерпретирующую байтовый код. Эти байтовые коды отличаются, однако машины стремятся предоставлять схожие функции. То, что у JVM называется «Java байт-кодом» у

.NET CLR называют инструкциями промежуточного языка «Intermediate Language» (IL). [11] Обе виртуальные машины имеют возможность компилировать свой входной байт-код на машинный язык компьютера, на котором они запущены прямо во время выполнения – это

называется «Just In Time Compilation» (JIT) [12], а полученный выходной код называют JIT-кодом. Таким образом, языки семейств Java и .NET могут взаимодействовать на уровне своей виртуальной машины.

Скомпоновав собранную информацию, получим диаграмму классификации взаимодействий языков программирования. Данная диаграмма представлена на рисунке 3.

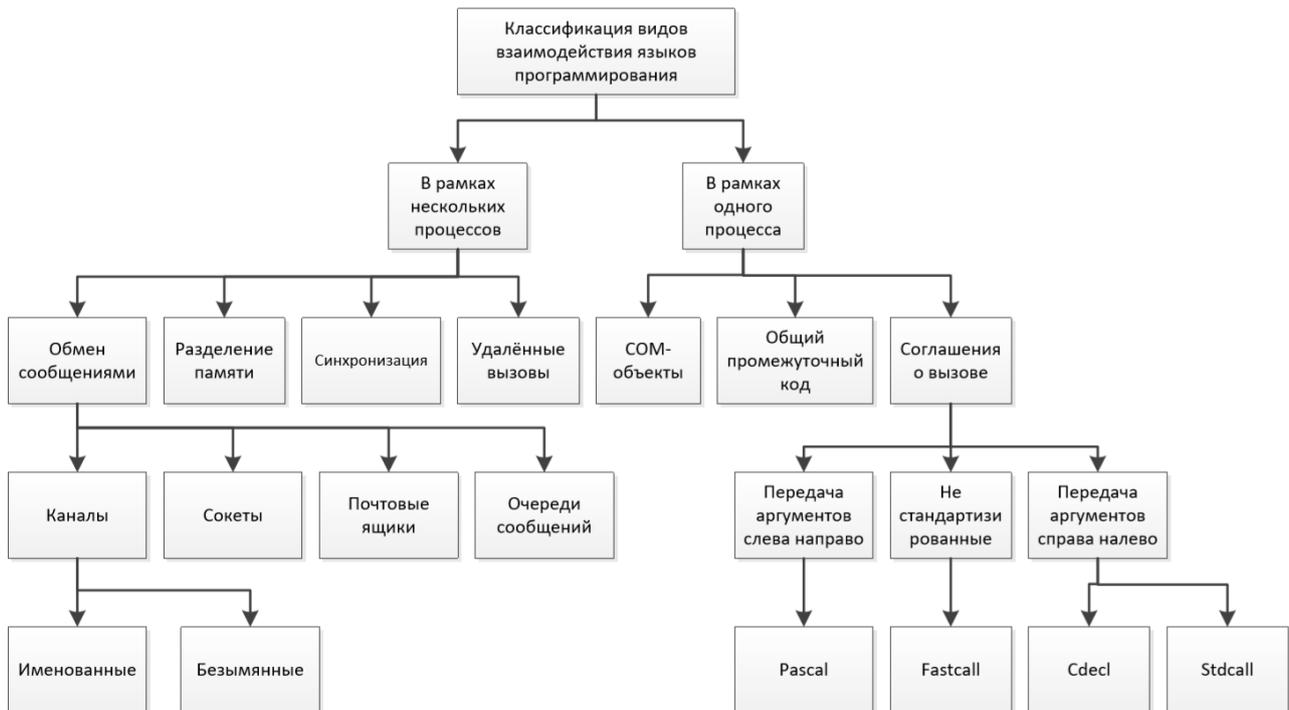


Рисунок 3 – Классификация взаимодействия языков программирования

Таким образом, вид взаимодействия в первую очередь определяется тем, в рамках скольких процессов происходит взаимодействие. Далее классификация определяет метод взаимодействия по способу передачи данных (и / или управления). Некоторые из способов, в свою очередь, также имеют подвиды, и их можно также классифицировать по характерным признакам.

### Список литературы

1. Анатольев А. Г., Конспект лекций по курсу «Сетевые технологии». Омск: ОмГТУ, 2013. 79 с.
2. Иртегов Д. В., Конспект лекций по курсу «Системы семейства UNIX». Новосибирск: НГУ, 2015. 146 с.
3. Межпроцессное взаимодействие (Операционные системы) [Электронный ресурс] / Национальная библиотека им. Н. Э. Баумана – Режим доступа: [https://ru.bmstu.wiki/Межпроцессное\\_взаимодействие\\_\(Операционные\\_Системы\)](https://ru.bmstu.wiki/Межпроцессное_взаимодействие_(Операционные_Системы)), свободный. (Дата обращения: 25.07.2019 г.).
4. Джонсон М. Харт, Системное программирование в среде Windows: пер. с англ. - М.: Издательский дом "Вильямс", 2005. - 592 с.

5. Именованный канал [Электронный ресурс] / Википедия – свободная энциклопедия – Режим доступа: [https://ru.wikipedia.org/wiki/Именованный\\_канал](https://ru.wikipedia.org/wiki/Именованный_канал), свободный. (Дата обращения: 25.07.2019 г.).
6. Неименованный канал [Электронный ресурс] / Википедия – свободная энциклопедия – Режим доступа: [https://ru.wikipedia.org/wiki/Неименованный\\_канал](https://ru.wikipedia.org/wiki/Неименованный_канал), свободный. (Дата обращения: 25.07.2019 г.).
7. Component Object Model [Электронный ресурс] / Википедия – свободная энциклопедия – Режим доступа: [https://ru.wikipedia.org/wiki/Component\\_Object\\_Model](https://ru.wikipedia.org/wiki/Component_Object_Model), свободный. (Дата обращения: 25.07.2019 г.).
8. Соглашение о вызове [Электронный ресурс] / Википедия – свободная энциклопедия – Режим доступа: [https://ru.wikipedia.org/wiki/Соглашение\\_о\\_вызове](https://ru.wikipedia.org/wiki/Соглашение_о_вызове), свободный. (Дата обращения: 25.07.2019 г.).
9. Суховилов Б. М., Конспект лекций по курсу «Информатика». Челябинск: ЮУрГУ, 2015. 59 с.
10. Что такое JVM? Знакомство с виртуальной машиной Java [Электронный ресурс] / Topjava.ru - курсы программирования на java – Режим доступа: <https://topjava.ru/blog/what-is-the-jvm>, свободный. (Дата обращения: 25.07.2019 г.).
11. Обзор среды CLR — .NET Framework | Microsoft Docs [Электронный ресурс] / Microsoft – официальная страница – Режим доступа: <https://docs.microsoft.com/ruru/dotnet/standard clr>, свободный. (Дата обращения: 25.07.2019 г.).
12. JIT для начинающих – devSchacht – Medium [Электронный ресурс] / Medium – a place to read and write big ideas and important stories – Режим доступа: <https://medium.com/devschacht/how-to-start-jitting-ee9fcbc9065a>, свободный. (Дата обращения: 25.07.2019 г.).

## References

1. Anatolyev A. G., Lecture notes on «Net technologies». Omsk: OmSTU, 2013. 79 p.
2. Irtegov D. V., Lecture notes on «UNIX family systems». Novosibirsk: NSU, 2015. 146 p.
3. Inter-process communication (operational systems) [Electronic resource] / National library of N. E. Bauman – Access mode: [https://ru.bmstu.wiki/Межпроцессное\\_взаимодействие\\_\(Операционные\\_Системы\)](https://ru.bmstu.wiki/Межпроцессное_взаимодействие_(Операционные_Системы)), free. (Request Date: 25.07.2019).
4. Johnson M. Hart, Windows system programming. 3-rd ed. The Addison-Wesley Microsoft Technology Series, 2005. 656 p.
5. Named pipe [Electronic resource] / Wikipedia, the free encyclopedia – Access mode: [https://en.wikipedia.org/wiki/Named\\_pipe](https://en.wikipedia.org/wiki/Named_pipe), free. (Request Date: 25.07.2019).
6. Anonymus pipe [Electronic resource] / Wikipedia, the free encyclopedia – Access mode: [https://en.wikipedia.org/wiki/Anonymous\\_pipe](https://en.wikipedia.org/wiki/Anonymous_pipe), free. (Request Date: 25.07.2019).
7. Component Object Model [Electronic resource] / Wikipedia, the free encyclopedia – Access mode: [https://ru.wikipedia.org/wiki/Component\\_Object\\_Model](https://ru.wikipedia.org/wiki/Component_Object_Model), free. (Request Date: 25.07.2019).
8. Calling convention [Electronic resource] / Wikipedia, the free encyclopedia – Access mode: [https://en.wikipedia.org/wiki/Calling\\_convention](https://en.wikipedia.org/wiki/Calling_convention), free. (Request Date: 25.07.2019).
9. Suhilov B. M., Lecture notes on «Informatics». Chelyabinsk: SUSU, 2015. 59 p.
10. What is JVM? Getting acquainted to Java [Electronic resource] / Topjava.ru – java programming courses – Access mode: <https://topjava.ru/blog/what-is-the-jvm>, free. (Request Date: 25.07.2019).

11. CLR overview — .NET Framework | Microsoft Docs [Electronic resource] / Microsoft – official page – Access mode: <https://docs.microsoft.com/ru-ru/dotnet/standard/clr>, free. (Request Date: 25.07.2019).
  12. JIT for beginners – devSchacht – Medium [Electronic resource] / Medium – a place to read and write big ideas and important stories – Access mode: <https://medium.com/devschacht/howto-start-jitting-ee9fcbc9065a>, free. (Request Date: 25.07.2019)
-



Международный журнал информационных технологий и энергоэффективности

Сайт журнала:

<http://www.openaccessscience.ru/index.php/ijcse/>



УДК 004.6

## В ТЕХНОЛОГИЯ BLOCKCHAIN, КРАТКИЙ ОБЗОР

**Савостин Н.В.**

*ФГБОУ ВО «НИУ «МЭИ», Москва, Россия (111250, г. Москва, ул. Красноказарменная, д.14), e-mail: nekit0939@yandex.ru*

---

**В статье объясняется функционирование технологии блокчейн и критически оценивается её потенциальная роль в совершенствовании услуг в банковской сфере, заключении договоров и системах баз данных.**

Ключевые слова: блокчейн, смарт-контракт, биткоин, криптовалюта.

## BLOCKCHAIN TECHNOLOGY, SHORT OVERVIEW

**Savostin N. V.**

*NRU «MPEI», Moscow, Russia (111250, Moscow, street Krasnokazarmennaya, 14), e-mail: nekit0939@yandex.ru*

---

**The paper explains the operation of the «blockchain technology» and critically evaluates its potential role in improving banking services, contracts and database systems.**

Keywords: blockchain, smart-contract, bitcoin, cryptocurrency.

### **I. Что такое технология блокчейн?**

Технология блокчейн – это изначально название, данное структуре, лежащей в основе работы цифровой валюты Биткоин. Создатель Биткоина никогда не использовал термин «блокчейн» в своей документации, и, читая эту документацию, складывается чёткое впечатление, что автор не вводил новую технологию в традиционном смысле этого слова, а разрабатывал программное обеспечение, основываясь на нескольких, уже существующих технологиях, позволяющих ему создать «одноранговую версию электронных денег» (Сатоши Накамото) [1].

Суть функционирования блокчейна в Биткоине заключается в том, что всякий раз, когда два участника сети совершают транзакцию, они заявляют о своей транзакции всем участникам сети (узлам), которые записывают транзакцию в блок ограниченной ёмкости. Как только блок заполняется,

узлы одновременно выполняют Proof-of-Work<sup>1</sup> – математические операции, которые трудно решить, но правильное решение которых легко проверить [2-7]. Эти математические операции не связаны с транзакциями биткоина, но они необходимы для функционирования системы, так как они заставляют проверяющие узлы расходовать вычислительную мощность, которая была бы потрачена впустую, если бы они включали какие-нибудь сфальсифицированные или недействительные транзакции. Первый узел, который справится с решением Proof-of-Work передаёт решение вместе с блоком транзакций остальным узлам. Узлы могут быстро и легко подтвердить точность транзакций и решений, и когда 51% вычислительной мощности сети подтверждает блок, узлы начинают записывать новые транзакции в новый блок, дополняя все предыдущие блоки.

Первый узел, который “решает” Proof-of-Work, получает некоторое количество валюты сети. Это вознаграждение делает проверку транзакций потенциально прибыльной и приводит к тому, что обычно называют «майнингом» (добычей), хотя «проверка», возможно, более функционально правильное описание этого процесса. Проверка Proof-of-Work блока намного дешевле, чем его корректное решение, что делает установление права владения валютой экономически эффективным и выгодным. Функционально – технология Блокчейн – это технология проверки: поскольку это гораздо более дорого выполнить подтверждение работы, чем проверить её правильность, правдивость – это единственная стратегия прибыльности для узлов, а результат – это запись, которая не оспаривается ни одним из членов сети.

Функционирование децентрализованной блокчейн-сети полностью зависит от решения Proof-of-Work и голосования за достоверность блоков узлами, занимающими процессор. Истинность транзакций подтверждается не какой-то уполномоченной службой, а соглашением участников сети с большинством вычислительных процессоров. С помощью такого механизма, Биткоин безошибочно записал более 140 миллионов транзакций в течение почти 8-ми лет.

К 2013-му году, некоторые проекты и компании продвигали идею использования технологии блокчейн без цифровой валюты [16]. В этих «блокчейнах с контролируемым доступом» только заранее подтверждённые участники могут записывать данные в блокчейн, который существует как распределённый регистр между всеми участниками. В них не производится расчёт Proof-of-Work, поскольку истинность транзакций основывается на том, что участники идентифицируемы и подконтрольны друг другу. На текущий момент не существует внедрённых в рамках коммерческих проектов блокчейн технологий, однако существуют детально описанные прототипы и проекты. В следующем разделе рассматриваются три наиболее распространённых применения технологии блокчейн для оценки их рентабельности.

## **II. Потенциальные способы применения технологии блокчейн**

Обзор стартапов и исследовательских проектов, связанных с технологией блокчейн показывает, что потенциальные способы применения блокчейна можно разделить на три основные области:

**а) Цифровые платежи** [11, 12, 13, 17, 18]: Текущие коммерческие алгоритмы для оплаты основаны на централизованных регистрах для записи всех транзакций и сохранения баланса счёта. По сути, транзакция передаётся один раз от участников сделки до посредника, проверяется на достоверность и оба счёта соответственно корректируются. В блокчейне транзакция рассылается всем узлам сети, что подразумевает намного больше пересылок, больше времени и большую производительность. Транзакция также становится частью цепочки блоков, скопированной на каждый компьютер, который является членом сети. Это медленнее и значительно дороже, чем централизованная оплата, и объясняет, почему

---

<sup>1</sup> Proof-of-Work – досл. доказательство выполнения работы - принцип защиты сетевых систем от злоупотребления услугами, основанный на необходимости выполнения на стороне клиента некоторой достаточно длительной работы (нахождение решения задачи), результат которой легко и быстро проверяется на стороне сервера.

Visa и MasterCard проводят 2000 транзакций в секунду, в то время как Биткоин может провести только 7. Биткоин использует блокчейн не потому что он обеспечивает более быстрые и дешёвые транзакции, а потому что он устраняет необходимость доверять стороннему посреднику: транзакции проходят потому что узлы конкурируют, чтобы подтвердить их, но при этом ни один узел не обязан быть доверенным. Невозможно представить, чтобы сторонние посредники могли повысить производительность, используя технологию, которая жертвует эффективностью и скоростью именно для удаления сторонних посредников. Для любой валюты, контролируемой централизованно, всегда будет более эффективно централизованно записывать транзакции. Является ли устранение сторонних посредников достаточно сильным преимуществом, способным оправдать повышенную неэффективность распределённых регистров – это вопрос, на который можно ответить только спустя несколько лет проверки восприятия цифровых валют рынком. Что действительно ясно видно – так это то, что платёжные приложения с блокчейном должны будут использовать не централизованно-контролируемые валюты, а собственную децентрализованную валюту [21].

### **в) Контракты**

В настоящее время контракты составляются юристами, рассматриваются судами и исполняются полицией [14, 19]. Криптографические системы с интеллектуальными контрактами (смарт-контракты), такие как Ethereum, кодируют контракты в блокчейн для их немедленного вступления в силу без возможности их обжалования или отмены, и вне зоны действия судов и полиции. «Код – это закон» – это девиз, используемый программистами интеллектуальных контрактов. Проблема этой идеи в том, что язык, который юристы используют для составления контрактов понимают гораздо больше людей, чем язык кода, используемый разработчиками смарт-контрактов. Существует наверное только несколько сотен людей во всём мире с техническими знаниями, достаточными, чтобы полностью понять сущность интеллектуальных контрактов и даже они могут пропустить явные программные ошибки. Всё это стало очевидным при первой реализации умных контрактов в сети Ethereum Децентрализованной Автономной организации. После того, как более 150-ти миллионов долларов были проинвестированы в этот умный контракт, злоумышленник смог выполнить программный код так, что перенаправил треть всех активов ДАО на их собственный счёт. Было бы проблематично сказать что эта атака является кражей, поскольку все вкладчики согласились с тем, что их деньги будут контролироваться только программным кодом и ничем другим, а злоумышленник не делал ничего, кроме как выполнил этот самый код, как только он был утверждён вкладчиками. После взлома ДАО, разработчики Ethereum попытались откатить свой блокчейн, чтобы отменить транзакции злоумышленника, в результате чего сеть Ethereum разделилась на две сети с двумя разными валютами, одна из которых подтверждала атаку ДАО, а другая отменила её. Это «разветвление» ставит под вопрос заявления о неизменяемости блокчейна Ethereum. Вычислительная мощность Ethereum (вторая по величине криптовалюта) достаточно мала, чтобы организованная группа программистов могла решить отменить транзакции, потому что в их контрактах были ошибки и суметь забрать себе большую часть вычислительных мощностей сети. Этот случай также поставил под вопрос целесообразность использования умных контрактов, поскольку он показал, что их не так уж сложно обойти. Ввиду того, что блокчейн можно откатить, умные контракты не заменили суды программным кодом, но они заменили суды разработчиками ПО с небольшим опытом, знаниями и ответственностью в арбитражном деле.

ДАО был первым и пока единственным сложным проектом с использованием умных контрактов на блокчейне и опыт показывает, что более до широкого внедрения ещё далеко. Все остальные проекты сейчас существуют только как прототипы. Возможно, в предполагаемом будущем, когда грамотность в написании кода будет более частым явлением, а сам код будет более предсказуемый и надёжный, такие контракты могут стать более распространены. В обозримом будущем спрос, вероятно, будет найден только на простые контракты, код которых может быть легко проверен и будет понятен многим. Единственные значимые существующие проекты с использованием блокчейн контрактов представляют собой простые запрограммированные по времени платежи и мультисигнатурные кошельки, которые все используют валюту самого блокчейна и работают в основном в сети Биткоин.

**с) Управление базами данных и записями:**

Блокчейн – это надёжная и защищённая база данных и регистр учёта средств, но только для собственной валюты блокчейна и только если валюта достаточно ценна для сети, чтобы иметь достаточно большие вычислительные мощности, чтобы противостоять атаке. Для любого другого актива, физического или цифрового, блокчейн надёжен только как ответственный за установление связи между активом и тем, кто или кто обращается к нему в сети. Здесь нет никакой эффективности или прозрачности от использования блокчейна с контролируемым доступом, потому что блокчейн надёжен только как сторона, предоставляющая разрешение записи данных [15, ]. Внедрение блокчейна как стороны, занимающейся ведением записей только сделает его медленнее при этом не добавляя ни безопасности, ни неизменяемости, так как не выполняется Proof-of-Work. Доверие сторонним посредникам должно по-прежнему оставаться, пока вычислительные мощности и время, необходимое для работы базы данных увеличиваются. Блокчейн, защищённый ключом безопасности, может быть использован в качестве нотариальной услуги, где контракты или документы хэшируются в блок транзакций, что позволяет любой стороне получить доступ к договору и быть уверенной, что отображаемая версия является именно той, которая была хэширована первоначально. Такой сервис обеспечит рынок ограниченного пространства блоков, но его работа невозможна ни с одной блокчейн сетью без валюты.

**III. Экономические недостатки технологии блокчейн**

Из изучения трёх вышеупомянутых потенциальных применений технологии блокчейн, можно определить 4 основных препятствия более широкого её внедрения.

**а) Избыточность**

Регистрация каждой транзакции каждым участником сети – это очень дорогостоящая избыточность данных, единственной целью которой является устранение посредничества. Для любого посредника, финансового или юридического, нет смысла добавлять эту избыточность, оставаясь посредником. У банка нет веских оснований делиться записями всех своих операций со всеми банками. Нет также никаких законных причин для того, чтобы какой-либо банк хотел получить полные данные о сделках других банков друг с другом. Эта избыточность предлагает повышенные затраты без какой-либо мыслимой выгоды [8, 9].

**б) Масштабирование:** распределенная сеть, где все узлы регистрируют все транзакции, будет иметь общий регистр транзакций, растущий экспоненциально быстрее, чем количество членов сети. Таким образом, нагрузка на хранилище и вычислительную нагрузку участников сети со временем станет слишком большой, чтобы они смогли справиться с ростом размера сети. Блокчейн всегда будет сталкиваться с этим барьером для эффективного масштабирования, и это объясняет, почему, когда разработчики биткойна ищут решения для масштабирования, они отходят от чисто децентрализованной модели блокчейна в сторону очистки платежей из блокчейна посредниками. Существует очевидный компромисс между масштабом и децентрализацией. В случае создания блокчейна для размещения больших объемов транзакций блоки должны быть увеличены, что увеличит стоимость присоединения к сети и приведет к уменьшению количества узлов, что сделает сеть более централизованной. Наиболее экономически эффективным способом выполнения большого объема транзакций является централизация в одном узле [20].

**с) Соблюдение норм:** блокчейны с собственной валютой, такой как биткойн, существуют независимо от закона, поскольку нет ничего, что государственные органы могли бы сделать, чтобы изменить их работу или как-то повлиять на неё, а председатель Федеральной резервной системы заявил, что не имеет полномочий регулировать биткойн. Транзакции будут

очищены, если действительны, и не будут очищены, если не действительны, и никакие регулирующие органы ничего не могут сделать, чтобы отменить консенсус вычислительной мощности сети. Применение технологии блокчейн в таких жестко регулируемых отраслях, как юриспруденция или финансы, с валютами, отличными от биткойна, приведут к проблемам с регулированием и юридическим осложнениям. Нормы были разработаны для инфраструктуры, сильно отличающейся от блокчейна и не могут быть легко адаптированы к его работе, основанной на полной открытости информации о распространении всех записей среди всех членов сети. Кроме того, блокчейны работают в реальном времени, параллельно в юрисдикциях различных регулирующих правил, что затрудняет обеспечение соблюдения всех норм [10].

**d) Необратимость:** с помощью платежей через посредников, человеческие или программные ошибки могут быть легко исправлены при обращении к посреднику. В блокчейне все гораздо сложнее. Если блок был подтвержден, и новые блоки прикрепляются к нему, возможно только отменить все его транзакции путем объединения 51% вычислительной мощности сети для "жесткого разветвления" сети, в случае когда все эти узлы согласны одновременно перейти на измененную цепочку блоков. В конце концов, технология Блокчейн предназначена для тиражирования кассовых транзакций в режиме онлайн - таким образом она повторяет необратимость кассовых операций и не имеет каких-либо преимуществ, по сравнению с технологиями с сохранением посредничества.

По всей вероятности, такое разделение никогда не будет успешным, если попытаться использовать биткойн, так как это потребовало слишком от многих различных участников согласовать их действия и потратить ресурсы без выгоды. После инцидента в DAO Становится очевидным, что для любого блокчейна, кроме биткойна, сетевой хэшрейт достаточно мал, и разработчики валюты достаточно влиятельны, чтобы откатить части блокчейна, которые им не нравятся. Это означает, что утверждение о неизменности технологии blockchain по-настоящему действительно только в случае биткойна. Для любого другого блокчейна операторы самого блокчейна или регулирующего органа могут фактически поменять данные. Изменяемый блокчейн - совершенно бессмысленная инженерная софистика: он использует очень сложный и дорогой метод для очистки, чтобы удалить посредников и установить неизменность, но затем дает посреднику возможность отменить неизменность. Текущий передовой подход в этих областях содержит обратимость и контроль со стороны юридических и регулирующих органов, но использует более дешевые, быстрые и более эффективные методы.

**e) Безопасность:** безопасность базы данных блокчейна полностью зависит от вычислительных затрат на проверку транзакций и Proof-of-Work. Технология Блокчейн может быть лучше понята, если говорить о ней, как о преобразовании электроэнергии в поддающиеся проверке беспорядочные записи о праве собственности и транзакциях. Чтобы эта система была безопасной, проверяющие, которые расходуют вычислительную мощность, должны быть вознаграждены валютой самой платежной системы, чтобы корректировать их вознаграждение по отношению к работоспособности и долговечности сети. Если бы оплата за вычислительную мощность производилась в любой другой валюте, тогда блокчейн по сути являлся бы частным архивом, поддерживаемым тем, кто платит за вычислительную мощность.

Безопасность системы опирается на безопасность центральной стороны, финансирующей майнеров, но она ставится под сомнение операциями в распределённом регистре, которые открывают множество возможностей для нарушения безопасности. Чем более открыта децентрализованная система и чем большее количество участников её сети расходуют свои вычислительные мощности на проверку, тем более безопасна эта система. Централизованная

система, зависящая от работоспособности одного элемента тем менее безопасна, чем больше, количество членов сети, способных записывать данные в блокчейн, так как каждый присоединённый участник сети является потенциальной угрозой безопасности.

#### **IV. Технология Блокчейн как механизм создания цифровых денег**

Пока единственное коммерчески успешное применение технологии блокчейн - это криптовалюта, и в частности, биткойн. Наиболее распространенные потенциальные применения, рекламируемые для технологии блокчейн, платежей, контрактов и реестра активов работают только в той степени, в которой они работают с использованием децентрализованной валюты блокчейна. Все блокчейны без валют не перешли от стадии прототипа к коммерческому внедрению, потому что они не могут конкурировать с текущими передовыми практиками на своих рынках. Техническая реализация Биткойна свободно доступна в сети почти 8 лет, и разработчики могут копировать и совершенствовать с её помощью коммерческие продукты, но таких продуктов не появилось.

Маркетинговое исследование показывает, что избыточность записи транзакций и проверки работоспособности может быть оправдана только если целью является создание цифровой кассы и платежной сети без посредничества третьих лиц. Информация о праве владения криптовалютой и о транзакциях может передаваться в данных очень маленького объема. Другие экономические случаи, требующие большего объема данных, такие как массовые платежи, контракты и реестры активов, быстро становятся неоправданно громоздким в модели блокчейна. Для любых приложений, которые подразумевают привлечение посредников, блокчейн предлагает неконкурентоспособное решение. Широкое внедрение технологии блокчейн в отраслях, зависящих от доверия посредникам, невозможно, так как простое присутствие посредника делает все расходы, связанные с использованием блокчейна, излишними.

Хорошая техническая разработка начинается с ясной проблемы и пытается найти оптимальное решение для нее, которая не только решает проблему, но также не содержит в себе никакого неуместного или лишнего избытка. Создатель Биткойна был мотивирован созданием «децентрализованной электронной валюты», и он разработал свой проект именно для этого. Нет никаких оснований ожидать, что он подойдет для других функций. После восьми лет и миллионов пользователей можно с уверенностью сказать, что его проект преуспел в производстве криптовалюты, и, что неудивительно, больше ни в чём. Эта цифровая валюта может иметь коммерческие применения, но обсуждать блокчейн технологию как технологическую инновацию, применимую самостоятельно в различных областях, не имеет смысла. Блокчейн лучше воспринимать как механизм создания криптовалюты. Это всего лишь винтик в механизме цифровых денег.

#### **Список литературы**

1. Satoshi Nakamoto. Bitcoin: A Peer-to-Peer Electronic Cash System. [Электронный ресурс], URL: <https://bitcoin.org/bitcoin.pdf>
2. Генкин А., Михеев А. Блокчейн. Как это работает и что ждёт нас завтра – М.: Альпина Паблишер, 2018.
3. Arvind Narayanan, Joseph Bonneau и др. Биткойн и криптовалютные технологии – Принстонский Университет, 2015 (перевод ForkLog.com).
4. Лоран Лелу. Блокчейн от А до Я. Всё о технологии десятилетия – М.: Эксмо, 2018.
5. Уильям Могайар, Виталик Бутерин. Блокчейн для бизнеса. М.: Эксмо, 2017.

6. Samuel Falcon. The story of the DAO – Its history and consequences. [Электронный ресурс], URL: <https://medium.com/swlh/the-story-of-the-dao-its-history-and-consequences-71e6a8a551ee>, 2017.
7. Имран Башир. Блокчейн: архитектура, криптовалюты, инструменты разработки, смарт-контракты. М.: ДМК-Пресс, 2019.
8. Jose Pagliery. Bitcoin is under attack. [Электронный ресурс] URL: <https://money.cnn.com/2014/02/12/technology/security/bitcoin-attack>, 2014.
9. Andreas M. Antonopoulos. The Blockchain. Mastering Bitcoin. — O'Reilly Media, Inc., 2014.
10. Pedro Franco. The Blockchain. Understanding Bitcoin: Cryptography, Engineering and Economics. — John Wiley & Sons, 2014.
11. Мелани Свон. Блокчейн: Схема новой экономики. — Олимп-Бизнес, 2016.
12. Алекс Тапскотт, Дон Тапскотт. Технология блокчейн - то, что движет финансовой революцией сегодня. М.: Эксмо, 2017.
13. Федотова В.В., Емельянов Б.Г., Типнер Л.М. Понятие Блокчейн и возможности его использования. – журнал European Science, 2018.
14. Акулич М. Блокчейн и логистика. Ridero, 2019.
15. Технология распределённых баз данных (перевод). IOSCO Research Report on Financial Technologies, 2017
16. Bitcoin and Cryptocurrency technologies – курс на основе лекций Принстонского университета [Электронный ресурс] URL: <https://www.coursera.org/learn/cryptocurrency>.
17. Бабкин А.В., Буркальцева Д.Д., Гук О.А., Тюлин А.С. Анализ развития и регулирования криптовалют: зарубежный и российский опыт. – журнал Мир (Модернизация. Иновации. Развитие), 2017.
18. Адам Теппер. Биткоин – деньги для всех. 2015.
19. Пискунов И. Блокчейн: атаки, безопасность и криптография. [Электронный ресурс], URL: [https://www.securitylab.ru/blog/personal/Informacionnaya\\_bezопасnost\\_v\\_detalyah/343072.php](https://www.securitylab.ru/blog/personal/Informacionnaya_bezопасnost_v_detalyah/343072.php), 2017.
20. Проблема масштабируемости Биткойна. [Электронный ресурс], URL: [https://ru.wikipedia.org/wiki/Проблема\\_масштабируемости\\_биткойна](https://ru.wikipedia.org/wiki/Проблема_масштабируемости_биткойна). 2018.
21. Биткоин распался на две валюты: как это произошло. [Электронный ресурс], URL: <https://www.bbc.com/russian/features-40795000>, 2017.

## References

1. Satoshi Nakamoto Bitcoin: A Peer-to-Peer Electronic Cash System – 2008.
2. Genkin A., Miheev A. Blockchain How does it work and what awaits us tomorrow? - М: Alpina Publisher, 2018. (in Russian)
3. Arvind Narayanan, Joseph Bonneau and etc. Bitcoin and cryptocurrency technologies – Princeton University, 2015 (translated by ForkLog.com).
4. Loran Lely. Blockchain from A to Z. Everything about technology of the decade. – М.: Exmo, 2018. (in Russian)
5. William Mogayar, Vitalik Buterin. Blockchain for business. М.: Exmo, 2017. (in Russian)

6. Samuel Falcon. The story of the DAO – Its history and consequences. [Electronic resource], URL: <https://medium.com/swlh/the-story-of-the-dao-its-history-and-consequences-71e6a8a551ee>, 2017.
  7. Imran Bashir. Blockchain: architecture, cryptocurrency, development tools, smart contracts. M.: DMK-press, 2019.
  8. Jose Pagliery. Bitcoin is under attack. [Electronic resource] URL: <https://money.cnn.com/2014/02/12/technology/security/bitcoin-attack>, 2014.
  9. Andreas M. Antonopoulos. The Blockchain. Mastering Bitcoin. — O'Reilly Media, Inc., 2014.
  10. Pedro Franco. The Blockchain. Understanding Bitcoin: Cryptography, Engineering and Economics. — John Wiley & Sons, 2014.
  11. Melanie Swan. Blockchain: Scheme of the new economy.— Olimp-Business, 2016.
  12. Alex Tapscott, Don Tapscott. Blockchain technology is what drives the financial revolution today. M.: Exmo, 2017.
  13. Fedotova V.V., Emelyanov B.G., Tipner L.M. The concept of blockchain and the possibility of its use.– journal European Science, 2018.
  14. Akulich M. Blockchain and Logistics. Ridero, 2019.
  15. Technology of distributed databases (translation). IOSCO Research Report on Financial Technologies, 2017
  16. Bitcoin and Cryptocurrency technologies – course based on Princeton University lectures [Electronic resource] URL: <https://www.coursera.org/learn/cryptocurrency>.
  17. Babkin A.V., Burkaltseva D.D., Guk O.A., Tyulin A.S. Analysis of the development and regulation of cryptocurrency: foreign and Russian experience. – Mir journal (Modernization. Innovations. Development), 2017.
  18. Adam Tapper. Bitcoin - money for everyone. 2015.
  19. Piskunov I. Blockchain: Attacks, Security, and Cryptography. [Electronic resource], URL: [https://www.securitylab.ru/blog/personal/Informacionnaya\\_bezopasnost\\_v\\_detalyah/343072.php](https://www.securitylab.ru/blog/personal/Informacionnaya_bezopasnost_v_detalyah/343072.php), 2017.
  20. Bitcoin scalability problem. [Electronic resource], URL: [https://ru.wikipedia.org/wiki/Проблема\\_масштабируемости\\_биткойна](https://ru.wikipedia.org/wiki/Проблема_масштабируемости_биткойна). 2018.
  21. Bitcoin broke up into two currencies: how it happened. [Electronic resource] URL: <https://www.bbc.com/russian/features-40795000>, 2017.
-



Международный журнал информационных технологий и энергоэффективности

Сайт журнала:

<http://www.openaccessscience.ru/index.php/ijcse/>



УДК 66.011

## ЭНЕРГОЭФФЕКТИВНОСТЬ МАССОПЕРЕНОСА В НАСАДОЧНЫХ АППАРАТАХ

Дударовская О.Г.

ФГБОУ ВО «Казанский государственный энергетический университет», Казань, Россия (420066, г. Казань, ул. Красносельская, 51); e-mail: dg.olga5@mail.ru

Настоящая статья посвящена оценке энергетической эффективности различных интенсификаторов в виде хаотичных насадочных элементов в массообменных аппаратах насадочного типа. Выбор наилучшего интенсификатора определяется по энергетическому комплексу, аналогично полученному на основе одного из наиболее универсальных критериев оценки энергоэффективности теплообменных аппаратов, что позволяет определить эффективность процесса с изменением исходных параметров. Для определения средних значений массообменных характеристик применяется модель пограничного слоя Owena с функцией турбулентной вязкости с учетом затухания пульсаций в вязком подслое. С целью выбора наиболее энергоэффективного интенсификатора процесса массообмена рассмотрены несколько видов нерегулярных насадок. Выполнены расчеты энергетического коэффициента  $E$  для канала, заполненного насадками различной формы и размера. По полученным данным построены графики зависимости энергетического коэффициента  $E$  от числа Рейнольдса (режима движения).

Ключевые слова: массообмен, интенсификаторы, энергоэффективность.

## ENERGY EFFICIENCY OF MASS TRANSFER IN NOZZLES

Dudarovskaya O.G.

Federal state-funded educational institution of the higher education "Kazan state power engineering university" Kazan, Russia (420066, Kazan, street Krasnoselsky, 51); e-mail: dg.olga5@mail.ru

This article is devoted to assessing the energy efficiency of various intensifiers in the form of random packing elements in packed-type mass transfer apparatuses. The choice of the best intensifier is determined by the energy complex, similar to that obtained on the basis of one of the most universal criteria for evaluating the energy efficiency of heat exchangers, which allows you to determine the efficiency of the process with a change in the initial parameters. To determine the average values of mass transfer characteristics, the Owena boundary layer model with a turbulent viscosity function is used taking into account the damping of pulsations in a viscous sublayer. In order to choose the most energy-efficient intensifier of the mass transfer process, several types of irregular nozzles are considered. The energy coefficient  $E$  was calculated for a channel filled with nozzles of various shapes and sizes. Based on the data obtained, graphs of the dependence of the energy coefficient  $E$  on the Reynolds number (motion mode) are constructed.

Keywords: mass transfer, intensifiers, energy efficiency.

В настоящее время основными потребителями топливно-энергетических ресурсов являются предприятия энергетической и нефтехимической отрасли, при этом одними из энергоемких процессов считаются массообменные.

Сейчас промышленные предприятия должны доказывать свою конкурентоспособность не только на внутренних, но и на внешних рынках, что в конечном счете требует уменьшения энергопотребления для снижения себестоимости выпускаемой продукции.

Наиболее эффективным и сбалансированным решением по совершенствованию технологических процессов и оборудования на данных предприятиях является использование технически прогрессивных технологий, а также привлечение высокоэффективного интенсифицированного оборудования.

В качестве инструмента интенсификации массообменных процессов нашли широкое применение высокоэффективные интенсификаторы, в виде хаотичных насадочных элементов, путем развития поверхности контакта и улучшения перемешивания потоков [1,2]. В настоящее время существует большое многообразие различных насадочных элементов [3,4] и поэтому выбор того или иного интенсификатора определяется рядом условий, а именно: значениями допустимых гидравлических потерь и энергетических затрат, соответствием критериев безотказной работы (прочность, коррозия), технологичностью, дешевизной изготовления и т.д.

Разумеется, выбор высокоэффективного интенсификатора в каждом случае индивидуален и определяется назначением аппарата, его конструкцией, свойствами рабочей среды и другими немаловажными факторами. Также как известно, применение любого из известных методов интенсификации сопровождается ростом гидравлического сопротивления, что увеличивает затраты мощности на прокачку жидкости, поэтому одним из основных показателей при расчете эффективности аппарата является его энергетическая эффективность.

В связи с чем рассматриваемая задача является актуальной и требует решение достаточно сложных расчетов, поэтому целесообразным является применение методов математического моделирования.

### **Определение энергетического коэффициента**

Рассмотрим процесс стационарной массоотдачи в потоке от элементов неподвижного насадочного слоя в канале (колонне).

Для определения энергетических характеристик массообменных аппаратов насадочного типа воспользуемся комплексом, аналогично полученным на основе одного из наиболее универсального критерия оценки энергоэффективности теплообменных аппаратов, предложенным В.И. Антуфьевым и М.В. Кирпичевым [5], который примет вид

$$E = \frac{M}{N} = \frac{\beta \cdot F}{Q \cdot \Delta P}, \quad (1)$$

где  $M$  – количество переданного вещества, кг/с;  $N$  – энергия, необходимая для подачи среды, Вт;  $\beta$  – коэффициент массоотдачи, м/с;  $F = a_v \cdot S \cdot H$  – поверхность массообмена для аппарата (канала) с насадкой, м<sup>2</sup>;  $a_v$  – удельная поверхность насадки, м<sup>2</sup>/м<sup>3</sup>;  $S$  – площадь поперечного сечения канала, м<sup>2</sup>;  $H$  – длина насадочного слоя, м;  $Q = u_0 \cdot S$  – объемный расход среды, м<sup>3</sup>/с;  $u_0$  – скорость потока в канале, м/с;  $\Delta P$  – потери давления в канале с насадкой, Па.

Потери давления в канале с насадкой определяются как [6]

$$\Delta P = \xi \cdot \frac{H}{d_3} \cdot \frac{\rho \cdot u_0^2}{2 \cdot \varepsilon_{св}}, \quad (2)$$

где  $\xi$  – коэффициент гидравлического сопротивления насадочного слоя;  $d_3$  – эквивалентный диаметр насадки, м;  $\rho$  – плотность потока, кг/м<sup>3</sup>;  $\varepsilon_{св}$  – удельный свободный объем насадки, м<sup>3</sup>/м<sup>3</sup>.

Таким образом, энергетический коэффициент получит вид

$$E = \frac{2 \cdot \beta \cdot a_v \cdot d_3 \cdot \varepsilon_{св}^2}{u_0^3 \cdot \xi \cdot \rho}. \quad (3)$$

Таким образом, задача нахождения энергетического коэффициента сводится к определению коэффициентов массоотдачи  $\beta$  и гидравлического сопротивления насадочного слоя  $\xi$ .

На основе аналогии процессов тепло - и массообмена запишем коэффициент массоотдачи

$$\beta = \frac{\alpha}{\rho c_p} \left( \frac{Pr}{Sc} \right)^{0,333}, \quad (4)$$

где  $\alpha$  – коэффициент теплоотдачи, Вт/(м<sup>2</sup>·К);  $c_p$  – теплоемкость при постоянном давлении, Дж/(кг·К);  $Pr$  – число Прандтля;  $Sc$  – число Шмидта.

Для определения коэффициента массоотдачи в канале, заполненном хаотичными насадочными элементами, воспользуемся моделью пограничного слоя Owena с функцией турбулентной вязкости с учетом затухания пульсаций в вязком подслое [7].

Запишем характеристики турбулентного обмена [7]

$$\frac{\nu_T}{\nu} = 0,001 (y^+)^3, \quad \text{где } y^+ \in [0; 5], \quad (5)$$

$$\frac{\nu_T}{\nu} = 0,012 (y^+ - 1,6)^2, \quad y^+ \in [5; 20], \quad (6)$$

$$\frac{\nu_T}{\nu} = \chi (y^+ - 10), \quad y^+ \in [20; \delta], \quad (7)$$

где  $\nu_T$  – турбулентный коэффициент вязкости, м<sup>2</sup>/с;  $\nu$  – коэффициент кинематической вязкости, м<sup>2</sup>/с;  $y^+$  – безразмерная координата в пограничном слое.

На основе выражения (4), используя трехслойную модель турбулентного пограничного слоя Owena (5) – (7), коэффициент массоотдачи примет вид

$$\beta = \frac{1,85 \cdot Re_3^{0,75} \cdot (\xi / 2)^{0,25} \cdot \nu}{Sc^{0,67} \cdot d_3 \cdot \left[ 1,48 Re_3^{0,125} / \xi^{0,25} + 2,5 \ln \left( 4 \cdot Re_3^{0,125} \cdot \xi^{0,5} \right) \right]}, \quad (8)$$

где  $Re_3$  – число Рейнольдса эквивалентное.

Также энергетический коэффициент  $E$  можно записать через безразмерный комплекс Стантона  $St_3$

$$E = \frac{8 \cdot St_3}{\xi \cdot \rho \cdot u_{ср}^2}, \quad (9)$$

где безразмерный комплекс Стантона имеет вид

$$St_3 = \frac{1,85 \cdot (\xi / 2)^{0,25}}{Re_3^{0,25} \cdot Sc^{0,667} \cdot \left[ 1,48 \cdot Re_3^{0,125} / \xi^{0,25} + 2,5 \cdot \ln \left( 4 \cdot Re_3^{0,125} \cdot \xi^{0,5} \right) \right]}. \quad (10)$$

В данной работе приведем выражения гидравлического сопротивления  $\xi$  для некоторых насадочных элементов [6]:

для насадок кольцеобразной формы

$$\xi = \frac{16}{Re_3^{0,2}}, \quad (11)$$

для насадок Инжихим – 2002

$$\xi = 1,34 \cdot \left( \frac{64}{Re_3} + \frac{1,8}{Re_3^{0,08}} \right), \quad (12)$$

для насадок седлообразной формы

$$\xi = \frac{133}{Re_3} + 2,34. \quad (13)$$

Применим полученные выражения энергетического коэффициента  $E$  для выбора наиболее энергоэффективного интенсификатора, в виде хаотичных насадочных элементов. Для этого рассмотрим случай, когда воздух пропускают через слой насадки, покрытых нафталином.

Исходные данные принимаем следующие [8]:

- - коэффициент диффузии паров нафталина в газовой фазе  $D_T = 5,1 \cdot 10^{-8} \text{ м}^2/\text{с}$ ;
- - коэффициент кинематической вязкости воздуха  $\nu = 2,2 \cdot 10^{-5} \text{ м}^2/\text{с}$ ;
- - плотность воздуха  $\rho = 0,963 \text{ кг}/\text{м}^3$ .

Сначала рассмотрим в качестве базисной насадки кольцевую форму как наиболее благоприятную с точки зрения гидродинамики. В таблице 1 приведены характеристики насадок колец Палля различного размера из полипропилена [9].

Таблица 1 – Характеристики колец Палля различного размера

Размер насадки, мм	Эквивалентный диаметр $d_э$ , м	Удельная поверхность $a_v$ , $\text{м}^2/\text{м}^3$	Удельный свободный объем $\varepsilon_{св}$ , $\text{м}^3/\text{м}^3$
кольца Палля (разм. 15)	0,01024	340	0,87
кольца Палля (разм. 25)	0,017	206	0,9
кольца Палля (разм. 50)	0,029	124	0,9

На рисунке 1 представлены результаты расчетов энергетического коэффициента  $E$  от числа Рейнольдса  $Re$  для интенсификаторов, в виде колец Палля различного размера.

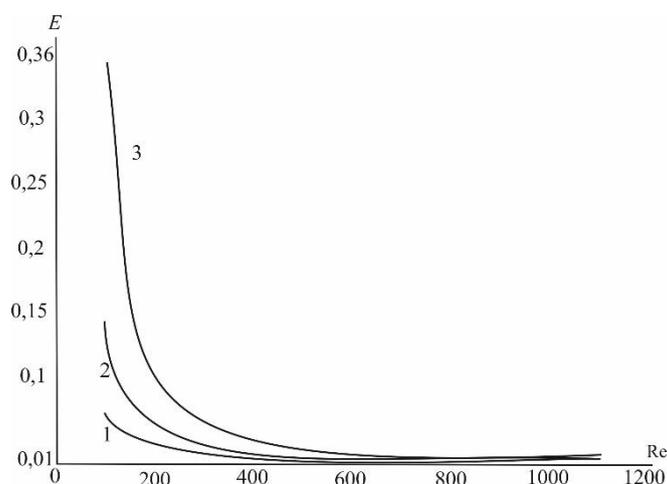


Рисунок 1 – Зависимость энергетического коэффициента  $E$  от числа Рейнольдса  $Re$ : 1 – кольца Палля (разм. 15 мм); 2 – кольца Палля (разм. 25 мм); 3 – кольца Палля (разм. 50 мм).

Как видно из рисунка, насадки Палля (разм. 50 мм) наиболее энергоэффективны по сравнению с двумя другими насадками (разм. 15 мм) и (разм. 25 мм), поскольку имеют более высокие значения энергетического коэффициента. Это связано с тем, что с увеличением размера насадки снижается ее удельная поверхность при этом коэффициент гидравлического сопротивления насадочного слоя уменьшается, а коэффициент массоотдачи увеличивается, что в целом приводит к повышению энергетического коэффициента.

Далее для анализа рассмотрим насадки различной формы, но одинакового размера, характеристики насадок [9] представлены в таблице 2.

Таблица 2 – Характеристики насадок одинакового размера (50 мм), но различной формы

Наименование насадки	Эквивалентный диаметр $d_э$ , м	Удельная поверхность $a_v$ , $m^2/m^3$	Удельный свободный объем $\varepsilon_{св}$ , $m^3/m^3$
кольца ГИАП-НЗ	0,038	101	0,96
Инжехим -2002	0,019	200	0,95
седла Инталокс	0,026	118	0,79

На рисунке 2 показаны результаты расчетов энергетического коэффициента  $E$  от числа Рейнольдса  $Re$  для насадок различной формы.

Как видно из характеристик насадок, насадки кольцеобразной и седлообразной формы имеют весьма близкие значения удельной поверхности  $a_v$ , однако из-за различия в конструкции их величины  $\Delta P$  различаются между собой, что связано с природой возникновения гидравлического сопротивления в насадочном слое. В целом, кольцеобразные насадки, имеющие внутреннюю полость, показывают хорошие результаты, но не сильно отличающиеся от седлообразных.

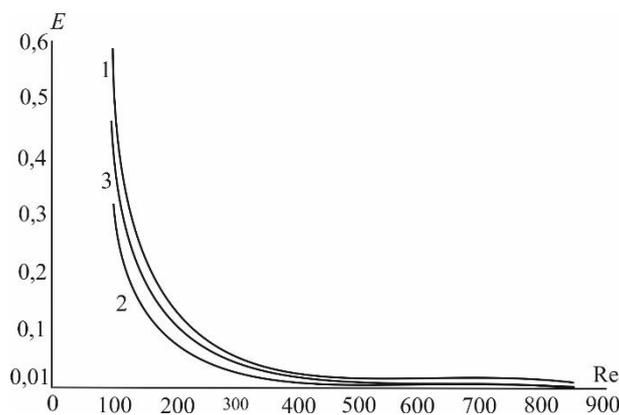


Рисунок 2 – Зависимость энергетического коэффициента  $E$  от числа Рейнольдса  $Re$  для насадок различной формы: 1 – кольца ГИАП-НЗ (разм. 50 мм); 2 – насадки Инжехим-2002 (разм. 50 мм); 3 – седла Инталокс (разм. 50 мм).

Таким образом, как показывают результаты расчетов с точки зрения интенсификации процесса массообмена представляется интересным применение достаточно крупной насадки кольцеобразной формы. И во всех рассмотренных случаях с увеличением числа Рейнольдса значение коэффициента энергетической эффективности  $E$  снижается, что связано с увеличением мощности на прокачивание среды пропорционально увеличению средней скорости среды  $u_{\text{ср}}^3$ .

Рассмотренный подход определения энергетического коэффициента позволяет проводить сопоставление различных по форме и размерам интенсификаторов, что удобно при выборе наиболее энергоэффективного и энергосберегающего интенсификатора.

Также данный подход позволит решить проблемы реальных производственных объединений при изменении типов или конструкций установленного оборудования, в том числе при преобразовании их структуры, а также учесть многочисленные факторы, влияющие на эффективность работы усовершенствованного оборудования в различных режимах работы.

### Список литературы

1. Дударовская О.Г. Модели интенсифицированного тепломассообмена и смешения сред в каналах с хаотичными насадочными слоями: дис. ...канд. техн. наук. Казань, 2016. 202 с.
2. Булкин В.А. Разработка и исследование массообменного аппарата с прямоточными вихревыми контактными устройствами: дис. ...канд. техн. наук. Казань, 1970. 137 с.
3. Микуленок И.О. Классификация насадок массообменных аппаратов // Химическая промышленность. 2011. № 2. С. 67.
4. Лаптев А.Г. Методы интенсификации и моделирования тепломассообменных процессов. Москва: Теплотехник, 2011. 288 с.
5. Кирсанов Ю.А. Циклические тепловые процессы и теория теплопроводности в регенеративных воздухоподогревателях. М.: ФИЗМАТЛИТ, 2007. 240 с.

6. Лаптев А.Г., Фарахов Т.М., Дударовская О.Г. Эффективность явлений переноса в каналах с хаотичными насадочными слоями. Страта: Санкт Петербург, 2016. 202 с.
7. Owen P. Dust deposition from a turbulent airstream. In: Aerodynamic Capture of Particles /Ed. E.G. Richardson. London. New York, 1960. pp. 8-25.
8. Павлов К.Ф., Романков П.Г., Носков А.А. Примеры и задачи расчета по курсу процессов и аппаратов химической технологии. М.: Химия, 1970. 626 с.
9. Каган А.М., Лаптев А.Г., Пушнов А.С., Фарахов М.И. Контактные насадки промышленных теплообменных аппаратов. Казань: Отечество, 2013. 454 с..

#### References

1. Dudarovskaya O.G. Models of intensified heat and mass transfer and medium mixing in channels with chaotic packed layers: dis. ... cand. tech. sciences. Kazan, 2016. 202 p.
  2. Bulkin V.A. Development and research of mass transfer apparatus with direct-flow vortex contact devices: dis. ... cand. tech. sciences. Kazan, 1970. 137 p.
  3. Mikulenok I.O. Classification of nozzles of mass transfer apparatus // Chemical industry. 2011. No. 2. P. 67.
  4. Laptev A.G. Methods of intensification and modeling of heat and mass transfer processes. Moscow: Heat engineer, 2011. 288 p.
  5. Kirsanov Yu.A. Cyclic thermal processes and the theory of thermal conductivity in regenerative air heaters. М.: FIZMATLIT, 2007. 240 p.
  6. Laptev A.G., Farakhov T.M., Dudarovskaya O.G. The effectiveness of transport phenomena in channels with chaotic packed layers. Strata: St. Petersburg, 2016. 202 p.
  7. Owen P. Dust deposition from a turbulent airstream. In: Aerodynamic Capture of Particles / Ed.E.G. Richardson. London New York, 1960. pp. 8-25.
  8. Pavlov K.F., Romankov P.G., Noskov A.A. Examples and tasks of calculation at the rate of processes and devices of chemical technology. М.: Chemistry, 1970. 626 p.
  9. Kagan A.M., Laptev A.G., Pushnov A.S., Farakhov M.I. Contact nozzles industrial heat and mass transfer apparatus. Kazan: Fatherland, 2013. 445 p..
-