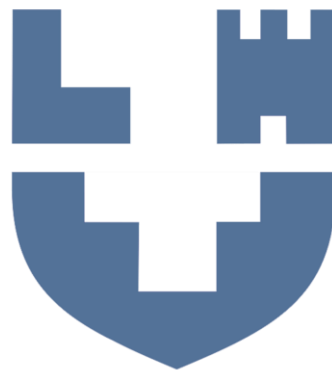


*МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ЛУЦЬКИЙ НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ*

**КОМП'ЮТЕРНО-ІНТЕГРОВАНІ
ТЕХНОЛОГІЇ:
ОСВІТА, НАУКА, ВИРОБНИЦТВО**

НАУКОВИЙ
ЖУРНАЛ



Головний редактор – професор, д.т.н., Гордєєв О.О.

№49 2022

м. Луцьк

РЕДАКЦІЙНА КОЛЕГІЯ:

Головний редактор:	
професор, д.т.н. Гордєєв О.О.	(м. Луцьк)
Відповідальний секретар:	
ст. викл., к.т.н. Христинець Н.А.	(м. Луцьк)
Члени редакційної колегії:	
проф., д.т.н. Андрущак І.Є.	(м. Луцьк)
доц., к.т.н. Ліщина Н.М.	(м. Луцьк)
проф., д.пед.н. Тулашвілі Ю.Й.	(м. Луцьк)
доц., к.т.н. Ліщина В.О.	(м. Луцьк)
доц., к.т.н. Здолбіцька Н.В.	(м. Луцьк)
проф., PhD. Milosz Marek	(Польща, м. Люблін)
проф., д.т.н. Мельник А.О.	(м. Львів)
проф., д.т.н. Мороз Б.І.	(м. Дніпро)
проф., д.т.н. Тарасенко В.П.	(м. Київ)
проф., PhD. Alison McMillan	(Великобританія)
проф., PhD. Дехтяр Ю.Д.	(Литва, м. Рига)
доц., к.т.н. Мельник К.В.	(м. Луцьк)
проф., д.пед.н. Чернящук Н.Л.	(м. Луцьк)
доц., к.т.н. Костючко С.М.	(м. Луцьк)
доц., к.т.н. Дуда О.М.	(м. Тернопіль)
доц., к.т.н. Назаревич О.Б.	(м. Тернопіль)
проф., д.т.н. Сайко В.Г.	(м. Київ)
доц., к.т.н. Дуда О.М.	(м. Тернопіль)
доц., к.т.н. Ткачук А.А.	(м. Луцьк)
доц., к.т.н. Мороз С.А.	(м. Луцьк)
проф., д.т.н. Сайко В.Г.	(м. Київ)
доц., к.т.н. Євсюк М.М.	(м. Луцьк)

Адреса редколегії:

Луцький національний технічний університет,
кафедра комп'ютерної інженерії та кібербезпеки
вул. Львівська 75, ауд.141
м.Луцьк, 43018
тел. (0332) 74-61-15
E-mail: cit@lntu.edu.ua,
сайт журналу: cit-journal.com.ua

КОМП'ЮТЕРНО-ІНТЕГРОВАНІ
ТЕХНОЛОГІЇ:
ОСВІТА, НАУКА, ВИРОБНИЦТВО

№49 2022р.

Журнал засновано у грудні 2010 р.
Свідоцтво про реєстрацію KB № 16705–5277 Р.
Засновник: Луцький національний технічний університет

**Рекомендовано до друку Вченою радою
Луцького національного технічного університету**
(протокол №4 засідання від 24.11.2022р.)

Журнал рішенням МОН України
наказом №515 від 16.05.2016р,
включено в перелік наукових фахових видань.

Видання індексується у
наукометричних та реферативних базах:
Open Academic Journals Index
Academic Resource Index ResearchBib
Rootindexing
Information Matrix for the Analysis of Journals
Ulrichsweb.

ISSN 2524-0560 (Online)
ISSN 2524-0552 (Print)

ЗМІСТ

АВТОМАТИКА ТА УПРАВЛІННЯ	
Колпаков П.С., Марченко О.І. Спосіб оптимізації передачі однотипних JSON-документів комп'ютерними мережами.	5
Тогосв О.Р. Метод деанонімізації користувачів iOS через протокол Airdrop.	12
ІНФОРМАТИКА ТА ОБЧИСЛЮВАЛЬНА ТЕХНІКА	
Багнюк Н.В., Бортник К.Я., Варченко Л.Л. Моніторингова система для операційної системи Windows.	18
Гаврильченко О.В., Корендій В.М., Шенбор В.С. Вплив параметрів вібраційного конвеєра на швидкість транспортування деталей.	24
Єрмоленко Д.В., Марченко О.І. Алгоритми способу порівняння програм, написаних Lisp-подібними мовами, на основі абстрактних семантичних дерев.	29
Козубцов І.М., Сілко О.В., Козубцова Л.М., Лукшенко В.В. Показники та критерії оцінювання рівня академічної доброчесності здобувачів вищої освіти.	38
Левицька Т.О., Герасимов Д.С. Робота присвячена моделюванню багатошарової нейронної мережі для прогнозування обсягів продажу груп товарів.	48
Ніколасвський О.Ю. Бізнес-інформаційні технології як головний аспект конкурентної переваги сучасного бізнесу.	54
Поплавська Г. В., Сачук Ю.В. Методичні аспекти навчання студентів розробки додатку для перевірки знань учнів із використанням Lasagus.	63
Свиридюк К.А., Христинець М.О., Черняшук Н.Л., Христинець Н.А. Визначення впливу ергономіки та пошукової оптимізації на фактори ранжування сайту.	69
Фещенко І. О., Дадиверін В. В., Потапова К.Р., Наливайчук М.В. Адаптування технології суперсемплінгу для збільшення продуктивності рендерингу зображення методом трасування променів у реальному часі.	74

ТЕЛЕКОМУНІКАЦІЇ ТА РАДІОТЕХНІКА

Бурбан О.В., Поліщук М.М., Бортник К.Я. Система голосового керування освітленням на базі Wemos D1 Mini.	80
Поліщук М.М., Саварин П.В., Фуркало С.Ю. Методи Webhooks та Long Polling для розміщення Telegram-бота.	86
Радзівілов Г.Д. Аналіз достовірності прийому повідомлень в IoT технологіях з односторонніми протоколами взаємодії.	93

DOI: <https://doi.org/10.36910/6775-2524-0560-2022-49-01>

УДК 004.42+004.45

Колпаков Пилип Сергійович, бакалавр,

<https://orcid.org/0000-0001-6375-676X>

Марченко Олександр Іванович, к.т.н., доцент,

<https://orcid.org/0000-0002-4537-3420>

Національний технічний університет України «Київський політехнічний інститут імені Ігоря Сікорського», м. Київ, Україна

СПОСІБ ОПТИМІЗАЦІЇ ПЕРЕДАЧІ ОДНОТИПНИХ JSON-ДОКУМЕНТІВ КОМП'ЮТЕРНИМИ МЕРЕЖАМИ

Колпаков П.С., Марченко О.І. Спосіб оптимізації передачі однотипних JSON-документів комп'ютерними мережами. У даній статті проаналізована проблема надлишковості та неефективності формату JSON. Розглянуті існуючі способи оптимізації та запропонований новий спосіб, що дозволяє досягти поставлених задач без втрати основних переваг зазначеного формату. Розглянутий спосіб базується на ідеї попереднього аналізу повідомлень в системі з метою їх структуризації та розділення на окремі класи. Результатом виконання дій нового способу є спеціальний бінарний формат документів, оптимізований для конкретної системи але з можливістю передачі будь-яких повідомлень.

Ключові слова: JSON-формат, JSON-документ, REST-архітектура, оптимізація передачі, бінарний формат, стиснення інформації.

Kolpakov P.S., Marchenko O.I. An optimization technique for transferring of identical JSON documents over computer networks. This paper contains the analysis of the problem of redundancy and inefficiency of the JSON format. The existing optimization techniques are considered and the new technique is proposed, which allows to solve the problem without losing the main advantages of this format. The proposed technique is based on the idea of preliminary analysis of messages in a system with the aim of structuring them and dividing them into separate classes. The result of performing the actions of the new technique is a special binary format optimized for the specific system but with a possibility of transmitting messages of any structure.

Keywords: JSON-format, JSON-document, REST-architecture, optimization of transfer, binary format, data compression.

Постановка проблеми

Формат JSON [1] широко використовується для передачі інформації комп'ютерними мережами. Завдяки своїй розповсюдженості він є простим у впровадженні, а завдяки текстовому представленню даних та динамічній типізації він є простим у використанні. Однак, формат JSON має також ряд серйозних недоліків:

- 1) оскільки JSON-формат є текстовим, а текстове (рядкове) представлення даних не є компактним, то дані у форматі JSON мають значну надлишковість;
- 2) оскільки JSON-формат є динамічно типізованим, структурна інформація (типи значень, назви полів) має передаватися в кожному повідомленні, що є також надлишковим у більшості випадків;
- 3) оскільки JSON-формат є текстовим та динамічно типізованим, процес зчитування даних потребує додаткової логіки обробки (додаткових обчислень).

Підсумовуючи, можна стверджувати: формат JSON є значною мірою надлишковим, і деякою мірою — повільним. У випадках передачі значних об'ємів інформації використання цього формату може стати тим самим "пляшковим горлечком", що суттєво обмежуватиме ефективність усієї програмної системи. Тому, розробка нових способів оптимізації передачі даних у форматі JSON є актуальною задачею.

Розглянемо основні випадки використання формату JSON, щоб знайти простір для оптимізації.

Найбільш широко формат JSON використовується у програмних системах, що реалізують REST-архітектуру [2]. JSON-формат, хоча й не є обов'язковою її складовою, але є типовим форматом передачі повідомлень для цієї архітектури. Як правило, розмір повідомлень є малим, і їх структура зводиться до невеликої множини подібних між собою класів. Фактично, повідомлення є однотипними.

Аналогічна ситуація виникає також з деякими документо-орієнтованими базами даних (MongoDB, DocumentDB), що, фактично, використовують JSON-формат для збереження та доступу до інформації. Ці бази даних вже використовують певні способи оптимізації [3], але ці

способи не є вичерпними, і простір для пропозиції нових рішень з метою збільшення швидкодії або зменшення об'єму стиснених даних все ще залишається.

Як підсумок зазначимо, що у основних випадках використання JSON-формату для передачі або збереження даних повідомлення є однотипними і, відповідно, подальший аналіз будемо проводити з урахуванням цього факту.

Термінологія

Формат JSON— широкоживаний текстовий формат даних, описаний у стандарті RFC8259 [1].

JSON-документ — дані у форматі JSON, представлені у текстовому вигляді об'єктом JSON (*object*, згідно стандарту).

JSON-повідомлення — те ж саме, що і JSON-документ, але з прив'язкою до процесу передачі даних комп'ютерною мережею.

Структурна інформація JSON-повідомлення — відомості щодо конкретного JSON-повідомлення, що вичерпно описують його структуру: імена полів, порядок вкладеності об'єктів/масивів, типи значень.

Атомарне значення — у контексті JSON-повідомлень — значення числового, рядкового типу або порожнього значення (*null*).

Неатомарне значення — у контексті JSON-повідомлень — об'єкт або масив.

Аналіз існуючих рішень

Розглянемо рішення зазначеної проблеми серед існуючих способів. Порівняємо їх за такими критеріями: простота та швидкість кодування та декодування повідомлень, надлишковість повідомлень, простота впровадження оптимізації/технології, безпека даних, збереження основних переваг JSON-формату, наявність додаткових особливостей, переваг, обмежень.

Були проаналізовані наступні рішення:

- 1) використання бінарних аналогів формату JSON [3, 4, 5, 6].
- 2) використання алгоритмів архівації для стиснення повідомлень [8, 9].
- 3) використання статично типізованих аналогів JSON [10, 11].

У якості підсумку зазначимо наступні моменти.

1. Використання бінарних аналогів JSON суттєво пришвидшує обробку даних [7], але зменшення розміру повідомлення не є значимим [7], оскільки ці формати не позбавляють від необхідності зберігати структурну інформацію в кожному повідомленні.

2. Використання алгоритмів архівації задля зменшення об'єму окремих повідомлень не є ефективним. Найчастіше JSON-формат використовується для передачі саме невеликих повідомлень, які по окремоті мають невисокий рівень надлишковості.

3. Використання статично типізованих аналогів JSON – gRPC [10] та Apache Thrift [11] – дозволяє значно зменшити розмір повідомлень та прискорити їх обробку. Для збереження даних в них використовується бінарний формат, що не вимагає передачі структурної інформації. Але ці технології позбавлені основних переваг формату JSON — динамічної типізації та простоти впровадження, оскільки програміст вимушений витратити багато часу на повний вичерпний опис усіх структур даних.

Виконаний аналіз показав, що жоден з проаналізованих способів передачі повідомлень або оптимізації передачі повідомлень не дозволяє вирішити поставлену задачу ефективно. Жоден зі способів не дозволяє достатньою мірою зменшити розмір повідомлення при щонайменше збереженні прийнятної швидкості обробки, або, навіть, прискоренні обробки.

На основі виконаного аналізу, розглянувши переваги та недоліки існуючих способів, можна зробити висновок, що з метою зменшення обсягу даних, що передаються, доцільним є використання бінарного формату даних, який не буде містити структурну інформацію (структура об'єктів, імена полів). Але при цьому, задля збереження основних переваг типової REST-архітектури, варто уникнути явного опису структур даних для кожного повідомлення.

Тому є актуальною розробка нового способу передачі великих обсягів інформації, який, з однієї сторони, вирішував би задачу оптимізації передачі JSON-документів за обсягом даних (в тому числі для прискорення процесу їх обміну), а, з іншої сторони, не вимагав би явного опису програмістом структур даних повідомлень та зберігав би прийнятну швидкість передачі документів. При розробці нового способу треба орієнтуватись на такий цільовий випадок, як передача великої кількості окремих однотипних повідомлень.

Спосіб оптимізації передачі однотипних JSON-документів

Будемо розглядати поставлену задачу при умові, що програмна система, для якої буде розроблюватися спосіб оптимізації передачі повідомлень, вже існує, і, відповідно, ми вже маємо зразки типових JSON-повідомлень, які використовуються в цій системі.

Потрібно зауважити, що якщо при передачі окремого JSON-повідомлення до даних не додається структурна інформація, то вона має зберігатися або передаватися окремо. Частина програмної системи, що відповідає за кодування та декодування повідомлень, може мати перелік усіх можливих класів в собі з самого початку (як у випадку прототипів *protobuf* в технології *gRPC*), або отримувати його на початку кожного сеансу окремим повідомленням. Реалізація як першого, так і другого варіанту передачі структурної інформації не має бути складною. Головне питання полягає в тому, як отримати структурну інформацію для усіх можливих повідомлень у системі.

Оскільки, як було зазначено вище, варіант ручного опису програмістом усіх структур даних вимагає великих витрат часу, доцільним виглядає розробка способу, в якому збір структурної інформації буде виконуватися автоматично. Очевидно, що такий аналіз повідомлень повинен виконуватися попередньо. Маючи деяку множину зразків повідомлень, можна виділити з них необхідну структурну інформацію окремим алгоритмом та зберегти її. Використовуючи цю збережену структурну інформацію, можна лише посилатись на неї, а не передавати її в повному обсязі для кожного окремого повідомлення, і, таким чином, зменшити надлишковість інформації в повідомленнях.

Автори пропонують спосіб оптимізації передачі однотипних JSON-документів комп'ютерними мережами, який складається з двох етапів: етапу попереднього аналізу та етапу кодування/декодування даних.

Розглянемо спочатку **загальну ідею** цього нового способу оптимізації, що полягає у компактному кодуванні JSON-повідомлень.

Маючи попередньо зібрану структурну інформацію для конкретних повідомлень, потрібно знайти спосіб відрізнити одні повідомлення від інших, розділити повідомлення на окремі класи. Оскільки формат JSON має динамічну типізацію, прив'язуватися до конкретного переліку полів конкретного типу виглядає недоцільним. В окремих випадках деякі поля можуть бути відсутні або мати інший тип значення (наприклад, масив замість об'єкту), і, в результаті, перелік усіх можливих класів повідомлень буде занадто великий. Тому було вирішено зберігати структурну інформацію не на рівні об'єктів, а на рівні полів. Розглянемо цю загальну ідею на прикладі такого JSON-документа:

```
{
  "foo": 1234,
  "bar": {
    "a": 5678,
    "b": "abc"
  }
}
```

На першому етапі способу (етапі попереднього аналізу) алгоритм має зібрати структурну інформацію з повідомлення. Для цього прикладу вона може виглядати наступним чином:

Запис_1: інформація про поле на рівні кореневого об'єкту: назва — *foo*, тип значення — *число*.

Запис_2: інформація про поле на рівні кореневого об'єкту: назва — *bar*, тип значення — *об'єкт*.

Запис_2.1: інформація про поле на рівні об'єкту *bar*: назва — *a*, тип значення — *число*.

Запис_2.2: інформація про поле на рівні об'єкту *bar*: назва — *b*, тип значення — *рядок*.

Тут «*Запис_1*», «*Запис_2*» і т.д. позначають записи з таблиці структурної інформації. Ця таблиця має бути остаточно сформована на етапі попереднього аналізу. Використовуючи посилання на записи у цій таблиці, на етапі кодування вищенаведений JSON-документ може бути записаний наступним чином:

```
Запис_1: 1234
Запис_2:
Запис_2.1: 5678
Запис_2.2: abc
Кінець файлу
```

Оскільки при такому кодуванні присутнє посилання на запис об'єкту з поля *bar* («Запис_2»), ми маємо можливість додатково спростити закодований документ, ввівши контекстно-залежні посилання та термінальний символ для об'єкту. Наприклад:

```
Запис_1: 1234
Запис_2:
    Запис_1: 5678
    Запис_2: abc
Кінець об'єкту
Кінець файлу
```

Програма, що зчитує повідомлення, зустрівши посилання на запис для об'єкту, переходить в контекст записів цього конкретного об'єкту, а після обробки термінального символу контекст замінюється на попередній. Тим самим ми досягаємо значного зменшення кількості можливих варіантів для посилань на записи й, відповідно, скорочення самих посилань.

Розглянемо тепер детально алгоритм збору необхідної структурної інформації, тобто **алгоритм попереднього аналізу** запропонованого способу.

Оскільки структурна інформація має бути розподілена по окремих контекстах, кожен з яких відповідає за конкретний об'єкт, доцільним буде представлення цієї інформації у вигляді дерева. Корінь цього дерева відповідає за кореневий об'єкт повідомлення та зберігає інформацію щодо усіх можливих полів та їх типів. Якщо значення поля є атомарним, то зберігається лише список можливих атомарних типів для значень цього поля. Якщо значення є неатомарним, то додатково зберігається посилання на наступний вузол дерева, що відноситься до відповідного об'єкта або масива. Таке дерево показане на рисунку 1.

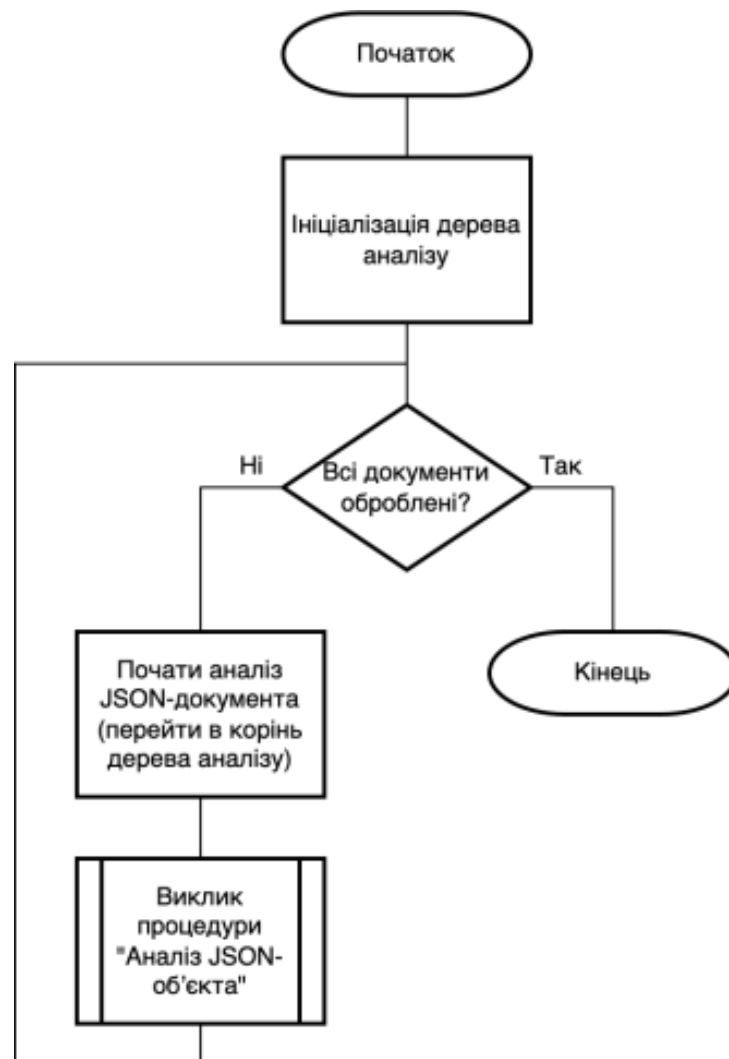


Рис. 1. Алгоритм попереднього аналізу: загальна частина

Це дерево ініціалізується один раз для усіх повідомлень, що оброблюються, а обробка кожного окремого JSON-документа починається з кореня (кореневого контексту). При обробці конкретного об'єкта або масива відбувається перехід у контекст відповідного поля й рекурсивно викликається відповідна процедура (рис. 2 та рис. 3).

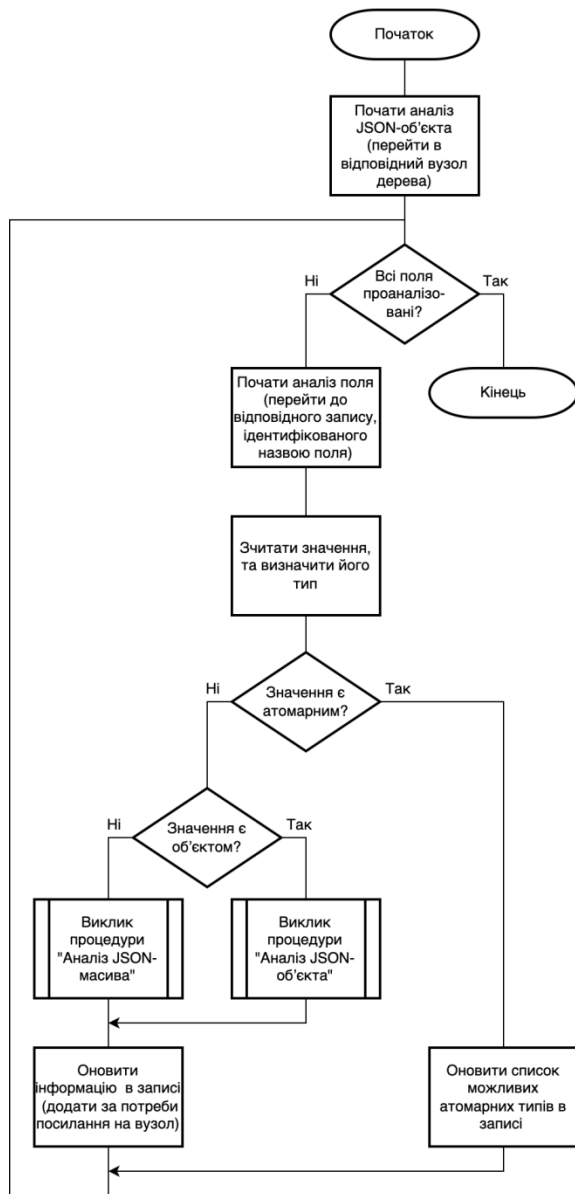


Рис. 2. Алгоритм попереднього аналізу: процедура «Аналіз JSON-об'єкта»

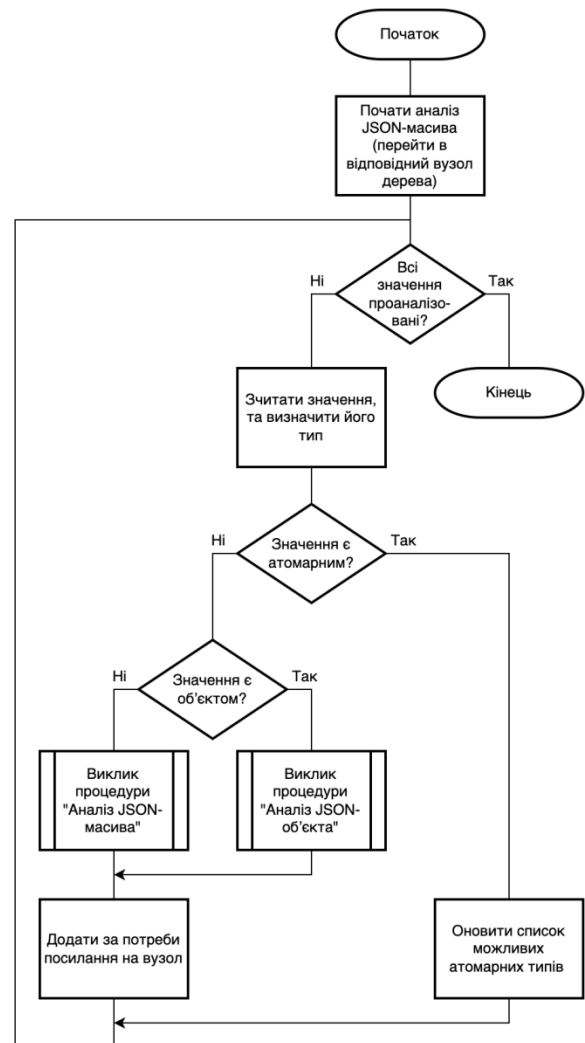


Рис. 3. Алгоритм попереднього аналізу: процедура «Аналіз JSON-масива»

Таким чином, маємо всю необхідну структурну інформацію. Після цього залишається зберегти її у зручному вигляді та згенерувати короткі унікальні ідентифікатори кожного поля для наступного етапу. Збережену структурну інформацію з проставленими ідентифікаторами надалі будемо називати **схемою повідомлень**.

Розглянемо тепер алгоритми кодування та декодування повідомлень.

Алгоритм кодування повідомлень є занадто складним для графічного представлення або детального опису в рамках цієї статті. Тому, опишемо його текстом в узагальненому вигляді.

1. Зчитування JSON-файлу виконується послідовно, а схема повідомлень зберігається в пам'яті.
2. Кореневий об'єкт ніяким чином не ідентифікується.
3. Послідовно зчитуються поля та елементи масиву.

4. Якщо значення поля не є атомарним, то контекст схеми перемикається на відповідний вузол.

5. Якщо назва поля та тип значення, що зчитуються, співпадають з даними у схемі, то записується байтовий ідентифікатор поля зі схеми, а після нього — *компактне детерміноване представлення відповідного значення*.

6. Якщо назва поля співпадає зі схемою, але тип значення інший, то записується байтовий ідентифікатор поля зі схеми, використовується *префікс зміни типу*, вказується конкретний тип значення й записується саме значення.

7. Якщо зчитане поле в схемі відсутнє, то записується *спеціальне зарезервоване значення ідентифікатора, що вказує на невідомий запис*. Після цього записується повна назва поля в рядковому вигляді, потім записується тип значення і тільки після цього — саме значення. Будемо називати такий запис поля *динамічним* (від “динамічна типізація”).

8. Якщо об'єкт записується *динамічним* способом, то усі його поля записуються тим же самим способом.

9. При записі елементів масиву записується тип значення елемента, а після цього саме значення. Якщо значення є неатомарним — відповідно, перемикаємо контекст схеми на відповідний вузол.

Алгоритм декодування повідомлень є алгоритмом, зворотнім до попереднього. Опишемо його також узагальнено.

1. Документ зчитується послідовно, схема повідомлень зберігається в пам'яті.

2. Якщо зустрічається ідентифікатор поля, то зчитується інформація зі схеми та декодується значення *відповідним детермінованим способом в залежності від типу*.

3. Якщо значення не є атомарним, то переходимо до відповідного вузла дерева схеми.

4. Враховується *префікс заміни типу* у разі його наявності.

5. Якщо зустрічається *зарезервоване значення ідентифікатора*, то зчитується назва поля, тип та значення без вказування схеми. Зчитується *динамічний* запис.

6. Масив зчитується аналогічно до об'єкта.

В описах обох алгоритмів було зазначено, що атомарні значення записуються так званим “спеціальним детермінованим способом”. Згідно цього способу рядкове значення записується як послідовність байтів (кодів символів у використовуваній кодовій таблиці), числове значення — як послідовність байтів самого числа, а null-значення записується послідовністю байтів нульової довжини, тобто взагалі не записується, оскільки інформації про тип вже достатньо для його ідентифікації.

Як зазначено в пункті 7 першого алгоритму та пункті 5 другого, **алгоритми кодування та декодування здатні коректно оброблювати дані з невідомою структурою**. Це і є основною відмінністю запропонованого способу від статично типізованих аналогів формату JSON. Звісно, що у разі даних з невідомою структурою використовується динамічне представлення, що є менш компактним (бо включає в себе структурну інформацію), але важливо, що таке представлення не містить додаткової інформації порівняно зі звичайним JSON-форматом. У загальному випадку застосування запропонованого способу завжди зменшує розмір повідомлення, а ступінь цього зменшення визначається тим, наскільки повними були дані на етапі попереднього аналізу. Якщо на цьому етапі програма не отримала деякі зразки документів, то алгоритм кодування буде менш ефективним саме для цих документів.

Висновки. В даній статті проаналізовані основні властивості існуючих способів передачі однотипних JSON-повідомлень та був запропонований новий спосіб, який відрізняється від існуючих врахуванням структури документів, що передаються, і використанням формату даних, який не містить структурну інформацію (структуру об'єктів, імена полів), та, в результаті, дозволяє оптимізувати передачу JSON-документів за обсягом, зменшуючи мережевий трафік та не вимагаючи явного опису програмістом структур даних повідомлень.

Запропонований спосіб поєднує у собі одночасно і статичну, і динамічну типізацію даних, що передаються. Якщо повідомлення або його частина мають одну з вже відомих структур, то використовується компактний бінарний формат без структурної інформації, аналогічний до розглянутих статично типізованих форматів, наприклад, protobuf [10]. Якщо повідомлення або його частина мають невідому структуру, то використовується менш компактний бінарний формат зі структурною інформацією, але який не поступається компактністю відомим бінарним аналогам формату JSON (BSON [3], Binary Ion [4] та інші).

Основна новизна запропонованого способу полягає в наявності етапу попереднього аналізу, оскільки саме на цьому етапі відбувається врахування структури документів, що передаються. Аналізуючи зразки повідомлень, формується схема повідомлень, яка дозволяє більшою чи меншою мірою використовувати саме перший спосіб кодування даних, який є більш компактним. Чим більше різних зразків повідомлень надано на етапі попереднього аналізу, тим більш компактним буде результуюче повідомлення, і, відповідно, тим краща буде загальна ефективність передачі однотипних повідомлень в цілому.

Описаний спосіб має гарні перспективи подальшого розвитку, оскільки в результуючому бінарному форматі може бути використаний будь-який підходящий спосіб представлення ідентифікаторів та будь-який підходящий спосіб представлення атомарних значень, а також не має обмежень на додавання нових типів даних (що може бути актуальним для, наприклад, представлення дат чи специфічних структур типу `ObjectId` [3]). Крім того, запропонований спосіб може використовуватись у поєднанні з потоковою архівацією, що може бути ефективним у разі повідомлень великого розміру.

Список бібліографічного опису

1. The JavaScript Object Notation (JSON) Data Interchange Format [Електронний ресурс] – Режим доступу до ресурсу: <https://www.rfc-editor.org/rfc/rfc8259>.
2. Erik Wilde, Cesare Pautasso. REST: From Research to Practice. — Springer Science & Business Media, 2011. — 528 р. — ISBN 978-1-4419-8303-9.
3. BSON (binary JSON): Specification [Електронний ресурс] – Режим доступу до ресурсу: <https://bsonspec.org/spec.html>.
4. Ion Binary Encoding [Електронний ресурс] – Режим доступу до ресурсу: <https://amzn.github.io/ion-docs/docs/binary.html>.
5. Concise Binary Object Representation (CBOR) [Електронний ресурс] – Режим доступу до ресурсу: <https://www.rfc-editor.org/rfc/rfc8949.html>.
6. MessagePack specification [Електронний ресурс] – Режим доступу до ресурсу: <https://github.com/msgpack/msgpack/blob/master/spec.md>.
7. Comparison of JSON Like Serializations – JSON vs UBJSON vs MessagePack vs CBOR [Електронний ресурс] – Режим доступу до ресурсу: <http://zderadicka.eu/comparison-of-json-like-serializations-json-vs-ubjson-vs-messagepack-vs-cbor/>.
8. GZIP file format specification version 4.3 [Електронний ресурс] – Режим доступу до ресурсу: <https://www.rfc-editor.org/rfc/rfc1952>.
9. RFC 7932: Brotli Compressed Data Format [Електронний ресурс] – Режим доступу до ресурсу: <https://www.rfc-editor.org/rfc/rfc7932>.
10. Protocol Buffers Version 3 Language Specification | Google Developers [Електронний ресурс] – Режим доступу до ресурсу: <https://developers.google.com/protocol-buffers/docs/reference/proto3-spec>.
11. Apache Thrift - Home [Електронний ресурс] – Режим доступу до ресурсу: <https://thrift.apache.org/>.

References

1. “The JavaScript Object Notation (JSON) Data Interchange Format” *Internet Engineering Task Force (IETF)*, <https://www.rfc-editor.org/rfc/rfc8259>. Accessed 30 Nov. 2022
2. Erik Wilde, Cesare Pautasso. REST: From Research to Practice. — Springer Science & Business Media, 2011. — 528 р. — ISBN 978-1-4419-8303-9.
3. “BSON (binary JSON): Specification” *bsonspec.org*, <https://bsonspec.org/spec.html>. Accessed 30 Nov. 2022
4. “Ion Binary Encoding” *Amazon, Amazon.com Inc.*, <https://amzn.github.io/ion-docs/docs/binary.html>. Accessed 30 Nov. 2022
5. “Concise Binary Object Representation (CBOR)” *Internet Engineering Task Force (IETF)*, <https://www.rfc-editor.org/rfc/rfc8949.html>. Accessed 30 Nov. 2022
6. “MessagePack specification” *Sadayuki Furuhashi*, <https://github.com/msgpack/msgpack/blob/master/spec.md>. Accessed 30 Nov. 2022
7. “Comparison of JSON Like Serializations – JSON vs UBJSON vs MessagePack vs CBOR” *Ivan Zderadicka*, <http://zderadicka.eu/comparison-of-json-like-serializations-json-vs-ubjson-vs-messagepack-vs-cbor/>. Accessed 30 Nov. 2022
8. “GZIP file format specification version 4.3” *Internet Engineering Task Force (IETF)*, <https://www.rfc-editor.org/rfc/rfc1952>. Accessed 30 Nov. 2022
9. “RFC 7932: Brotli Compressed Data Format” *Internet Engineering Task Force (IETF)*, <https://www.rfc-editor.org/rfc/rfc7932>. Accessed 30 Nov. 2022
10. “Protocol Buffers Version 3 Language Specification” *Google Developers*, <https://developers.google.com/protocol-buffers/docs/reference/proto3-spec>. Accessed 30 Nov. 2022
11. “Apache Thrift — Home” *Apache Software Foundation*, <https://thrift.apache.org/>. Accessed 30 Nov. 2022

DOI: <https://doi.org/10.36910/6775-2524-0560-2022-49-02>

УДК 004.056

Тогоєв Олексій Романович, аспірант.

<https://orcid.org/0000-0003-3465-7767>

Чорноморський національний університет імені Петра Могили, м. Миколаїв, Україна

МЕТОД ДЕАНОНІМІЗАЦІЇ КОРИСТУВАЧІВ IOS ЧЕРЕЗ ПРОТОКОЛ AIRDROP

Тогоєв О.Р. Метод деанонізації користувачів iOS через протокол Airdrop. У статті запропоновано алгоритм відстеження для пристроїв iOS, які використовують динамічно рандомізовані MAC-адреси та Advertising-повідомлення під час зв'язку за стандартом BLE.

Ключові слова: iOS, BLE, рандомізована MAC-адреса, Advertising-повідомлення, визначення місцезнаходження смартфона

Tohoiev O.R. A method for deanonymizing iOS users through the Airdrop protocol. The paper proposes a tracking algorithm for iOS devices using dynamically randomized MAC addresses and BLE advertisements.

Keywords: iOS, BLE, randomized MAC address, Advertising messages, definition of smartphone location

Постановка наукової проблеми.

Технологія Bluetooth сприяла поширенню миттєвого бездротового підключення, починаючи від особистих підключених аксесуарів, закінчуючи розумними будинками, локалізованим та персоналізованим досвідом покупок на основі розташування. З моменту своєї першої появи в мобільних телефонах в 2000 році, Bluetooth зазнав п'яти основних специфікацій, редакцій з численними поправками [1].

У ранніх версіях специфікації Bluetooth постійна Bluetooth MAC-адреса пристроїв регулярно трансливалася у відкритому вигляді, що викликало серйозні побоювання з приводу конфіденційності через можливість небажаного відстеження. Це було вирішено у базовій специфікації Bluetooth 4.0 із введенням стандарту Bluetooth Low Energy (BLE), також відомого як Bluetooth Smart. BLE дозволяє виробникам пристроїв використовувати тимчасові випадкові адреси бездротового зв'язку замість їх постійної адреси для запобігання відстеження [2]. Однак ці функції анонімізації визначаються таким чином, що залишають виробникам певний ступінь гнучкості. Необов'язковість таких функцій захисту конфіденційності має особливе значення, оскільки стандарт BLE був розроблений спеціально для підтримки пристроїв з низьким енергоспоживанням, таких як смарт-годинник та інші пристрої, що носяться, і є привабливою метою для ворожого відстеження своїх користувачів.

Пристрої BLE транслиють так звані "Advertising" на незашифровані загальнодоступні канали, щоб сигналізувати про свою присутність іншим пристроям [3]. Термін "advertising" з англійської перекладається як "рекламний". Однак це не є точним технічним описом процесу. Застосовують також і термін "широкомовні пакети". Він також не точний, але загально зрозумілий. З метою дотримання максимальної точності, на наш погляд, можна уживати термін без перекладу.

Описана вище публічна трансляція містить всю необхідну інформацію для виконання функцій пристрою, не допускаючи витоку непотрібної особистої інформації про пристрій або його користувача. У деяких випадках, однак, пристрої можуть передавати дані, які розкривають конфіденційну інформацію про них самих або навіть про інші пристрої. Дослідженню таких випадків присвячена публікація.

Аналіз досліджень.

Проблеми з конфіденційністю та безпекою Bluetooth акцентуються та вивчаються дослідниками та практиками з моменту його появи. Анонімізація пристроїв у загальнодоступних каналах зв'язку стала доступною лише з впровадженням BLE у Bluetooth 4.0. Багато досліджень ефективності рандомізації MAC-адрес зосереджені на Wi-Fi, де вирішення проблеми конфіденційності аналогічне та існує проблема ширококомовних постійних ідентифікаторів.

Однак, вирішення проблеми вразливості, яке пропонується для Wi-Fi, нелегко перенести на Bluetooth, оскільки вони засновані на різних технологіях та сферах і залежить від мережевого стека Wi-Fi. Тому відстеження Bluetooth та Wi-Fi здійснюється методами та утилітам, специфічним для BLE.

Існує кілька методів прослуховування зв'язку Bluetooth 2.0 з використанням Bluetooth-сніфера на основі GNU Radio та програмно-визначеного радіообладнання USRP [4]. Такі підходи

засновані на перехопленні пакетів і реінжинірингу всіх параметрів, необхідних для підслуховування зв'язку засобами Bluetooth. Проте, висновки, що стосуються реалізації Bluetooth 2.0, втратили актуальність з появою BLE та Bluetooth 5.

У 2015 році М. І. Jameel та J. Dungen представили бібліотеку з відкритим вихідним кодом для сканування Bluetooth Low Energy (LE) та активними ширококомовними пакетами RFID [5]. Їхня робота узагальнює різні доступні протоколи Beacon, які використовують ширококомовні протоколи на основі близькості та дозволяють всі види локалізованих взаємодій зі смартфонами та іншими пристроями Bluetooth через ширококомовні канали BLE. Крім того, автори опублікували бібліотеку під назвою advlib, яка обробляє необроблені Advertising-повідомлення BLE і декодує їх у формат відкритих даних. Ця бібліотека дозволяє розробникам програмного забезпечення легко інтегрувати Advertising-функції BLE в їх програмне забезпечення без необхідності декодувати вручну низькорівневі протоколи. Бібліотека також підтримує "колаборативний репозиторій" з відкритим вихідним кодом Sniffypedia, який представляє велику кількість загальновідомих Advertising-ідентифікаторів BLE у зручному для пошуку та доступному форматі. Ця платформа може допомогти класифікувати класи пристроїв Bluetooth для розвідувальних цілей, але пристрій не пропонує можливості відстеження.

Використовуються також методи отримання доступу до постійних MAC-адрес шляхом використання запитів у Wi-Fi [6]. Такі алгоритми залежать від часових характеристик та знайдених порядкових номерів у запитах перевірки Wi-Fi для ідентифікації пристроїв незалежно від своєї MAC-адреси. Також вони описують варіант так званої "кармічної атаки", використовуючи той факт, що багато пристроїв надаватимуть інформацію нібито відомим та довіреним мережам шляхом створення універсального доступу. Часто подання своєї постійної MAC-адреси відбувається у довіреному контексті.

Незважаючи на наявність рандомізації MAC-адрес із збереженням конфіденційності в Bluetooth 4.0 LE, не всі пристрої використовують цю функціональність і, отже, уразливі для відстеження. Крім того, зловмисне розповсюдження відповідного програмного забезпечення для відстеження серед кількох мобільних пристроїв – "BLE Botnet" – розширює можливості відстеження далеко за межами локальної дальності передачі звичайного Bluetooth-зв'язку.

Рандомізація MAC-адрес часто зазнає невдачі через неправильну або послідовну реалізацію, частково засновану на попередніх підходах. Крім того, недоліки рандомізації адрес Android-телефонів полягають у тому, що можливо зробити висновок про типи пристроїв через префікси адрес, а також їх глобальні MAC-адреси через інформацію, знайдену у ширококомовних атрибутах WPS. Також атака контрольного кадру ефективно розкриває постійну MAC-адресу Wi-Fi пристрою.

Виділення раніше невирішених питань.

Незважаючи на наявність ґрунтовних розробок, недостатньо дослідженим залишається поєднання конфіденційності BLE з поняттям потенційно глобального ботнету, яке розширює загрозу конфіденційності з боку локального виявлення присутності для відстеження місцезнаходження цільових користувачів серед великої кількості пристроїв.

У той час, як більшість наведених досліджень зосереджені на визначенні пристроїв, які використовують статичні адреси, доцільно зупинитися на відстеженні пристроїв, які використовують динамічно рандомізовані адреси. Ці адреси змінюються залежно від параметрів регенерації, встановлених виробниками. Підхід, запропонований нижче, не отримує інформацію через звернення хешей або інших методів зворотного проектування, а використовує повністю пасивне сніф-«вивчення» для отримання інформації, яка дозволить відстеження пристрою. Метод заснований на вилученні ідентифікуючих токенів з корисного навантаження Advertising-повідомлень та використовує певні функції Bluetooth. Цим він відрізняється від описаних іншими дослідниками.

Формулювання мети і завдань дослідження.

Метою статті є висвітлення можливостей деанонізації користувачів за допомогою використання вразливості Airdrop-технології обміну файлами між пристроями Apple без підключення до Інтернету – та демонстрація того, що така вразливість стосується всіх девайсів на платформі iOS.

Методи.

Спочатку необхідно виявити різні типи Advertising-пакетів та показати так звані ідентифікаційні токени, які є унікальними для пристроїв і залишаються незмінними протягом

достатньо тривалого часу, щоб слугувати вторинними ідентифікаторам. Також розроблено онлайн-алгоритм, який використовує незмінність ідентифікаційних токенів та випадкових адрес, щоб синхронно, постійно перевіряти пристрій, незважаючи на впровадження заходів анонімізації. Такий підхід може бути застосований до усіх Windows 10, iOS та пристроїв MacOS. Алгоритм не вимагає розшифровки повідомлень або порушення безпеки Bluetooth. Він повністю заснований на загальнодоступному незашифрованому Advertising-трафіку.

Виклад основного матеріалу й обґрунтування отриманих результатів дослідження.

AirDrop-технологія дозволяє обмінюватися файлами між пристроями Apple без підключення до Інтернету. AirDrop використовує Bluetooth для створення однорангової мережі Wi-Fi між пристроями. При цьому повинні бути включені і Wi-Fi, і Bluetooth. Відстань між пристроями має не перевищувати 9 метрів. Коли AirDrop включений, він запускає Bluetooth для пошуку інших пристроїв iOS з підтримкою AirDrop, що знаходяться поблизу. В момент активування AirDrop, пристрій починає відправляти Advertising-пакети BLE.

Advertising-пакети BLE є одним із найважливіших аспектів Bluetooth Low Energy. Вони допомагають знизити енергоспоживання, пришвидшити з'єднання та підвищити надійність. BLE Advertisements також є ключем до маяків, які допомагають визначити місцезнаходження та здійснювати відстеження.

Bluetooth Low Energy або також Bluetooth Smart має два способи зв'язку. Перший — це використання advertising-повідомлень, коли периферійний пристрій BLE транслює пакети на всі пристрої навколо нього. Пристрій-одержувач може діяти на основі цієї інформації або підключитися, щоб отримати додаткову інформацію. Отже, перший спосіб — це принцип маяка — проста передача пакетів. Другий спосіб зв'язку полягає в отриманні пакетів за допомогою з'єднання, де як периферійний, так і центральний пристрій надсилають пакети. Проаналізуємо, як працюють Advertising-повідомлення.

1. Користувач не може встановити з'єднання між двома пристроями без використання Advertising-повідомлення.

2. Велика кількість продуктів BLE «спить» більшу частину часу, прокидаючись лише для Advertising-повідомлень та підключення, коли це необхідно.

3. Користувачам потрібні адаптивні продукти, а інтервал транслювання Advertising-повідомлень має вирішальне значення для швидких з'єднань.

4. Advertising-оголошення є основою маяків iBeacon, EddyStone та інших, тому вони використовуються постійно.

Advertising-оголошення за задумом односпрямовані. Центральний пристрій не може надсилати дані на периферійний пристрій без підключення.

В кожному Advertising-пакеті є такі дані користувача:

- номер телефона;
- AppleId;
- Email;
- OS;
- ім'я пристрою;
- інше.

Ці конфіденційні дані Advertising-пакеті захешовані зазвичай за допомогою SHA-256 (рис. 1).

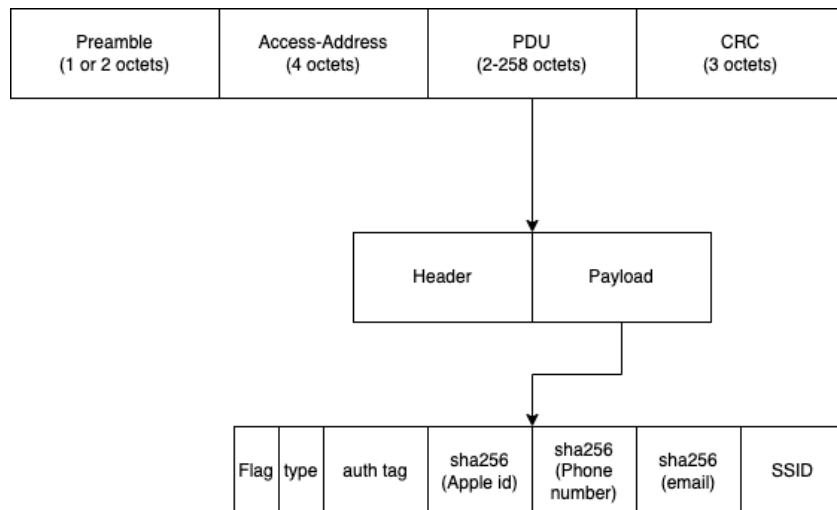


Рис. 1. Структура Advertising-пакета

Продемонструємо роботу розробленого алгоритму на прикладі визначення користувача мобільної системи України. База даних, що зберігає інформацію про кожного користувача мережі стільникового зв'язку, має аббревіатуру HLR (англ. *The Home Location Register*). HLR зберігає інформацію про статус послуг, що пов'язані з певною SIM-карткою. Кожна картка має унікальний номер, IMSI, що використовується як ключ для HLR. В цій базі зберігається наступна інформація про абонента:

- номер телефону (MSISDN) абонента, що пов'язаний з даною SIM-карткою;
- GSM-сервіси, що доступні користувачеві;
- місце знаходження користувача (VLR та SGSN);
- налаштування переадресації;

Відомо, що всі номери в Україні починаються з +38, таким чином, з 13 символів залишиться 10. Але також відомо, що всі оператори використовують маски, та їх всього 15 (табл. 1).

Таблиця 1. Маски мобільних операторів України

Оператор	Маска
Київстар	039, 067, 068, 096, 097, 098
МТС	050, 066, 095, 099
life:)	063, 093
Utel	091
PEOPLEnet	092
Інтертелеком	094

На основі викладеного вище, алгоритм ідентифікації користувача може виглядати наступним чином:

1. Згенерувати номери телефонів.
2. Захешувати сгенеровані номери телефонів.
3. Порівняти перебором всі захешовані номери телефонів.
4. Відправити HLR-запит на встановлення валідності номеру і локації.

Нижче наведений приклад застосування запропонованого алгоритму.

Припустимо, що в Advertising-пакеті був номер +380634376494. Тоді хеш його буде cc3f08c8f31ba94d1677ced94e992c00979314f43783620a9f3f0561543d7e12.

Отже, в Advertising-пакеті передавались **cc3f08**.

Для отримання усіх можливих хешів можна використати формулу (1):

$$p = u + m + r, \quad (1)$$

де u – це код України (+38);
 m – маска оператора;
 r – рандомні 7 цифр.

Таким чином можливо отримати 149'999'985 комбінацій. Для генерування і хешування у наведеному прикладі було витрачено біля 8 хвилин.

У згенерованому масиві номерів лише у 5 номерів хеш починається з «cc3f08» (табл.2).

Таблиця 2. Номери телефонів та їх хеші

Хеш	Номер телефону
cc3f084a1b3a342406d41531ff7b08d6a486322ecb8d48696d3c0e6d8f7f132a	+380671117394
cc3f0869a06859793951d31d2becd8f93c9a032dc67185f52424c826c422540c	+380966020098
cc3f08b465c5bb829abbb70bf2f5c1f6ad404f4ea4e6687edc10b18da6a2d73a	+380506610974
cc3f08c8f31ba94d1677ced94e992c00979314f43783620a9f3f0561543d7e12	+380634376494
cc3f080861729f449c63d55cfc841ebd1f239fa5035c19b1af07f837b99ba52f	+380948919817

Наступним кроком алгоритму є порівняння отриманих номерів із даними оператора HLR (табл. 3).

Таблиця 3. Результати порівняння згенерованих номерів та даних оператора HLR

Телефон	MCCMNC	Оператор	Статус
380671117394	25502 (Київстар)	Україна, Київстар, Миколаївська обл.	Доставлено
380966020098	25502 (Київстар)	Україна, Київстар, Кіровоградська обл.	Неможливо доставити, абонент не існує
380506610974	25501 (МТС)	Україна, МТС, Закарпатська обл.	Передано оператору
380634376494	25506 (lifecell)	Україна, lifecell, Київська обл.	Доставлено
380948919817	25505 (Інтертелеком)	Україна, Інтертелеком, Київська обл.	Неможливо доставити, абонент не існує

Отже, якщо відомо, в якому місті перебувають користувачі, можливо точно визначити, хто саме з них потрібний.

Таким чином, запропонований алгоритм, час роботи якого не перевищує 8 хвилин, може бути ефективно використаний для визначення місцеположення користувачів.

Висновки та перспективи подальшого дослідження.

Специфікація Bluetooth 5 розширює корисний діапазон зв'язку до сотень метрів у прямій видимості. У той час, як атака з відстеженням, запропонована в цій статті, розглядає відстеження одного противника в такій радіусі дій, можливо припустити, що локальне відстеження методів BLE може бути значно ускладнено за рахунок їх координатії у ботнеті зловмисників. Зазначений підхід має значний потенціал у сфері спостереження.

Список бібліографічного опису

1. Padiya S. D., Gulhane V. S. Analysis of Bluetooth versions (4.0, 4.2, 5, 5.1, and 5.2) for IoT applications. In book: Implementing Data Analytics and Architectures for Next Generation Wireless Communications. Chapter: 10. IGI Global, 2021. P. 153-178. DOI: 10.4018/978-1-7998-6988-7.ch010.

2. Cha S.-C., Yeh K.-H., Chen J.-F. Toward a robust security paradigm for Bluetooth low energy-based smart objects in the Internet-of-Things. *Sensors*. 2017. Vol. 17, no. 2348. P. 1-17. DOI:10.3390/s17102348.
3. Gangwal A., Singh S., Spolaor R., Srivastava A. BLE Whisperer: Exploiting BLE Advertisements for data exfiltration. In book: *Computer Security. ESORICS, 2022*. DOI: 10.1007/978-3-031-17140-6_34.
4. Kim T.-Yo., Lee H.-J. Vulnerability analysis of Bluetooth communication based on GNU radio. *The Journal of the Korean Institute of Information and Communication Engineering*. 2016. Vol. 20, Is. 11. P. 2014-2020. DOI: 10.6109/jkiice.2016.20.11.2014.
5. Jameel I., Dungen J. Low-power wireless advertising software library for distributed M2M and contextual IoT. 2015 IEEE 2nd World Forum on Internet of Things (WF-IoT). Jan. 2015. P. 597-602. DOI: 10.1109/WF-IoT.2015.7389121.
6. Vanhoef M. A time-memory trade-off attack on WPA3's SAE-PK. *ACM Asia Conference on Computer and Communications Security (ASIA CCS '22)*, May 2022. P. 27-37. DOI:10.1145/3494105.3526235.

References

1. Padiya S. D., Gulhane V. S. Analysis of Bluetooth versions (4.0, 4.2, 5, 5.1, and 5.2) for IoT applications. In book: *Implementing Data Analytics and Architectures for Next Generation Wireless Communications*. Chapter: 10. IGI Global, 2021. P. 153-178. DOI: 10.4018/978-1-7998-6988-7.ch010.
2. Cha S.-C., Yeh K.-H., Chen J.-F. Toward a robust security paradigm for Bluetooth low energy-based smart objects in the Internet-of-Things. *Sensors*. 2017. Vol. 17, no. 2348. P. 1-17. DOI:10.3390/s17102348.
3. Gangwal A., Singh S., Spolaor R., Srivastava A. BLE Whisperer: Exploiting BLE Advertisements for data exfiltration. In book: *Computer Security. ESORICS, 2022*. DOI: 10.1007/978-3-031-17140-6_34.
4. Kim T.-Yo., Lee H.-J. Vulnerability analysis of Bluetooth communication based on GNU radio. *The Journal of the Korean Institute of Information and Communication Engineering*. 2016. Vol. 20, Is. 11. P. 2014-2020. DOI: 10.6109/jkiice.2016.20.11.2014.
5. Jameel I., Dungen J. Low-power wireless advertising software library for distributed M2M and contextual IoT. 2015 IEEE 2nd World Forum on Internet of Things (WF-IoT). Jan. 2015. P. 597-602. DOI: 10.1109/WF-IoT.2015.7389121.
6. Vanhoef M. A time-memory trade-off attack on WPA3's SAE-PK. *ACM Asia Conference on Computer and Communications Security (ASIA CCS '22)*, May 2022. P. 27-37. DOI:10.1145/3494105.3526235

DOI: <https://doi.org/10.36910/6775-2524-0560-2022-49-03>

УДК 004.94:656.02

Багнюк Наталія Володимирівна, к.т.н., доцент

<https://orcid.org/0000-0002-7120-5455>

Бортник Катерина Яківна, к.т.н., доцент

<https://orcid.org/0000-0001-5282-099X>

Варченко Леонід Леонідович, магістр

Луцький національний технічний університет, м.Луцьк, Україна

МОНІТОРИНГОВА СИСТЕМА ДЛЯ ОПЕРАЦІЙНОЇ СИСТЕМИ WINDOWS

Багнюк Н.В., Бортник К.Я., Варченко Л.Л. Моніторингова система для операційної системи Windows. У даній статті розглядається створення моніторингової системи для операційної системи Windows та специфіка її використання.

Ключові слова: Windows, Grafana, моніторингова система, операційна система, сервіс.

Bahniuk N.V., Bortnyk K.Y., Varchenko L.L. Monitoring system with Windows operating system. This article discusses about monitoring Windows operating system and specific things to solve in process.

Keywords: Windows, Grafana, monitoring system, operating system, service.

Постановка проблеми та аналіз досліджень. В даній статті описано створений альтернативний програмний комплекс для відслідковування та аналізування ресурсів, які задіяні під час роботи операційної системи Windows. Всі антивіруси ґрунтуються суто на хеш-сумах виконуваних файлів, в той же як моніторингова система надасть можливість проаналізувати поведінку зловмисного програмного забезпечення або будь-якого іншого.

Так як Windows є найпоширенішою операційною системою, то питання її безпеки є критично важливим. Велика популярність даної системи пояснює кількість різних зловмисних програм, що орієнтуються не тільки на безпосередньо пряму атаку та викрадення, а й приховування своєї присутності для нанесення якнайбільшої шкоди, збитків для користувача. Тому превентивні методи визначення шкідливого ПЗ є актуальною темою на даний час та в майбутньому.

Виділення невіршених раніше частин загальної проблеми

Створюваний програмний комплекс орієнтований на клієнт-серверну архітектуру, а саме - створення інтерфейсу для програм та кінцевих користувачів. Причини вибору такого принципу є подальша можливість розробити масштабовану систему не тільки для одного юзера, а й цілої системи для моніторингу групи пов'язаних між собою систем. Для систем типу UNIX більшість шкідливих програм користуються принципом неавторизованого доступу до системи. Тобто, для них притаманне використання доступу через програми або "дірки" в коді, які поки що існують в системі. В Windows ситуація зводиться до комплексів, які не тільки намагаються отримати неавторизований доступ через програми, а намагаються використати безпосередньо користувача як "винуватця" зараження. Тим і ця тема є цікава для розгляду. Тому антивірусні програми часто безсилі – користувач самостійно надає програмі адміністративний доступ, що і є причиною його проникнення. [4]

Формулювання мети дослідження

Одним з найважливіших етапів у розробці ПЗ є, власне, опрацювання структури програми, визначення архітектурних рішень і т.д. Основною ідеєю під час етапу проектування є розробка системи, яка легко масштабується, легко тестується, зберігає можливість як серверного так і локального розташування.

Для операційної системи Windows не так легко встановити моніторингову систему через деякі особливості роботи, тому для розробки використовуються Prometheus, Node Exporter і Grafana.

Виклад основного матеріалу дослідження

Prometheus – головна частина. Prometheus збирає показники з кількох служб і отримує їх в одному місці.

Проста програма під назвою Node Exporter збирає показники операційної системи та пропонує до них HTTP-доступ. Prometheus збирає дані з одного або кількох екземплярів Node Exporter.

Grafana представляє інформаційні панелі з графіками та діаграмами, які містять дані з Prometheus.

Для забезпечення вищезазначених параметрів використовується віртуалізація програм в різні процеси. Для цього використовується різні докер-контейнери, які будуть запущені для кожних компонентів програми як унікальний контейнер (рис. 1).

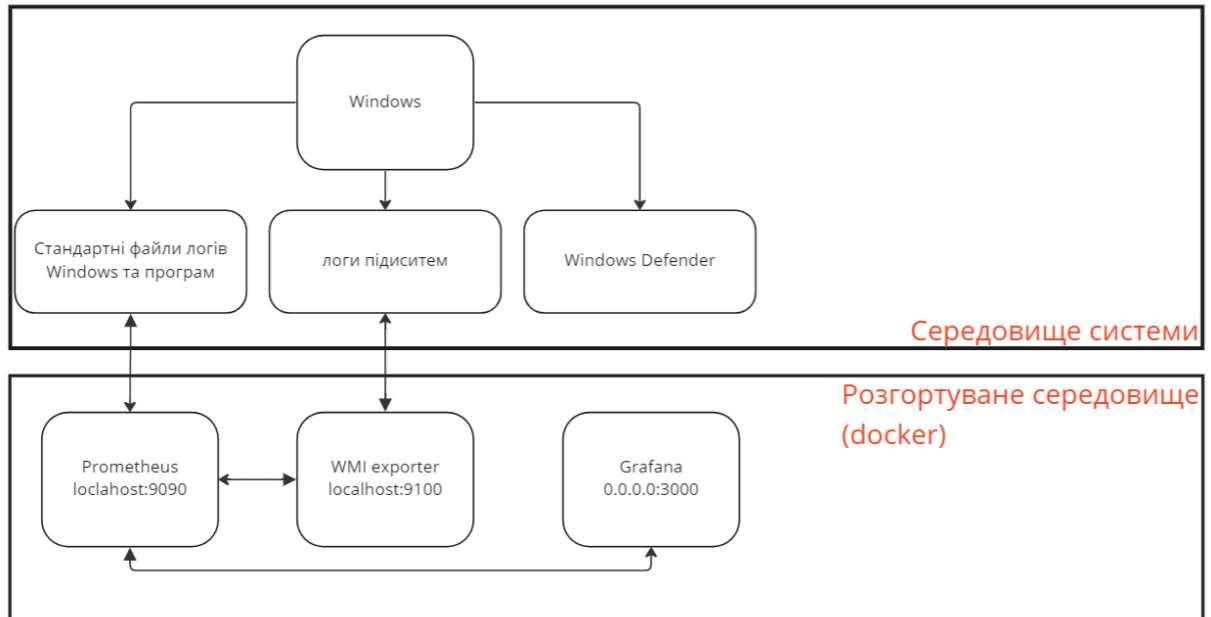


Рисунок 1 – Структурна схема програми

З рисунка 1 видно загальні налаштування сервісів. Кожен контейнер має свої налаштування, різні версії програм, "образи" та попередні команди у вигляді скрипт-файлів для підготовки контейнеру до роботи.

Prometheus. Образ "prom/prometheus" при установці "закидає" в контейнер файл prometheus.yml та використовує локальну папку prometheus_data спільно з хостом. При запуску використовує певні підготовчі команди та відкриває в мережі порт 9090.

Grafana. Завдяки команді depends_on, образ "grafana/grafana" під час встановлення чекає запуску контейнеру з prometheus. Після цього в контейнер надсилає файли налаштувань з папки grafana/provisioning/ та використовує локальну папку grafana_storage спільно з хостом. При запуску використовує певні підготовчі команди та відкриває на хості порт 3000.

Розглянемо структурну схему програми (рис. 1), де зображено взаємозв'язки між сервісами.

Для зручної розробки ПЗ моніторингу, інструменти віртуалізації є необхідністю. У випадку розробки цим інструментом є docker. Docker – спеціальне середовище віртуалізації. "Середовищем" виступають невеликі або повноцінні операційні системи, які запускаються в середині віртуалізації, та можуть бути пов'язані між собою за допомогою внутрішньої, недоступної ззовні мережі. Такий підхід надає декілька переваг: відслідковування роботи програми, побудова клієнт-серверної системи, можливість швидкого запуску програм, копіювання усіх логів та дій програм, обмеження ресурсів споживання.

Prometheus. ПЗ, що записує показники в режимі реального часу в базу даних часових рядів (що забезпечує високу розмірність), створену з використанням моделі HTTP pull, із гнучкими запитом та сповіщеннями в реальному часі.

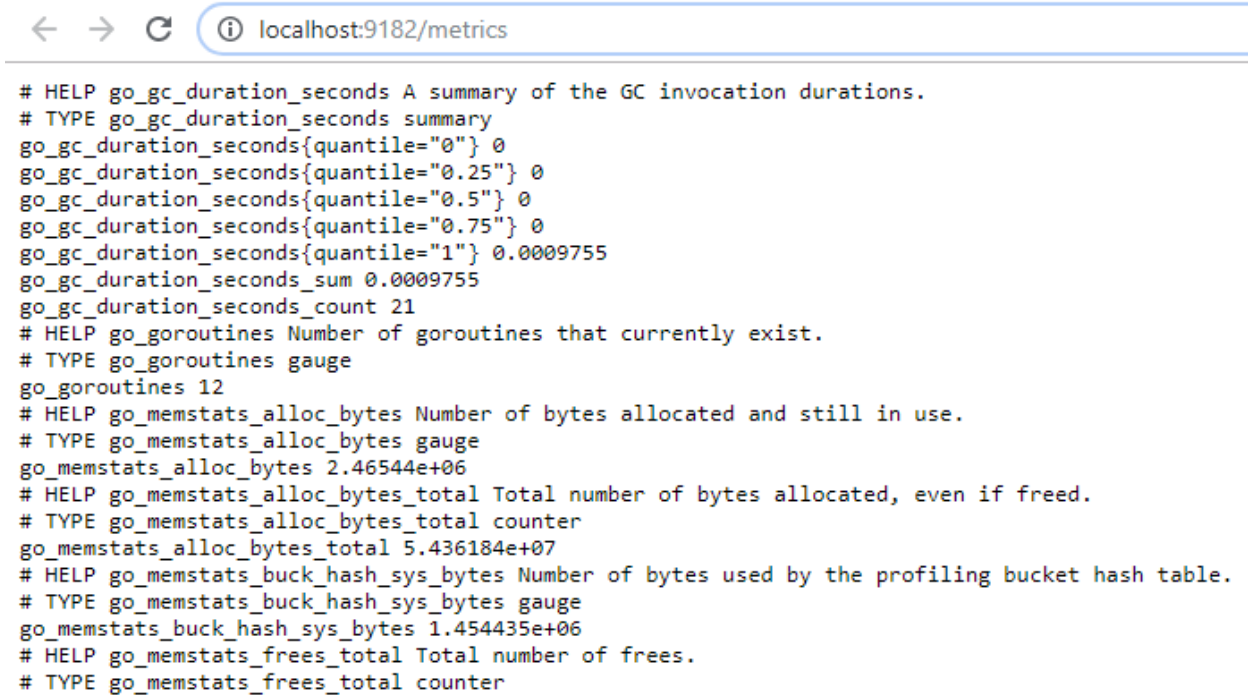
Проект написаний на Go та ліцензований за ліцензією Apache 2, з вихідним кодом, доступним на GitHub, і є випускним проектом Cloud Native Computing Foundation разом з Kubernetes і Envoy. Prometheus збирає та зберігає свої метрики як дані часових рядів, що означає, що інформація про показники зберігається разом із міткою часу, коли вона була отримана, і не обов'язковими парами ключ-значення, відомими як мітки. [1] Добре працює для запису будь-яких чисто числових часових рядів. Він підходить як для машинного моніторингу, так і для моніторингу високодинамічних сервіс-орієнтованих архітектур. У світі мікросервісів його підтримка збору багатовимірних даних і запитів є особливою перевагою. [5]

Одного інструменту тільки для збору даних мало, додатковим, але не обов'язковим, інструментом є візуалізатор даних. Для означених раніше цілей найкраще підходить інструмент візуалізації Grafana.

Програмним забезпеченням з відкритим кодом Grafana можна запитувати, переглядати, сповіщати та аналізувати свої показники, журнали та трасування незалежно від того, де вони зберігаються.

Також можна використовувати інструменти, надані Grafana OSS, щоб створювати графіки та візуалізації з даних у базі даних часових рядів (рис. 2). Після налаштування першої інформаційної панелі залежно від потреб, є широкий вибір альтернатив, коли вперше використовується Grafana. Можна створити список відтворення, наприклад, щоб відстежувати інформацію про погоду та статистику щодо розумного будинку. Можна налаштувати надання та автентифікацію, якщо адміністратор підприємства керує Grafana для різних команд. [3]

Допоміжну роль виконує сервіс експортер WMI (рис. 3). Цілі – це вузли, до яких Prometheus має доступ і які показують метрики за певною URL-адресою. Такі цілі постачаються як «експортери», які, насправді є двійковими файлами, які працюють з ціллю та відповідають за збір показників про хост. Запуск «Node Exporter», який відповідає за отримання статистичних даних про використання ЦП або дискового вводу/виводу, який зараз використовується – це спосіб моніторингу системи Linux. Експортер WMI відповідатиме за збір статистики про систему та працюватиме як служба Windows. [6]



```
# HELP go_gc_duration_seconds A summary of the GC invocation durations.
# TYPE go_gc_duration_seconds summary
go_gc_duration_seconds{quantile="0"} 0
go_gc_duration_seconds{quantile="0.25"} 0
go_gc_duration_seconds{quantile="0.5"} 0
go_gc_duration_seconds{quantile="0.75"} 0
go_gc_duration_seconds{quantile="1"} 0.0009755
go_gc_duration_seconds_sum 0.0009755
go_gc_duration_seconds_count 21
# HELP go_goroutines Number of goroutines that currently exist.
# TYPE go_goroutines gauge
go_goroutines 12
# HELP go_memstats_alloc_bytes Number of bytes allocated and still in use.
# TYPE go_memstats_alloc_bytes gauge
go_memstats_alloc_bytes 2.46544e+06
# HELP go_memstats_alloc_bytes_total Total number of bytes allocated, even if freed.
# TYPE go_memstats_alloc_bytes_total counter
go_memstats_alloc_bytes_total 5.436184e+07
# HELP go_memstats_buck_hash_sys_bytes Number of bytes used by the profiling bucket hash table.
# TYPE go_memstats_buck_hash_sys_bytes gauge
go_memstats_buck_hash_sys_bytes 1.454435e+06
# HELP go_memstats_frees_total Total number of frees.
# TYPE go_memstats_frees_total counter
```

Рисунок 3 – Wmi exporter

Таке поєднання програм дозволить забезпечити не тільки зручність у роботі, а й можливу масштабованість проекту. Масштабованість забезпечена за рахунок мультиопераційності програм та легкості їх поєднання. Означені технології розробки дозволять якнайефективніше розподілити розробку самої моніторингової системи та визначити отримані результати вже на ранніх етапах розробки.

Перш ніж переходити до інструментів Prometheus, важливо зрозуміти цю модель даних повністю. У Prometheus використовуються пари ключ-значення. Значення зберігає фактичне значення як число, тоді як ключ визначає, що ми вимірюємо. Prometheus не призначений для зберігання неструктурованих даних, таких як звичайний текст. Він зберігає вимірювання, які були зібрані протягом усього часу.

Звідси маємо перший і наголовніший маркер – як змінюється використання ресурсів операційною системою різними програмами під час роботи в середовищі, де існує зловмисна програма. Визначення “зміна використання ресурсів” ґрунтується на припущенні, що під час зараження комп’ютер (представлений фізично) та операційна система (представлена в значенні ядра) гарантовано повинні змінити кількість ресурсів, які будуть використовуватися до аномальної межі – це є маркер фізичного стану. За даним припущенням спостерігає дашборд Win_phys_state та Win_stats. [8]

Win_phys_state – дашборд Grafana, призначений для забору фізичних метрик споживання ресурсів. В даному відображенні міститься інформація суто про стан фізичних властивостей пристроїв, які входять до даного комп’ютера, такі як:

- для процесора: температура центрального процесора, частота роботи, кількість зайнятих фізичних ядер і тд;
- оперативна пам’ять: кількість завантаженої пам’яті, швидкість;
- диск: тип, швидкість читання/писання, середня швидкість відгуку.

Для даного дашборда, в сервісі Win Exporter використовуються спеціальні колектори. Для кожного роду даних існують спеціальні колектори, відповідно до вище описаних властивостей.

Win_stats призначений для відслідкування системних метрик, які будуть важливим маркером не тільки для визначення зловмисного ПЗ, а й при відслідкуванні стану операційної системи, як такої. Метрики, що входять до даного дашборду:

- за процесами: час роботи системи, загальна кількість процесів, кількість дескрипторів, кількість відкритих потоків;
- використання диску: графік активного часу, графік обміну даних, коефіцієнт середньої завантаженості;
- використання мережі різними процесами в момент часу.



Рисунок 4 – Дашборд Win_phys_state

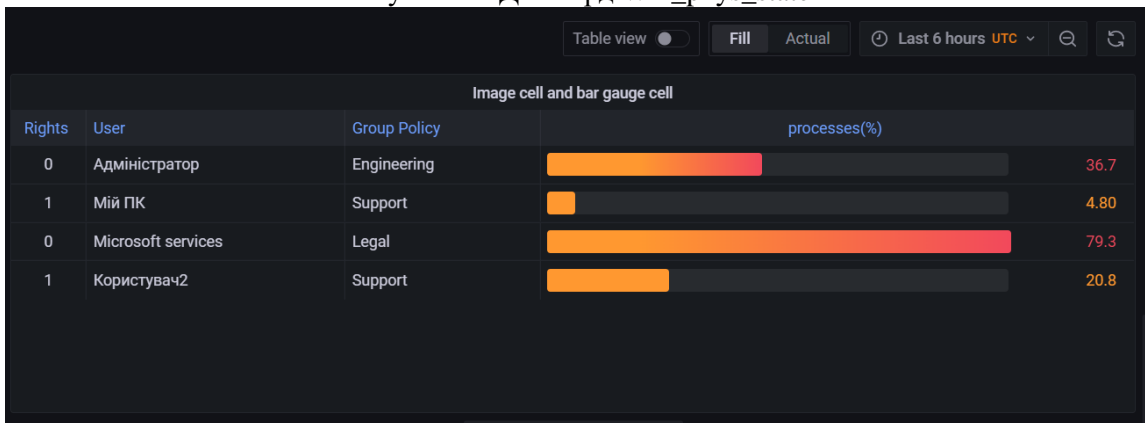


Рисунок 5 – Метрики запуску користувача (дашборд Win_user_info)

На основі даного дашборду отримуємо загальну інформацію про ресурси, які можуть бути задіяні системою, і чітко можна отримати інформацію, чи дійсно надлишок ресурсів використовуються в даний момент. Даний дашборд є загальною інформацією про стан процесів та відповідних служб в момент часу.

Після визначення першого маркера потрібно зосередити увагу на конкретних маркерах поведінки саме операційної системи та запущених програм. В даному випадку основними метриками є: споживання ресурсів програмами (Prog_user_stats), споживання підсистеми Windows (Win_sys_stats), ресурси Windows програм (Win_exe_stats) та користувачі і політики безпеки (Win_user_info). Дані метрики супроводжують використання ресурсів процесами. З даного дашборду можна зробити висновок про аномальну поведінку окремого процесу, відірваного від загальної системи Windows.

Під час запуску процесів, належних до користувачів, ми отримуємо можливість прослідкувати за викликаним процесом, отримати ресурси, які він використовує (дочірні процеси та їхні привілеї). Далі можемо прослідкувати сервіси, які використовує процес (так як при виклику сервісу ми отримуємо те,

що один з пов'язаних процесів почав використовувати відповідні сервіси). І в кінці кінців це і становить перехід з дашборду Win_user_info до Win_exe_stats і згодом в Win_sys_stats.

Дашборд Win_sys_stats – комплекс відслідковування сервісів та підсистеми Windows. Даний інтерфейс є по суті списком сервісів та підсистем з графіком використання процесами. Даний графік надасть результати про надмірне споживання будь-яким процесом ресурсів ядра через сервіси, на графіку це буде чітко видно. Також вартий уваги дашборд Win_TCP_IP, призначений для проглядання споживання мережі процесами. Даний дашборд надає просту інфографіку про загальне користування та процеси на фоні загальної. [7]

Для визначення та дослідження ресурсів безперечно важливим є грамотне розподілення всіх метрик рівномірно по всім дашбордам. Звісно, що для повної інформативності метрик неможливо не повторюватися, але це тільки заради реального моніторингу системи. Основною задачею є дослідження ресурсів операційної системи Windows, але в проміжному результаті очікується насамперед рівномірний розвиток системи, адже проміжні результати будуть надавати все більше і більше інформації, яку необхідно враховувати в подальших дашбордах. Такий розвиток буде точно виділений, причиною цього однозначно буде складність операційної системи.

Дана моніторингова система має можливість отримати в майбутньому наступний маркер – функцію моніторингу стану обладнання та операційної системи, забезпечення інформування про аномальну зміну в роботі обладнання та рекомендацій щодо його обслуговування. Також постійно відслідковуються окремі процеси Windows, які пов'язані з оновленням та запуском деяких служб, список яких можна отримати з панелі управління службами або в реєстрі (regedit). Ці сервіси та служби є вже не такі пріоритетні як ті, що вказані на автозапуск, тому під час запуску системи вона деякий час "підвисає" - додаткові системні служби, служби та сервіси перевірок починають виконувати звичайну системну роботу. І різниця між цими службами тільки в тому, що деякі мають звичайний системний характер, а саме – якщо це перезапуск, ці служби не будуть проводити ці процедури.

Щодо оцінки працездатності моніторингової системи – основне можна побачити з дашборду Monitor_health_host (рис. 6). На ньому видно, що частка ресурсу, яким користується програма хост є дуже мала, навіть для комп'ютерної системи з обмеженим ресурсом.



Рисунок 6 – Моніторинг стану сервісів, які складають моніторинг-систему

Висновки та перспективи подальших досліджень

Даний програмний комплекс надає змогу отримувати інформацію про стан операційної системи. Поданий у статті порядок програм забезпечує всі покладені на них задачі та витрачається мала кількість ресурсів, що є важливим для ефективної роботи системи, яку піддають моніторингу. Дашборди, створені для даного програмного забезпечення, повністю відображають дійсний стан ресурсів операційної системи.

Дану розробку можна використовувати не тільки для виявлення шкідливого програмного забезпечення, а й для моніторингу системи. Тобто, дані результати також можуть допомогти отримати результати щодо технічного стану апаратної частини ПК. Також на основі цих даних отримано результати щодо стану операційної системи: швидкість відгуку програм, коректність роботи ядра ОС, стан системних файлів.

Подальші дослідження по цій темі дозволять створити прецедент для наступних досліджень в даній області. Всі дослідження в даній темі дадуть розвиток програмних комплексів, які будуть займатися виявленням зловмисного програмного забезпечення в атоматизованій системі моніторингу.

Список бібліографічного опису

1. Bastos J. Hands-On Infrastructure Monitoring with Prometheus: Implement and scale queries, dashboards, and alerting across machines and containers / J. Bastos, P. Araújo.
2. Docker Desktop | Docker Documentation [Електронний ресурс]. – 2022. – Режим доступу до ресурсу: <https://docs.docker.com/desktop/>.
3. Documentation | Grafana Labs [Електронний ресурс]. – 2022. – Режим доступу до ресурсу: <https://grafana.com/docs/>.
4. Microsoft security documentation - Security documentation [Електронний ресурс] – Режим доступу до ресурсу: <https://learn.microsoft.com/uk-ua/security/>.
5. Overview | Prometheus [Електронний ресурс]. – 2022. – Режим доступу до ресурсу: <https://prometheus.io/docs/introduction/overview/>.
6. prometheus-community/windows_exporter: Prometheus exporter for Windows machines [Електронний ресурс] – Режим доступу до ресурсу: https://github.com/prometheus-community/windows_exporter#readme.
7. Russinovich M., Margosis A. Troubleshooting with the Windows Sysinternals tools. Redmond : Microsoft, 2015.
8. Salituro E. Learn Grafana 7.0: A beginner's guide to getting well versed in analytics, interactive dashboards, and monitoring / Eric Salituro.. – (1st Edition, Kindle Edition).

References

1. Bastos J. Hands-On Infrastructure Monitoring with Prometheus: Implement and scale queries, dashboards, and alerting across machines and containers / J. Bastos, P. Araújo.
2. Docker Desktop | Docker Documentation [Electronic resource]. – 2022. – URL: <https://docs.docker.com/desktop/>.
3. Documentation | Grafana Labs [Electronic resource]. – 2022. – URL: <https://grafana.com/docs/>.
4. Microsoft security documentation - Security documentation [Electronic resource]. – 2022. – URL: <https://learn.microsoft.com/uk-ua/security/>.
5. Overview | Prometheus [Electronic resource]. – 2022. – URL: <https://prometheus.io/docs/introduction/overview/>.
6. prometheus-community/windows_exporter: Prometheus exporter for Windows machines [Electronic resource]. – 2022. – URL: https://github.com/prometheus-community/windows_exporter#readme.
7. Russinovich M., Margosis A. Troubleshooting with the Windows Sysinternals tools. Redmond : Microsoft, 2015.
8. Salituro E. Learn Grafana 7.0: A beginner's guide to getting well versed in analytics, interactive dashboards, and monitoring / Eric Salituro.. – (1st Edition, Kindle Edition).

DOI: <https://doi.org/10.36910/6775-2524-0560-2022-49-04>

УДК 621.867

Гаврильченко Олександр Віталійович, к.т.н., професор

Корендій Віталій Михайлович, к.т.н., доцент,

<https://orcid.org/0000-0002-6025-3013>

Шенбор Владислав Станіславович, ведучий інженер

<https://orcid.org/0000-0002-3530-8975>

Національний університет "Львівська політехніка", м. Львів, Україна

ВПЛИВ ПАРАМЕТРІВ ВІБРАЦІЙНОГО КОНВЕЄРА НА ШВИДКІСТЬ ТРАНСПОРТУВАННЯ ДЕТАЛЕЙ

Гаврильченко О.В., Корендій В.М., Шенбор В.С. Вплив параметрів вібраційного конвеєра на швидкість транспортування деталей. Розглянуто один з найбільш ефективних способів підвищення швидкості транспортування деталей та виробів вібраційними конвеєрами, нанесенням на транспортуючу поверхню ворсового покриття. Наведені теоретичні та експериментальні визначення коефіцієнта швидкості транспортування, як найбільш показову характеристику ефективності вібраційного конвеєра.

Ключові слова: вібраційний конвеєр, швидкість транспортування, коефіцієнт швидкості, ворсове покриття, режими транспортування.

Gavrilenko O.V., Korendii V.M., Shenbor V.S. The influence of the parameters of the vibrating conveyor on speed of transportation of parts. One of the most effective ways to increase the speed of transporting parts and products by vibrating conveyors, by applying a pile coating on the transporting surface, is considered. Theoretical and experimental definitions of the transportation speed coefficient are given as the most indicative characteristic of the vibratory conveyor efficiency.

Key words: vibrating conveyor, transportation speed, speed coefficient, pile coating, transportation modes.

Постановка проблеми. Вібраційні конвеєри одержали широке застосування у внутрішньо-цеховому та міжопераційному транспорті оброблюваного та складального обладнання різних галузей виробництва завдяки ряду відомим перевагам, які спонукають до їх удосконалення, основними напрямками, яких є збільшення швидкості транспортування, зниження потужності та маси. Недоліками таких конвеєрів є значна маса, яка припадає на одиницю амплітуди коливань робочого органу, залежність від навантаження, а також мала амплітуда коливань, яка обмежена допустимим значенням повітряного проміжку між якорем та магнітопроводом електромагнітного віброзбудника.

Проведені дослідження з використанням ворсового покриття були зведені до застосування конструкцій вібраційних конвеєрів, які використовуються для транспортування виробів направленими та незалежними коливаннями [1,2,3,6]. Такий підхід не дозволяє використати переваги застосування ворсового покриття, а саме – використання лише одного віброзбудника та розроблення конвеєрів значної довжини. Переваги конвеєрів з лише поздовжніми коливаннями розглянуті в [1,2,3].

У роботі розглядається вібраційні конвеєри є електромагнітним силовим збудженням коливань, основними перевагами якого є простота регулювання амплітуди коливань і можливість її регулювання під час роботи конвеєра, що дозволяє здійснювати автоматичне регулювання продуктивності під час роботи.

Аналіз останніх досліджень і публікацій. Найбільше розповсюдження одержали двомасові вібраційні конвеєри з еліптичними коливаннями робочого органу, що забезпечується незалежним збудженням поздовжніх та нормальних коливань з заданим зсувом фаз між ними, за таких умов транспортування здійснюються у безвідривних режимах, а також вібраційні конвеєри з направленими коливаннями.

Однак, найвищою швидкістю транспортування було досягнуто при застосуванні ворсового покриття на транспортуючій поверхні. Переважно пропонують використовувати ті ж самі конструкції вібраційних конвеєрів з направленими та незалежними коливаннями, однак раціональніше застосовувати конвеєри лише з поздовжніми коливаннями.

Існує спосіб суттєво підвищити швидкість транспортування за допомогою анізотропної робочою транспортуючої поверхні утвореною короткою щіткою з полімерних матеріалів нахиленою під певним кутом у напрямку транспортування. Використання анізотропного покриття

дозволяє значно підвищити швидкість транспортування за рахунок значної різниці коефіцієнта тертя виробу у напрямку транспортування та у зворотному.

У даній роботі досліджуються швидкість транспортування деталей з різних матеріалів на вібраційних конвеєрах, які здійснюють лише поздовжні коливання, що дозволяє уникнути паразитних нормальних коливань, та надає можливість розробляти конвеєри значно більшої довжини [6,7,8,10].

Формулювання мети дослідження. Дослідження проводились з метою одержання результатів, які слугували б одержанню практичних рекомендацій необхідних при інженерному проектуванню промислових зразків швидкісних конвеєрів.

Основні результати дослідження. Експериментальні дослідження, результати яких наведені у даній роботі проводились на вібраційному конвеєрі наведеному на рис.1. Транспортуюча поверхня конвеєра (рис.1) складається з двох мас 1 і 2 встановлених плоскими пружинами 3 на рамі 4, електромагніта 5 та якоря 6 які закріплені між транспортуючими поверхнями (масами) 1,2 між якими розташована пружина 7. На транспортуючій поверхні конвеєра прикріплене ворсове покриття 8.

Розглянемо рух виробів по несучій поверхні розміщеній горизонтально і здійснюючій гармонічні коливання вздовж напрямку руху виробів.

Рівняння руху транспортуючої поверхні конвеєра відносно нерухомої системи координат мають вигляд

$$\frac{d^2x_{\Pi}}{dt^2} = A\omega^2 \cos \omega t$$

де A - амплітуда поздовжніх коливань транспортуючої поверхні ;

ω - кутова частота коливань;

t - час.

Загальне диференціальне рівняння руху деталі в початковий період відносно транспортуючої поверхні в координаті X , яка направлена вздовж транспортування конвеєра

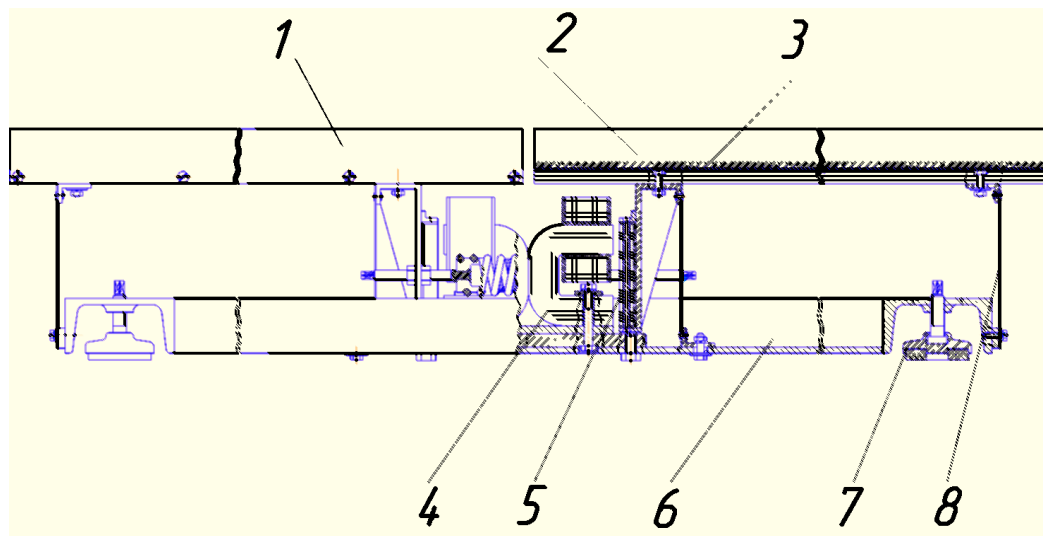


Рис.1. Вібраційний конвеєр з ворсовим покриттям

$$m \frac{d^2x}{dt^2} = -mA\omega^2 \cos \omega t + F$$

де m – маса деталі;

F – сила тертя.

Сили тертя у випадку переміщені деталі та транспортуючої поверхні за напрямком транспортування F_+ та протилежному $-F_-$ визначаються за виразом

$$F_+ = mgf_{+ \text{ або}} \quad F_- = mgf_-$$

Деталі під дією коливань транспортуючої поверхні конвеєра безперервно переміщуються у заданому напрямку, у напрямку нахилу ворсу. Конвеєр здійснює коливання в нерухомій системі

координат, за період одного коливання транспортуюча поверхня півперіода рухається у напрямку транспортування деталей, а на протязі другого півперіода – у протилежному і сповільнює швидкість деталі, гальмує. За рахунок ворсу нахиленого під кутом деталі транспортуються у напрямку його нахилу. Коефіцієнт тертя у напрямку руху f_+ суттєво відрізняється від коефіцієнта тертя у протилежному напрямку f_- , очевидно, що f_- значно перевищує f_+ , що і характеризує анізотропію транспортуючої поверхні конвеєра.

Транспортування виробів здійснюється поверхнею конвеєра і складається з декількох етапів:

- прискорений рух у напрямку руху деталі коли напрямок транспортування співпадає з рухом конвеєра, але немає спільного руху (деталь переміщується відносно конвеєра);
- сповільнений рух деталей коли ці рухи не співпадає, конвеєр рухається у зворотному напрямку деталі;
- сумісний рух деталі та конвеєра деталь не переміщується відносно конвеєра, такий етап можливий лише при певних незначеннях частоти та коливання конвеєра.

У даній роботі розглядається двоетапне переміщення деталей, оскільки переміщення деталей у три і більше етапів можливий при незначній частоті та амплітуді коливань, точно вказати не можливо, бо наявність таких режимів залежить ще й від довжини та кута нахилу ворсу. Рівняння руху деталі описується рівнянням на етапі однонаправленого руху з прискоренням буде мати вигляд

$$\frac{d^2x}{dt^2} = gf_- - A\omega^2 \cos \varphi$$

де φ – фазовий кут, $\varphi = \omega t$.

Аналогічно для випадку коли деталь та транспортуюча поверхня рухаються в протилежних напрямках

$$\frac{d^2x}{dt^2} = -gf_- - A\omega^2 \cos \varphi \quad (1)$$

З кожним циклом швидкість деталі зростає і досягає максимального значення, очевидно, відповідно до коефіцієнтів тертя та параметрів коливання, тобто амплітуди A та частоти ω

Етапи руху деталі чергуються. На початкових етапах деталь переміщується з зростаючою середньою швидкістю, максимум якої завжди менший від максимальної швидкості транспортуючої поверхні конвеєра. Тому критерієм ефективності конвеєра був прийнятий коефіцієнт швидкості $K_{ш}$, який вказує наскільки середня швидкість заготовки наближається до максимальної швидкості лотка

$$K_{ш} = \frac{V}{V_{max}}$$

де V_d - швидкість деталі;

V_{max} - максимальна швидкість транспортуючої поверхні конвеєра.

$$V_d = 2\pi v A K_{ш} \left[\frac{MM}{c} \right]$$

де v – частота коливань в секунду.

На етапі розгону початкова швидкість $V_{п}$ визначається з рівняння (1)

$$V_{п} = \frac{g}{\omega} f_{\leftarrow} \varphi + C$$

Звідси ($\varphi_{п}$ - фазовий кут на початковому етапі):

$$C = V_{п} + \frac{g}{\omega} f_{\leftarrow} \varphi_{п} \quad (2)$$

$$V_p = V_{II} + \frac{lg}{\omega}(\varphi - \varphi_{II})$$

Для етапу гальмування

$$V_{\Gamma} = V_{II} - \frac{lg}{\omega}(\varphi - \varphi_{II})$$

Визначивши швидкості деталі на етапі розгону та гальмуванні розраховується загальна швидкість деталі.

Результати експериментальних досліджень залежності коефіцієнта швидкості від довжини ворсу для різних матеріалів наведені на рис.2. , а залежність від амплітуди поздовжніх коливаний на рис.3.

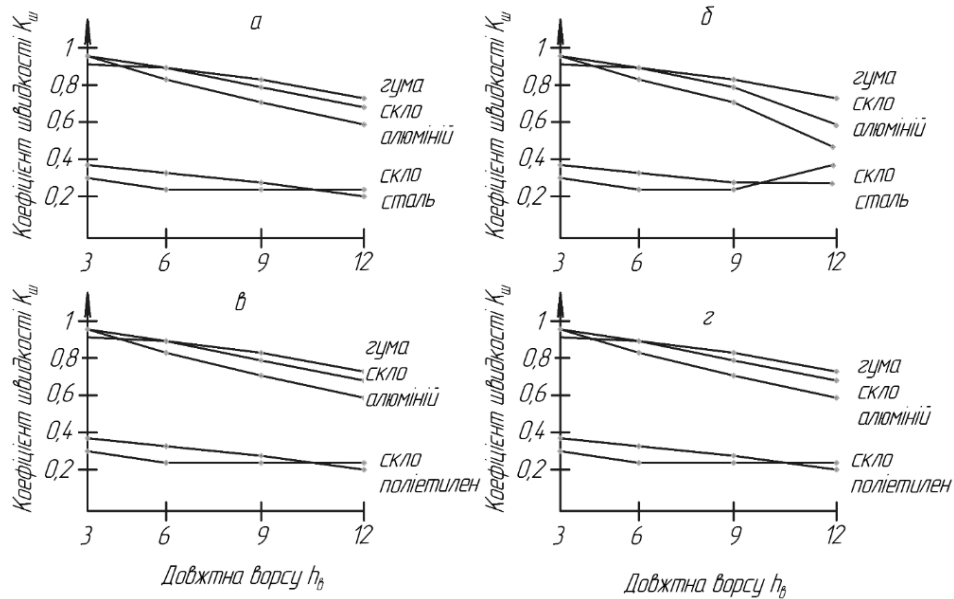


Рис.2. Залежність коефіцієнту швидкості від довжини ворсу для різних матеріалів

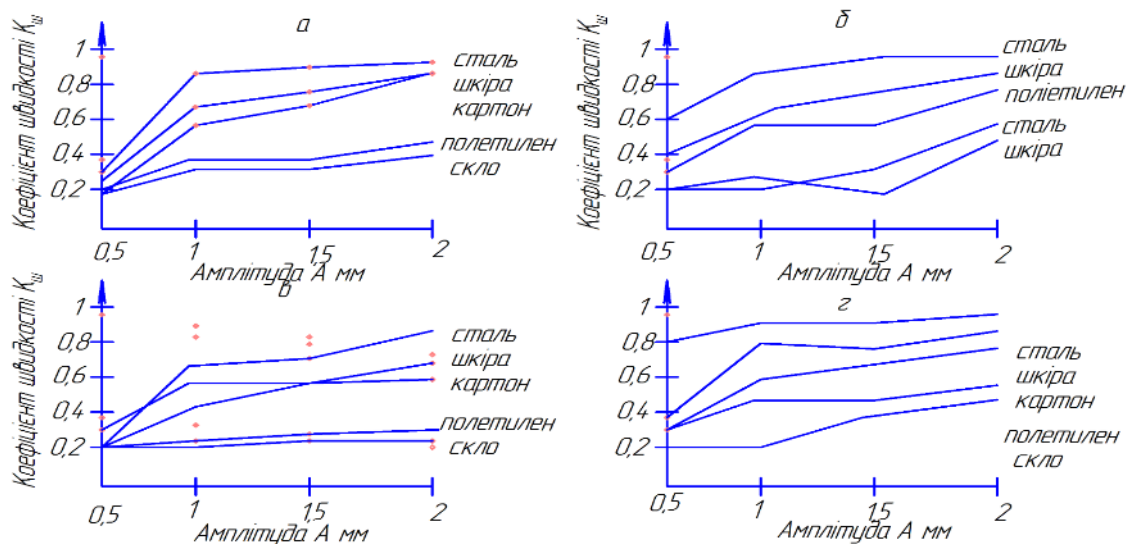


Рис.3. Залежність коефіцієнту швидкості від амплітуди поздовжніх коливаний транспортуючої поверхні

Переважно, при розрахунках для гладких поверхонь транспортуючої поверхні коефіцієнт тертя у прямому та зворотному напрямках приймаються однаковими.

Висновки: проведені дослідження встановили, що використання ворсового покриття на транспортуючій поверхні конвеєра дозволяє суттєво підвищити швидкість транспортування деталей за рахунок простого, низьковартісного засобу.

Список бібліографічного опису

1. Блехман И.И., Джанелидзе Г.Ю. Вибрационное перемещение. М.: Изд-во Наука, 1964. С. 412.
2. Вибрации в технике: Справочник. В 6-ти т., Т. 4. Вибрационные процессы и машины. /Под ред. Э.Э.Лавендела. – Машиностроение. 1981. – С. 509.
3. Врублевський І. Й. Наближені обчислення швидкості вібротранспортування та кута зсуву фаз при еліптичних коливаннях // Вісник Національного університету "Львівська політехніка" "Оптимізація виробничих процесів і технічний контроль у машинобудуванні та приладобудуванні". – 2010. – № 679. – С. 45–48.
4. Дунаевский А.В. Оптимальный синтез параметров вибротранспортирования // Технология судостроения и судоремонта. – Калининград: Калининградский судостроительный институт. 1968. – С. 50-56.
5. Лавендел Э.Э. Синтез оптимальных вибромашин. – Рига: Зинатне. – 1970. С. 211.
6. Нагаев Р.Ф. Периодические режимы вибрационного перемещения. М.: Наука, 1978. С.160.
7. Повідайло В.О.. Вібраційні процеси та обладнання. – Львів: Вид-во Національного університету "Львівська політехніка". 2004. – С. 248.
8. Повідайло В. А., Врублевский И. И. Сравнительный анализ режимов вибротранспортирования штучных изделий при прямолинейных и эллиптических колебаниях // Автоматизация производственных процессов в машиностроении и приборостроении. – Львов: Вища шк., 1983. – Вып. 22. – С. 109-115.
9. Спиваковский А.О., Гончаревич И.Ф. Вибрационные конвейеры, питатели и вспомогательные устройства. – М.: Машиностроение, 1972, - С.328.
10. Сорочак О.З., ВеличкотЛ.Д. Математичне моделювання вібротранспортування виробів анізотропною поверхнею при поздовжніх коливаннях. / Автоматизація виробничих процесів у машинобудуванні та приладобудуванні. Український міжвід. наук.-техн. зб. – Львів: НУ "Львівська політехніка", 2005. – Вип. 39. С.76-82.

References

1. Blehman I.I., Dzhanelidze G.Yu. Vibrational displacement. M.: Publishing house «Nauka», 1964. P. 412.
2. Selections in technology: Reference book. In 6th t., T. 4. Vibrational processes and machines. / Ed. E. E. Lavendela. - Mechanical engineering. 1981. - P. 509.
3. Vrublevsky, I.Y. Approximate calculations of the speed of vibration transport and the angle of phase shift during elliptical oscillations // Bulletin of the National University "Lviv Polytechnic" "Optimization of production processes and technical control in machine building and instrument building". – 2010. – No. 679. – P. 45–48.
4. Dunaevetsky A.V. Optimum synthesis of vibration transport parameters // Technology of shipbuilding and ship repair. – Kaliningrad: Kaliningrad Shipbuilding Institute. 1968. - P. 50-56.
5. Lavendel E.E. Synthesis of optimal vibrating machines. – Riga: Excellent. - 1970. P. 211.
6. Nagaev R.F. Periodic modes of vibrational movement. M.: Nauka, 1978. P.160.
7. Povadaylo V.O.. Vibrational processes and equipment. – Lviv: Branch of the Lviv Polytechnic National University. 2004. - P. 248.
8. V. A. Povydaylo, I. I. Vrublevskii Comparative analysis of modes of vibration transport of artificial products with straight-line and elliptical oscillations // Automation of production processes in machine construction and instrumentation. – Lviv: Higher School, 1983. – Issue 22. – P. 109-115.
9. Spivakovsky A.O., Goncharevich I.F. Vibrating conveyors, feeders and auxiliary devices. - M.: Mashinostroenie, 1972, - P.328.
10. Sorochak O.Z., Velichkot L.D. Mathematical modeling of vibration transport of products by an anisotropic surface during longitudinal oscillations. / Automation of production processes in mechanical engineering and instrument engineering. Ukrainian interdiv. science and technology coll. – Lviv: Lviv Polytechnic University, 2005. – Issue 39. P.76-82.

DOI: <https://doi.org/10.36910/6775-2524-0560-2022-49-05>

УДК 004.4'23

Єрмоленко Денис Вадимович, бакалавр,

<https://orcid.org/0000-0002-5122-4069>

Марченко Олександр Іванович, к.т.н., доцент,

<https://orcid.org/0000-0002-4537-3420>

Національний технічний університет України «Київський політехнічний інститут імені Ігоря Сікорського», м. Київ, Україна

АЛГОРИТМИ СПОСОБУ ПОРІВНЯННЯ ПРОГРАМ, НАПИСАНИХ LISP-ПОДІБНИМИ МОВАМИ, НА ОСНОВІ АБСТРАКТНИХ СЕМАНТИЧНИХ ДЕРЕВ

Єрмоленко Д.В., Марченко О.І. Алгоритми способу порівняння програм, написаних Lisp-подібними мовами, на основі абстрактних семантичних дерев. У даній статті запропоновано два алгоритми виявлення переміщених фрагментів дерев. Ці алгоритми є частинами реалізації способу порівняння програм, написаних Lisp-подібними мовами, на основі абстрактних семантичних дерев. Перший з алгоритмів для виявлення переміщених фрагментів дерев використовує обхід лісу дерев, а другий алгоритм з цією метою використовує хеш-таблицю. Було проведено порівняння швидкодії запропонованих алгоритмів. Зазначено напрямки подальших досліджень щодо покращення цих алгоритмів.

Ключові слова: лisp, s-вираз, абстрактне семантичне дерево, хеш-таблиця, обхід дерева, порівняння дерев.

Yermolenko D., Marchenko O. Algorithms of the technique for comparing programs written in Lisp-like languages based on abstract semantic trees. This article proposes two algorithms for detecting moved tree fragments. These algorithms are part of implementation of the technique for comparing programs written in Lisp-like languages based on abstract semantic trees. The first of the algorithms for detecting moved tree fragments uses traversal of the forest of trees, and the second algorithm uses a hash table. Speed of the proposed algorithms was compared. A way for further research on improving these algorithms is offered as well.

Keywords: lisp, s-expression, abstract semantic tree, hash table, tree traversing, tree comparing.

Постановка проблеми. Кодова база сучасних програмних проєктів містить велику кількість рядків коду, а розробка проєктів виконується з використанням систем контролю версій, таких як git, subversion тощо, де зміни вносяться невеликими частинами коду. Стандартні програми для перегляду змін між конкретними версіями проєкту, такі як git diff, gitk, git GUI, або сайти такі, як gitlab, чи github виконують порівняння файлів різних версій проєкту на текстовому рівні без враховування синтаксису та семантики тих мов програмування, на яких написані конкретні файли проєкту. У свою чергу, порівняння саме на структурно-семантичному рівні, на відмінну від текстового порядкового порівняння, дозволяє відкидати чисто стилістичні зміни й сфокусуватись розробникам на аналізі суттєвих структурних змін у коді, наприклад, додавання, видалення, модифікацію чи переміщення відповідних фрагментів коду. Засоби порівняння на структурному рівні ґрунтуються на способах порівняння абстрактних семантичних дерев, що враховують специфіку саме тих мов програмування, для яких були створені. Такі засоби існують для таких популярних мов програмування як C, C++, Java, C# тощо. Але менш популярні мови програмування, чи мови, що тільки набирають свою популярність таких засобів не мають. Останніми роками популярність функціонального програмування та мов, що підтримують функціональну парадигму програмування швидко зростає, наприклад мови Clojure [1]. Тому дослідження способів порівняння програм, написаних мовами із Lisp-подібним синтаксисом є задачею актуальною. У тезах конференції [2] авторами був запропонований спосіб порівняння абстрактних семантичних дерев програм, написаних Lisp-подібними мовами. Дана стаття є поглибленим та більш детальним описом алгоритмів цього способу.

Термінологія.

АСД – абстрактне семантичне дерево.

Алгоритм пошуку TED (tree edit distance) – це алгоритм пошуку відстані редагування між впорядкованими деревами, що є послідовністю мінімальних за ціною операцій редагування, які перетворюють одне дерево в інше [3].

EBNF (extended Backus-Naur form) – розширена нотація Бекуса Наура, яка є способом запису правил контекстно-вільної граматики.

UML (unified modeling language) – уніфікована мова моделювання, яка є мовою графічного опису для візуалізації, специфікації, конструювання та документування програмних систем.

LCS (longest common subsequence) – найдовша спільна підпоследовність.

S-вираз – є варіантом представлення інформації у вигляді вкладених списків. Поділяються на 2 види: списки (у контексті даної статті lisp-списки) та атоми [4].

Атом – будь-що, що не є lisp-списком. Наприклад числа, символи, рядки тощо. Є термінальними елементами s-виразів.

Lisp-список – це список, чиї елементи, а саме інші ліспові списки чи атоми, записуються зазвичай всередині круглих дужок та розділені між собою пробільними символами.

Віртуальний lisp-список – це lisp-список, який семантично існує в певному контексті, але в текстовому вигляді ніяк не виділений. Класичними прикладами такого lisp-списку є тіло функції, опис аргументів функції чи явне вказання передачі певних ключових аргументів при виклику певної функції.

Def-s-вираз – це s-вираз, який є виразом верхнього рівня та який визначає певний структурний елемент, як: функція, метод, клас, змінна, константа, пакет, макрос тощо.

Емпіричний режим пошуку переміщених піддерев – це режим, коли при збігу додаткової інформації (хешів, кількості вузлів тощо) в кореневих вузлах піддерев, вони вважаються однаковими й не проводиться перевірка на точний їх збіг.

Аналіз останніх досліджень і публікацій.

В роботі [5] представлено дослідження, у якому висловлюються критичні зауваження для застосування класичного варіанту алгоритму пошуку TED у контексті порівняння саме АСД. Було вказано такі його недоліки [5]:

- виконується в кращому випадку за час порядку $O(n^2m^2)$, де n та m це відповідно кількість вузлів у деревах, що порівнюються. Це не дозволяє використовувати його для проектів з дуже великою кількістю рядків коду, де будемо мати АСД десь з десятками тисяч вузлів на один файл;
- операція виявлення “reabeled” (тобто перемаркованого) вузла АСД не враховує семантичного чи синтаксичного сенсу, що може, наприклад, призвести до виявлення факту хибного перемаркування вузла з цілочисельною константою у вузол, що позначає умовний оператор.
- не виконує виявлення переміщення фрагментів дерева (найважливіший недолік).

У роботі [6] показано, що, в загальному випадку, порівняння дерев з операцією знаходження переміщених фрагментів є NP-повною задачею. Через це в роботах, де досліджується порівняння АСД з знаходженням переміщених фрагментів, пропонуються певні обмеження, спрощення чи емпіричні підходи для того, щоб ця задача перестала бути NP-повною. Наприклад в [5] при попередній підготовці двох АСД, які відповідають програмам, написаним на C-подібних мовах, перед етапом порівняння пропонуються виконувати такі операції, як:

- згортання певних груп вузлів дерева, які представляють певні синтаксичні категорії, в один вузол. Наприклад для випадку Java програм, можна згортати вузли для класів, інтерфейсів, визначень, методів, полів чи блоків коду тощо;
- сплющення піддерев;
- видалення з вхідних дерев їх спільних піддерев на основі збігів хешів згорнутих вузлів та сплющених піддерев.

Але досліджень та способів порівняння, які були б спеціально адаптованими для порівняння АСД програм, написаних мовами із Lisp-подібним синтаксисом, авторами знайдено не було.

Постановка завдання.

Як зазначалося вище, авторами був запропонований спосіб порівняння абстрактних семантичних дерев програм, написаних Lisp-подібними мовами [2]. Завданням даної статті є розробка детальних алгоритмів реалізації цього способу та оцінювання їх швидкодії.

Особливості синтаксису Lisp-подібних мов.

Відзначимо важливу особливість синтаксису Lisp-подібних мов, яка полягає у використанні s-виразів. S-вирази є однією із зручних форм запису деревоподібних даних у текстовому вигляді, і їх можна поділити на дві категорії: списки та атоми. Синтаксис запису s-виразів у формі EBNF показано на рисунку 1.

$\langle S\text{-expr} \rangle ::= \langle \text{Atom} \rangle \mid \langle \text{List} \rangle;$
 $\langle \text{List} \rangle ::= '(\langle S\text{-expr} \rangle^*)';$

Рис. 1. Синксист s-виразів у формі EBNF

Важливо зазначити, що нетермінал $\langle \text{Atom} \rangle$ розкривається надалі тільки у термінальні символи s-виразів, і тому атоми будуть термінальними вузлами відповідного йому дерева, а корені піддерев, які у свою чергу представляють списки, будуть нетермінальними вузлами дерева. Треба також зауважити, що пустий список, тобто "()", сприймається як специфічний атом і також буде термінальним вузлом дерева.

Відзначимо також, що при побудові АСД для s-виразів можна взагалі не змінювати форму початкового дерева розбору, з якого будується АСД, окрім випадків виділення віртуальних списків, оскільки такі списки визначаються семантикою тих чи інших s-виразів і їх неможливо виділити на етапі синтаксичного розбору.

Розглянемо коротко суть запропонованого авторами в [2] способу порівняння АСД програм, написаних Lisp-подібними мовами.

Автори пропонують виділяти наступні 3 типи вузлів дерев при порівнянні АСД двох def-s-виразів, def-1 та def-2:

- вузли типу "same" – відповідний вузол є однаковим в def-1 та def-2, тобто не змінив свого положення в дереві;
- вузли типу "unique" – відповідний вузол є унікальним для def-v1 або def-v2, тобто зустрічається тільки в одному із порівнюваних def-s-виразів;
- вузли типу "moved" – відповідний вузол та дерево, коренем якого він є, було переміщено з АСД виразу def-1 до іншої локації АСД виразу def-2.

Запропонований спосіб ґрунтується на використанні цих категорій вузлів і складається із таких етапів:

- підготовка вхідних абстрактних семантичних дерев;
- формування пар def-s-виразів для порівняння;
- виявлення вузлів типу "same" у конкретній парі порівняння;
- виявлення вузлів типів "moved" та "unique".

Опис представлення інформації у вхідних АСД. Ієрархія класів вузлів вхідних АСД показана у вигляді UML діаграми на рисунку 2. У цій ієрархії маємо базовий клас AstNode, що зберігає інформацію, яку мають усі вузли АСД. Базовий клас AstNode має підкласи AtomAstNode та ListAstNode, які відповідно зберігають інформацію про атоми, які є термінальними вузлами, та list-списки, які є нетермінальними вузлами дерев.

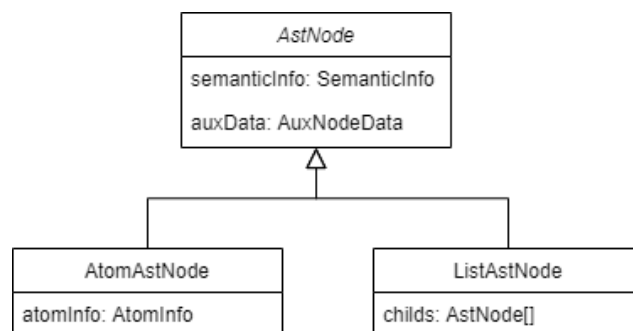


Рис. 2. UML діаграма ієрархії класів вузлів АСД

Поля підкласів AstNode повинні заповнюватися для кожного вузла на етапі підготовки вхідних АСД. На рисунку 3 показано UML діаграму класу AuxNodeData, який містить додаткову інформацію для роботи алгоритмів.

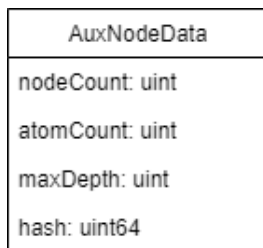


Рис. 3 UML діаграма класу AuxNodeData, який містить додаткову інформацію вузлів АСД

Поле maxDepth цього класу зберігає інформацію про максимальну глибину піддерева, коренем якого є даний вузол, а поле hash зберігає інформацію про хеш цього піддерева.

Алгоритм підготовки вхідних АСД. Для етапу підготовки вхідних АСД розроблено алгоритм prepareAst, псевдокод якого показано на рисунку 4, а алгоритм хешування вузлів дерева був взятий з [7]. Зазначимо, що для даного випадку рекомендується використовувати швидкі функції хешування, а не криптографічні.

```
function prepareAst(treeRoot: AstNode) -> AuxNodeData
    var auxData: AuxNodeData := new AuxNodeData();
    if (treeRoot instanceof AtomAstNode)
        auxData.atomCount := 1;
        auxData.nodeCount := 1;
        auxData.maxDepth := 1;
        auxData.hash := hash(treeRoot.atomInfo) +
            hash(treeRoot.semanticInfo);
    else
        var subTreesAuxData: AuxNodeData[treeRoot.childs.size];
        for ([subtreeRoot: AuxNodeData, i: Integer] in treeRoot.childs)
            subTreesAddData[i] := prepareAst(subtreeRoot);
        auxData += hash(treeRoot.semanticInfo);
        for ([stAddData, i] in subTreesAddData)
            auxData.atomCount += stAddData.atomCount;
            auxData.nodeCount += stAddData.nodeCount;
            auxData.hash += pow(31, i + 1) * auxData.hash;
            auxData.maxDepth := findMaxDepth(subTreesAddData) + 1;
        setAuxData(treeRoot, auxData);
    return auxData;
```

Рис. 4. Псевдокод алгоритму підготовки вхідних АСД

Алгоритм формування пар def-s-виразів для порівняння. Для реалізації наступного етапу запропонованого способу був розроблений алгоритм формування пар def-s-виразів однакових типів для подальшого їх порівняння за їх іменами. Вхідними даними цього алгоритму є списки інформації про def-s-вирази класу Def-s-expr, UML-діаграма якого показана на рисунку 5.

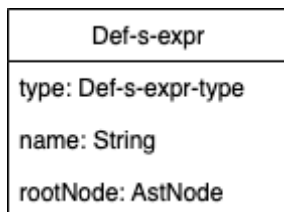


Рис. 5. UML діаграма класу Def-s-expr, який містить інформацію про def-s-вираз

Псевдокод цього алгоритму показано на рисунку 6.

```
type-alias Def-s-expr-pairs = List<Pair<Def-s-expr,Def-s-expr>>

function get-def-s-expr-pairs(def-s-exprs-1 : List<Def-s-expr>,
                             def-s-exprs-2 : List<Def-s-expr>) -> Def-s-expr-pairs
var def-s-expr-pairs = new Def-s-expr-pairs();
for (def-s-expr in def-s-exprs-1)
  var found-def-s-expr: Def-s-expr = findByTypeAndName(def-s-expr, def-s-exprs-2);
  if (found-def-s-expr is not null)
    addTo(def-s-expr-pairs, makePair(def-s-expr, found-def-s-expr));
return def-s-expr-pairs;
```

Рис. 6. Псевдокод алгоритму формування пар def-s-виразів за іменем

Приклад результату роботи формування пар порівняння для файлів v1 та v2, які містять програми, записані мовою Common Lisp, показано на рисунку 7.

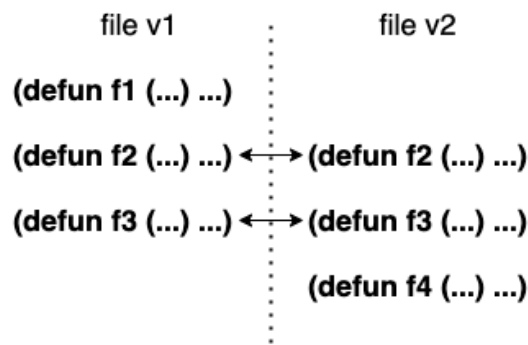


Рис. 7. Приклад утворення пар порівняння за іменем та типом def-s-вираза

Алгоритм етапу виявлення вузлів типу «same». Для виявлення «same» вузлів використовується модифікований алгоритм пошуку LCS, який був запропонований в [8].

Цей алгоритм формує співпадаючі пари вузлів в абстрактних семантичних деревах виразів def-1 та def-2, позначені як «same» вузли, а також відповідні списки піддерев def-1-unsure-subtrees та def-2-unsure-subtrees, які містять корені піддерев, статус вузлів яких допоки невідомий.

Алгоритми виявлення вузлів типів «moved» та «unique». Виявлення вузлів типу «moved» – це фактично операція виявлення переміщених піддерев. Пропонується два алгоритми для виконання цього етапу:

- алгоритм з використанням обходу вузлів лісу піддерев def-2-unsure-subtrees;
- алгоритм з використанням хеш-таблиці, куди заносяться всі піддерева з def-2-unsure-subtrees.

Обидва ці алгоритми використовують одну спільну структуру даних, а саме список maybe-moved-from-v1, який формується з піддерев списку def-1-unsure-subtrees. Основний інваріант списку maybe-moved-from-v1 полягає у тому, що елементи списку (тобто піддерева) мають знаходитися в порядку спадання кількості вузлів у піддеревах.

Обидва алгоритми використовують однаковий базовий алгоритм, псевдокод якого показано на рисунку 8.

```
function find-moved-and-unique (maybe-moved-from-v1: List<AstNode>,
                               maybe-moved-in-v2: T)
while (maybe-moved-from-v1 is not empty)
    val pattern-tree: AstNode = pop(maybe-moved-from-v1);
    val equal-tree-in-v2: AstNode = find-in(maybe-moved-in-v2,
                                           pattern-tree);

    if (equal-tree-in-v2 is not null)
        mark-all-node(pattern-tree, MOVED);
        mark-all-node(equal-tree-in-v2, MOVED);
    else
        mark-root(pattern-tree, UNIQUE);
        val list-of-SubTs: List<AstNode> = decay(pattern-tree);
        add-correctly-to(maybe-moved-from-v1, list-of-SubTs);
```

Рис. 8. Псевдокод базового алгоритму

Різниця між цими двома алгоритмами полягає в типі аргументу maybe-moved-in-v2 та реалізації функції find-in (на рисунку 8 вони виділені жовтим кольором).

Опишемо деталі реалізації алгоритму з використанням обходу вузлів лісу піддерев.

Функція цього алгоритму має параметр maybe-moved-in-v2 типу List<AstNode>, до якого при виклику цієї функції передається аргумент def-2-unsure-subtrees, що є списком вузлів АСД.

Функція find-in реалізує обхід по лісу піддерев, що знаходиться в списку maybe-moved-in-v2 й намагається знайти в ньому піддерево, у якого статус не є «moved» та всі додаткові параметри збігаються з додатковими параметрами дерева шаблону pattern-tree. Якщо при цьому використовується емпіричний режим пошуку переміщених піддерев, то вважається, що виявлене піддерево збігається з шаблонним деревом pattern-tree емпірично, в іншому випадку, проводиться точна перевірка на збіг. Особливе значення для оптимізації обходу дерева при роботі цього алгоритму має інформаційне поле вузла max-depth: якщо значення поля max-depth шаблонного дерева pattern-tree вже збігається зі значенням поля max-depth поточного дерева current-tree, що перевіряється, але співпадіння ще не було виявлене, то подальшого заглиблення у дерево current-tree не відбувається.

Опишемо деталі реалізації алгоритму з використанням хеш-таблиці.

Функція цього алгоритму має параметр maybe-moved-in-v2 типу HashTable<uint64, List<AstNode>>, до якого при виклику цієї функції передається аргумент maybe-moved-in-v2-ht, що є хеш-таблицею з ключами типу uint64, значеннями якої є список вузлів АСД.

Хеш-таблиця maybe-moved-in-v2-ht формується проходом по списку maybe-moved-in-v2 та занесенням піддерев із цього списку до комірок хеш-таблиці, в яких знаходяться списки піддерев, що мають однаковий з ними хеш. Приклад такої ініціалізації показано на рисунку 9. При створенні хеш-таблиці потрібно враховувати, що її розмір не буде перевищувати сумарної кількості всіх піддерев, що знаходяться у списку def-2-unsure-subtrees. Тому, якщо відома сумарна кількість всіх піддерев, то доцільно передавати цю кількість до функції створення хеш-таблиці, щоб задати її початковий розмір, оскільки це прискорить ініціалізацію хеш-таблиці.

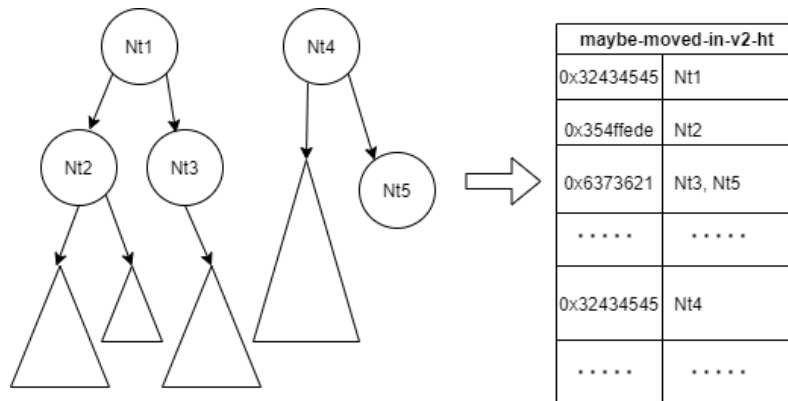


Рис. 9. Приклад ініціалізації хеш-таблиці maybe-moved-in-v2-ht піддеревами зі списку def-2-unsure-subtrees

Функція find-in бере у шаблонного дерева pattern-tree його хеш та шукає відповідний йому список піддерев в хеш-таблиці maybe-moved-in-v2-ht. Приклад роботи функції find-in у алгоритмі з використанням хеш-таблиці показано на рисунку 10. Якщо відповідний список було знайдено, то у ньому виконується пошук піддерева, тип якого є відмінним від "moved" та яке емпірично чи точно збігається з шаблонним деревом pattern-tree. Якщо таке піддерево було виявлено, то воно зі списку не видаляється, оскільки для подальшої роботи алгоритму це не потрібно, але така операція тільки збільшить час його роботи. Тобто під час роботи структура хеш-таблиці не змінюється, а змінюються тільки статуси піддерев, що знаходяться у ньому.

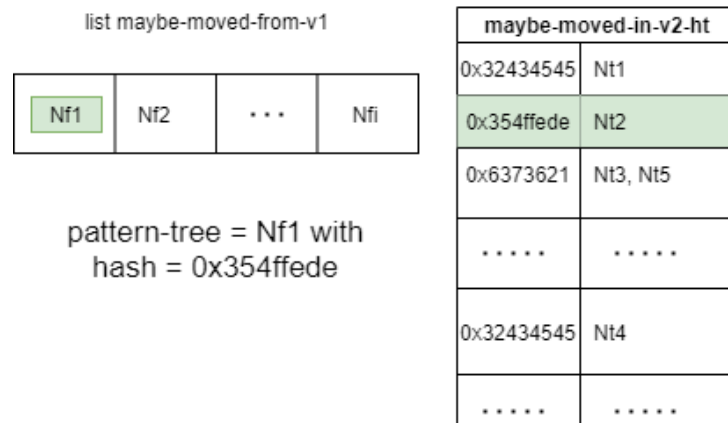


Рис. 10. Приклад роботи функції find-in у алгоритмі з використанням хеш-таблиці

Порівняння швидкодії алгоритмів виявлення переміщених фрагментів. Для вимірювання швидкодії алгоритмів було створено програму, яка генерує дві версії Lisp-файлу, що мають певну задану структуру, але є різними. Ці файли містять одне визначення функції ім'я якої однаково у двох версіях вхідних файлів. Ця функція містить три виклики довільних функцій, передаючи їм три аргументи, які можуть бути або атомами у вигляді символів, або іменами інших довільних функцій з трьома параметрами. Одним із головних параметрів при генерації Lisp-файлів є значення максимальної глибини викликів max-depth.

Перший випадок вимірювання швидкодії: тіло певної функції в обох версіях є зовсім різним, але має однакову структуру. Цей випадок вимірювання визначає швидкість виявлення повної незбіжності тіл функцій і цікавий тим, що всі можливі для виділення піддерева не є переміщеними й, таким чином, відбувається «максимальне» навантаження на представлені алгоритми. Порівняння швидкості пошуку обома алгоритмами для цього випадку показано на рисунку 11. Як видно з цього рисунку, алгоритм з обходом лісу має перевагу до значення max-depth=2, після чого алгоритм з використанням хеш-таблиці стає швидшим.

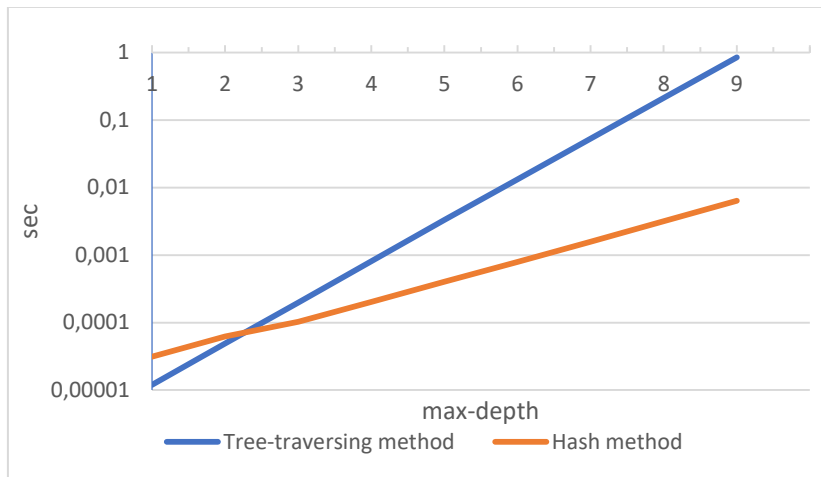


Рис. 11. Графік тестування 1-го випадку вимірювання швидкодії

Другий випадок вимірювання швидкодії: дерева вхідних АСД в обох версіях є різними до ярусу, де вже знаходяться переміщені фрагменти (назвемо цей параметр як `depth-of-moved-fragments`). Значення `max-depth` не змінюється й встановлюється рівним 14. Порівняння швидкості пошуку обома алгоритмами для цього випадку (рис. 12) показує цікавий результат: алгоритм з обходом лісу дерев має менший час роботи, тобто виграє по швидкодії до значення `depth-of-moved-fragments=2`, а далі починає лінійно уповільнюватися. З іншої сторони, алгоритм з використанням хеш-таблиці не тільки почав показувати кращу швидкість роботи, але його швидкодія навіть трохи зростала до значення `depth-of-moved-fragments=4`. Це пов'язано з тим, що при збільшенні значення `depth-of-moved-fragments` зменшується кількість вузлів, які треба перевіряти на збіг. Але цей факт не допомагає першому алгоритму з обходом лісу дерев, оскільки для цього алгоритму зі зменшенням кількості вузлів, які треба перевіряти, одночасно збільшується кількість вузлів, які треба обійти, щоб дійти до кореня відповідного переміщеного фрагмента.

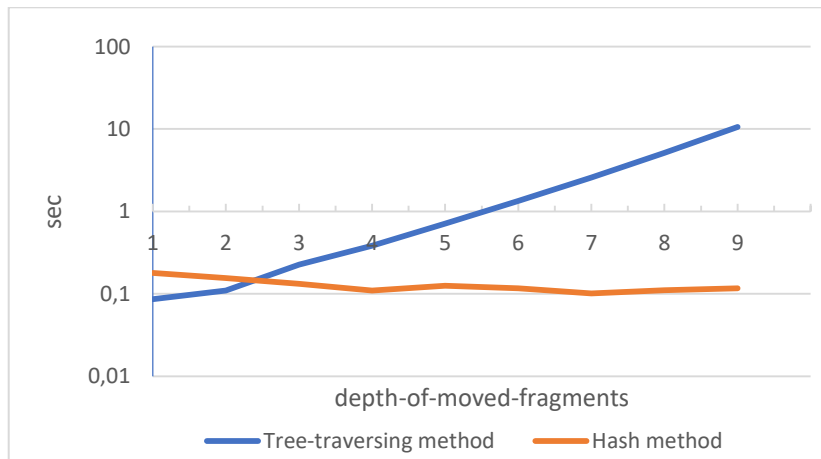


Рис.12. Графік тестування 2-го випадку вимірювання швидкодії

Висновки. У даній статті запропоновано два алгоритми виявлення переміщених фрагментів дерев, які є частинами реалізації раніше запропонованого авторами способу порівняння програм, написаних Lisp-подібними мовами, на основі абстрактних семантичних дерев. Перший з алгоритмів з метою виявлення переміщених фрагментів дерев використовує обхід лісу дерев, а другий алгоритм використовує хеш-таблицю.

Було також проведено порівняння швидкодії обох запропонованих алгоритмів та отримано результати, що демонструють більшу швидкодію алгоритму, який використовує хеш-таблиці під час знаходження переміщених фрагментів, перед алгоритмом, який використовує обхід лісу дерев, майже в усіх випадках.

Напрямок подальших досліджень для виявлення переміщених фрагментів програм може бути пошук варіантів врахування факту перейменування змінних, які не змінюють семантику цих переміщених фрагментів.

Список бібліографічного опису

1. State of Clojure 2022 [Електронний ресурс]. – 2022. – Режим доступу до ресурсу: <https://clojure.org/news/2022/06/02/state-of-clojure-2022>.
2. Єрмоленко Д. В., Марченко О.І. Спосіб порівняння абстрактних семантичних дерев програм написаних Lisp-подібними мовами. Прикладна математика та комп'ютинг. ПМК, 2022 : п'ятнадцята наук. конф. магістрантів та аспірантів, 16–18 листопада 2022 р.: зб.тез доп./[редкол.: Дичка І.А. та ін.]. – К. : Просвіта, 2022. – с. 261-265.
3. Tree edit distance [Електронний ресурс] – Режим доступу до ресурсу: <http://tree-edit-distance.dbresearch.uni-salzburg.at/>.
4. S-expression [Електронний ресурс] – Режим доступу до ресурсу: <https://www.computerhope.com/jargon/s/s-expression.htm>.
5. Hashimoto, Masatomo, Mori. Diff/TS: A Tool for Fine-Grained Structural Change Analysis. Proceedings [Електронний ресурс] / Hashimoto, Masatomo, Mori, Akira // Working Conference on Reverse Engineering, WCRE. 279 - 288. – 2008. – Режим доступу до ресурсу: <https://doi.org/10.1109/WCRE.2008.44>
6. Magniez F., Rougemont M. Property testing of regular tree languages [Електронний ресурс] / F. Magniez and M. de Rougemont // Algorithmica, 49(2):127–146. – 2007. – Режим доступу до ресурсу: <https://doi.org/10.1007/s00453-007-9028-3>
7. Hashing tree structure [Електронний ресурс] – Режим доступу до ресурсу: <https://www.baeldung.com/cs/hashing-tree>.
8. Єрмоленко, Д. В. Засіб порівняння версій програм на мові LISP з використанням абстрактного семантичного дерева : дипломний проєкт бакалавра : 123 Комп'ютерна інженерія / Єрмоленко Денис Вадимович. – Київ, 2021. – 75 с.

References

1. "State of Clojure 2022", Robert Randolph , Clojure, <https://clojure.org/news/2022/06/02/state-of-clojure-2022>. Accessed 20 Nov. 2022.
2. Yermolenko D., Marchenko O. Algorithms of the technique for comparing programs written in Lisp-like languages based on abstract semantic trees. Applied Mathematics and Computing. AMC, 2022 : fifteen scientific conference of master students and postgraduates, 16–18 листопада 2022 р.: зб.тез доп./[редкол.: Дичка І.А. та ін.]. – К. : Просвіта, 2022. – с. 261-265.
3. "Tree edit distance", Mateusz Pawlik, <http://tree-edit-distance.dbresearch.uni-salzburg.at/>. Accessed 21 Nov. 2022
4. "S-expression", Computer Hope, <https://www.computerhope.com/jargon/s/s-expression.htm>. Accessed 21 Nov. 2022
5. Hashimoto, Masatomo & Mori, Akira. (2008). Diff/TS: A Tool for Fine-Grained Structural Change Analysis. Proceedings - Working Conference on Reverse Engineering, WCRE. 279 - 288. 10.1109/WCRE.2008.44. <https://doi.org/10.1109/WCRE.2008.44>. Accessed 21 Nov. 2022
6. F. Magniez and M. de Rougemont. Property testing of regular tree languages. Algorithmica, 49(2):127–146, 2007. <https://doi.org/10.1007/s00453-007-9028-3>. Accessed 21 Nov. 2022
7. "Hashing tree structure", Said Sryheni, Baeldung, <https://www.baeldung.com/cs/hashing-tree>. Accessed 22 Nov. 2022
8. Yermolenko D. V. (2021). Tool for comparing versions of programs in the LISP language using an abstract semantic tree [Diploma project, NTUU KPI]

DOI: <https://doi.org/10.36910/6775-2524-0560-2022-49-06>

УДК 001.893

Козубцов Ігор Миколайович¹, д.пед.н., к.т.н., с.н.с.

<https://orcid.org/0000-0002-7309-4365>

Сілко Олексій Вікторович¹, к.т.н., доцент

<https://orcid.org/0000-0001-6605-6945>

Козубцова Леся Михайлівна¹, к.т.н.

<https://orcid.org/0000-0002-7866-8575>

Лукашенко Вікторія Вікторівна², к.т.н., доцент

<https://orcid.org/0000-0002-8898-2269>

¹ Військовий інститут телекомунікацій та інформатизації ім. Героїв Крут, м. Київ, Україна

² Національний авіаційний університет, м. Київ, Україна

ПОКАЗНИКИ ТА КРИТЕРІЇ ОЦІНЮВАННЯ РІВНЯ АКАДЕМІЧНОЇ ДОБРОЧЕСНОСТІ ЗДОБУВАЧІВ ВИЩОЇ ОСВІТИ

Козубцов І.М., Сілко О.В., Козубцова Л.М., Лукашенко В.В. Показники та критерії оцінювання рівня академічної доброчесності здобувачів вищої освіти. У науковій статті обґрунтовано показники академічної доброчесності здобувачів вищої освіти. Відповідно до кожного показника академічної доброчесності запропоновано математичні критерії оцінювання їх рівня. Запропоновано додатково можливість обчислювати інтегральний показник академічної доброчесності здобувачів вищої освіти. Встановлено міждисциплінарний зв'язок академічної доброчесності з культурологічною характеристикою здобувачів вищої освіти. Наукова новизна. Вперше запропоновано окремі показники та критерії оцінювання рівня академічної доброчесності здобувачів вищої освіти. Практичне значення. На основі одержаних показників та критеріїв можна розробити методiku оцінювання рівня академічної доброчесності здобувачів вищої освіти.

Ключові слова: показники, критерії, оцінювання, плагіат, академічна доброчесність, здобувач вищої освіти.

Kozubtsov I., Silko O., Kozubtsova L., Lukashenko V. Indicators and criteria for assessing the level of academic integrity of applicants for higher education. The scientific article substantiates the indicators of academic integrity of applicants for higher education. According to each indicator of academic integrity, mathematical criteria for assessing their level are proposed. Additionally, it is proposed to calculate integral index of academic integrity of applicants of higher education. An interdisciplinary connection between academic integrity and the cultural characteristics of applicants for higher education has been established. Scientific novelty. For the first time, separate indicators and criteria for assessing the level of academic integrity of applicants for higher education have been proposed. Practical significance. Based on the obtained indicators and criteria, a methodology for assessing the level of academic integrity of applicants for higher education can be developed.

Keywords: indicators, criteria, assessment, plagiarism, academic integrity, higher education applicant.

Постановка завдання і зв'язок її з важливими науковими завданнями. Стійкою перешкодою до розвитку навичок доброго академічного письма в Україні є традиція неякісних, несумісних із кращим світовим досвідом і вимогами практик академічного письма [1]. Тиск цих традицій суттєво загальмовує поширення кращих практик академічного письма в Україні як серед наукових працівників і науково-педагогічних працівників, так і серед студентів. Науково-педагогічні працівники по суті навчають студентів тих самих навичок виконання письмових робіт, якими володіють самі.

Під час навчання здобувачі вищої освіти виконують різні види письмових робіт. Основними з них є дипломні роботи (проекти), курсові роботи (проекти), звіти з практики, поточні письмові роботи (реферати, есеї, аналітичні записки тощо).

Аналіз вимог, які висувають заклади вищої освіти до кількості та обсягу письмових робіт студентів та курсантів, кількості використовуваних джерел та інших формальних показників, свідчить, що ці вимоги часто не узгоджені із часом, який надається на виконання відповідних робіт.

Проблема забезпечення академічної доброчесності при підготовки здобувачів вищої військової освіти у вищих навчальних закладах наразі не вирішена до кінця. Згідно тлумачення статті 42 Закону України «Про освіту» [2]: «Академічна доброчесність – це сукупність етичних принципів та визначених законом правил, якими мають керуватися учасники освітнього процесу під час навчання, викладання та провадження наукової (творчої) діяльності з метою забезпечення довіри до результатів навчання та/або наукових (творчих) досягнень».

Проблема доброчесності існувала постійно, оскільки людина вдосконалюється і їй хочеться

чогось більшого від життя, вона починає наслідувати та копіювати більш талановитих і успішних колег. Одним людям вдається оволодіти професійною діяльністю та з часом набувають майстерності до створення власних творів, використовуючи накопичений попередніми поколіннями досвід. Є також люди, що закінчили навчальні заклади, але і на далі залишаються не здатними до творчої діяльності. В таких людей дуже часто виникає комплекс до намагання чужу роботу видавати за свою, щоб не здаватися гірше за інших. Бажання «не здаватися гірше за інших» і штовхає людей на подібні вчинки. Із цим явищем у житті доводиться зіштовхуватися таки часто.

Дуже гостро постало питання плагіату у дисертаціях. Велике зростання захищених дисертацій сигналізує про потребу посиленого контролю щодо можливої наявності у них плагіату.

У наш час проблеми доброчесності та плагіату, як ніколи хвилюють суспільство. Не зважаючи на наявність чисельних рекомендації щодо забезпечення принципів академічної доброчесності, наприклад Міністерством освіти і науки України [3], проблема по суті не вирішується, і цьому підтвердження аналіз останніх наукових досліджень і публікацій за напрямком.

Аналіз останніх досліджень і публікацій. Дослідженню питання плагіату, як складової доброчесності, присвячено багато наукових досліджень.

У статті [4] проаналізовано причини прояву академічної не доброчесності у здобувачів вищої освіти в американських університетах та запропоновано шляхи розв'язання цієї проблеми. На думку авторів, до причин академічної не доброчесності належать наступні: бажання випередити інших, страх невдачі, слабкі навички управління часом, нерозуміння неприпустимості плагіату тощо. Для подолання академічної не доброчесності у здобувачів вищої освіти пропонується суворе дотримання заходів проти шахрайства та покарання шахраїв. Роль викладачів дослідники вбачають, у першу чергу, у роз'ясненні здобувачам наслідків обману у навчанні.

Головний висновок, до якого дійшли дослідники [5] у результаті кількісного аналізу результатів, – існує розрив у розумінні академічної доброчесності між академічним персоналом та студентами та їх реакції на нечесну поведінку.

На думку авторів роботи [6] в умовах сучасного стану університетської освіти України «принципи академічної доброчесності або сприймаються як щось абстрактне, модне, про що часто згадують, але не застосовують».

З точки зору застосування об'єктивного підходу в оцінці окремих складових академічної доброчесності, цікавою є методика та методологія об'єктивної оцінки [7;8].

Відповідь на питання «Який відсоток схожості можна вважати допустимим для академічних робіт» подано на сайті [unicheck](http://unicheck.com) [9]. Рекомендовані показники оригінальності за величиною коефіцієнта подібності становлять: не більше 20% – оригінальна робота, від 21% до 50% – задовільно оригінальна робота, від 51% до 90% – умовно оригінальна робота, понад 90% – неоригінальна робота.

У статті [10] обговорюється проблема коректності роботи системи «Антиплагіат» під час перевірки наукових праць та доцільності використання систем такого роду під час проведення наукового дослідження. Неможливо зрозуміти, на якій підставі береться цифра в 70-80%, що встановлюється як вимога до оригінальності наукової роботи (часто – і до навчальної)? Хто і як розрахував, що оригінальна наукова думка у всіх випадках, тобто при вирішенні будь-якого дослідницького завдання та при розкритті будь-якого предмета, дає при формальній оцінці цього відсотка?

Більш узагальненими є критерії та показники оцінювання забезпечення академічної доброчесності подані як рекомендації для експертів Національного агентства із забезпечення якості вищої освіти [11].

За результатами аналізу систем та програм для виявлення запозичень, автор дослідження підтверджує, що жодна з них не може дати 100% гарантію, що в роботі знайдено всі унікальні фрагменти [12]. І тому для отримання більш точного результату перевірки краще перевірити текст кількома системами або звернутися за експертизою, хоча цей спосіб є платним і довготривалим.

Проте аналіз наукової літератури виявив відсутність обґрунтованих критеріїв та показників оцінювання рівня плагіату окрім [9], як складової академічної доброчесності.

Резюмуючи аналіз останніх досліджень і публікацій збігається наша точка з авторами роботи [13], що плагіат є важливим чинником і індикатором якості освіти, а академічна

добросесність – основа для забезпечення якісної освіти [14]. Людина показує себе у тому, що вона залишає після себе.

Мета статті. Отже, метою статті є обґрунтований вибір критерії та показників оцінювання рівня плагіату, як складової академічної добросесності здобувачів вищої освіти.

Для досягнення мети поставлено такі задачі:

1. Проаналізувати останні дослідження та публікацій за визначеною проблемою.

2. Обґрунтувати вибір критерії та показників оцінювання рівня плагіату, як складової академічної добросесності здобувачів вищої освіти.

Матеріали й методи. Для вирішення поставлених завдань використовувалася сукупність методів теоретичного дослідження: історичного аналізу та узагальнення наукової літератури щодо проблеми дослідження; структурно-генетичного аналізу та синтезу при уточненні об'єкта та предмета дослідження; метод сходження від абстрактного до конкретного; метод аналітично-порівняльного аналізу при аналітично-порівняльному оцінюванні новизни результатів дослідження; синтез та узагальнення – для обґрунтування методологічних та методичних засад дослідження; узагальнення – формулювання висновків та рекомендацій щодо продовження подальших досліджень.

Виклад основного матеріалу. Згідно з пунктом 4 статті 42 Закону України «Про освіту» [2] на рис. 1 подано ключові види порушенням академічної добросесності.

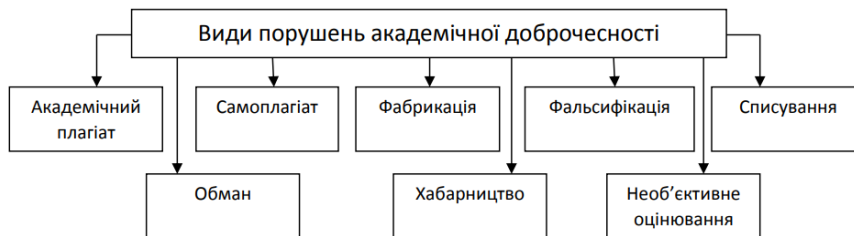


Рисунок 1 – Види порушень академічної добросесності згідно ЗУ «Про освіту»

Етап 1. Розробка системи показників та критеріїв їх оцінювання.

Академічній добросесності (AD) протиставляється категорія академічної не добросесності, основні прояви якої знаходимо у таких видах діяльності [1]: академічний плагіат (Pl); самоплагіат (SPl); фабрикація (Fr); фальсифікація (Fs); списування (Sp); обман (Ob); хабарництво (Ha); необ'єктивне оцінювання (NO); хибне співавторство (HSa).

Перелічені категорії раціонально розглядати як множину показників, що в сукупності характеризують академічну добросесність суб'єкта – здобувача вищої освіти (1) :

$$AD = \{SPl; Pl; Fr; Fs; Sp; Ob; Ha; HSa\} \quad (1)$$

Зобразимо академічну добросесність здобувача вищої освіти як множина показників (див. рис. 1).

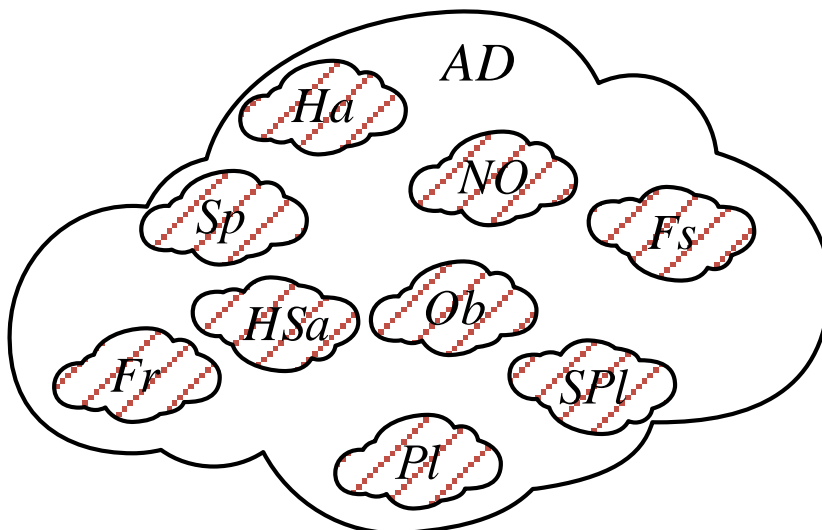


Рис. 1. Академічна добросесність здобувача вищої освіти як множина показників

Показник «академічний плагіат» (Pl). Академічний плагіат – «оприлюднення (частково або повністю) наукових (творчих) результатів, отриманих іншими особами, як результатів власного дослідження (творчості), та/або відтворення опублікованих текстів (оприлюднених творів мистецтва) інших авторів без зазначення авторства» [1].

Вирізняють такі основні різновиди академічного плагіату:

- дослівне запозичення текстових фрагментів без оформлення їх як цитат з посиланням на джерело (в окремих випадках некоректним вважають навіть використання одного слова без посилання на джерело, якщо це слово використовують в унікальному значенні, наданому цим джерелом);

- використання інформації (факти, ідеї, формули, числові значення тощо) з джерела без посилання на це джерело;

- перефразування тексту джерела у формі, що є близькою до оригінального тексту, або наведення узагальнення ідей, інтерпретацій чи висновків з певного джерела без посилання на це джерело;

- подання як власних робіт (дисертацій, монографій, навчальних посібників, статей, тез, звітів, контрольних, розрахункових, курсових, дипломних та магістерських робіт, есеїв, рефератів тощо), виконаних на замовлення іншими особами, у тому числі робіт, стосовно яких справжні автори надали згоду на таке використання.

Виходячи з обраних меж значень плагіату (Pl) визначимо рівень наукової новизни (SN) в тексті, що перевіряється. Коректніше говорити про наукову унікальність тексту (SU), а ніж про наукову новизну (SNt). Між науковою унікальністю тексту та наукою новизною існує наступне співвідношення (2):

$$SU = (1 - Pl) \quad (2)$$

Критерії оцінювання рівня показника плагіату в тексті та співвідношення із рівнем її наукової унікальності подано в табл. 1.

Таблиця 1 Критерії оцінки рівня показника плагіату та наукової унікальності

Плагіат		Наукова унікальність	
Критерій плагіату	Лінгвістичний опис рівня показника плагіату	Критерій унікальності	Лінгвістичний опис рівня показника наукової унікальності
$0 \leq Pl \leq 0,25$	Дуже низький рівень плагіату	$1 \leq SU \leq 0,85$	Дуже високий рівень наукової унікальності
$0,25 < Pl \leq 0,5$	Низький рівень плагіату	$0,85 < SU \leq 0,5$	Високий рівень наукової унікальності
$0,5 < Pl \leq 0,75$	Середній рівень плагіату	$0,5 < SU \leq 0,25$	Середній рівень наукової унікальності
$0,75 < Pl \leq 0,9$	Високий рівень плагіату	$0,25 < SU \leq 0,1$	Низький рівень наукової унікальності
$0,9 < Pl \leq 1$	Дуже високий рівень плагіату	$0,1 < SU \leq 0$	Відсутня наукова унікальність

Показник «самоплагіат» (SPI). Самоплагіат – «оприлюднення (частково або повністю) власних раніше опублікованих наукових результатів як нових наукових результатів».

Типовими прикладами самоплагіату є:

- дублікація публікацій – публікація однієї і тієї самої наукової роботи (цілком або з несуттєвими змінами) в декількох виданнях, а також повторна публікація (цілком або з несуттєвими змінами) раніше оприлюднених статей, монографій, інших наукових робіт, як нових наукових робіт;

- дублювання наукових результатів – публікація одних і тих самих наукових результатів, в різних статтях, монографіях, інших наукових працях, як нових результатів, які публікуються вперше;

- подання у звітах з виконання наукових проектів результатів, що містилися у попередніх роботах, як отриманих при виконанні відповідного проекту;

- агрегування чи доповнення даних – суміщення старих і нових даних без їх чіткої

ідентифікації з відповідними посиланнями на попередні публікації;

- дезагрегування даних – публікація частини раніше опублікованих даних без посилання на попередню публікацію;
- повторний аналіз раніше опублікованих даних без посилання на попередню публікацію цих даних та раніше виконаний їх аналіз.

$$SPI \approx PI \quad (3)$$

Для оцінювання рівня показника самоплагіату рекомендуємо наступні критерії табл. 2.

Таблиця 2. Критерії оцінювання рівня показника самоплагіату (*SPI*)

Критерій самоплагіату	Лінгвістичний опис рівня показника самоплагіату
$0 \leq SPI \leq 0,25$	Дуже низький рівень самоплагіату
$0,25 < SPI \leq 0,5$	Низький рівень самоплагіату
$0,5 < SPI \leq 0,75$	Середній рівень самоплагіату
$0,75 < SPI \leq 0,9$	Високий рівень самоплагіату
$0,9 < SPI \leq 1$	Дуже високий рівень самоплагіату

Показник «фабрикація» (*Fr*). Фабрикація – «вигадування даних чи фактів, що використовуються в освітньому процесі чи наукових дослідженнях». Проблема наукових публікацій, дисертацій, дипломних і курсових робіт студентів, яка полягає у тому, що в цих роботах нерідко наводяться дані про експерименти, емпіричні дослідження, вимірювання, розрахунки, опитування, інші види досліджень та їх апробацію, які насправді не виконувалися.

Критерії оцінювання рівня показника фабрикації подано в табл. 3.

Таблиця 3. Критерії оцінювання рівня показника фабрикації (*Fr*)

Критерій фабрикації	Лінгвістичний опис рівня показника фабрикації
$Fr = 0$	Відсутні ознаки фабрикації
$Fr = 1$	Присутні ознаки фабрикації

Показник «фальсифікація» (*Fs*). Фальсифікація – «свідома зміна чи модифікація вже наявних даних, що стосуються освітнього процесу чи наукових досліджень». Проблема наукових публікацій, дисертацій, дипломних і курсових робіт студентів, яка полягає у тому, що автори вносять зміни до результатів досліджень або приховують окремі результати з метою позбавлення від даних, що спростовують або не підтверджують гіпотези, які вони захищають, висновки, які вони роблять, тощо.

Критерії оцінювання рівня показника фальсифікації подано в табл. 4.

Таблиця 4. Критерії оцінювання рівня показника фальсифікації (*Fs*)

Критерій фальсифікації	Лінгвістичний опис рівня показника фальсифікації
$Fr = 0$	Відсутні ознаки фальсифікації
$Fr = 1$	Присутні ознаки фальсифікації

Фальсифікація може стосуватися неповного або свідомо викривленого опису методик дослідження з метою приховування: виявлених авторами методичних помилок; використання застарілого або непридатного для відповідних досліджень обладнання; застосування непридатних для цілей дослідження алгоритмів та програмного забезпечення; інших хиб, що могли вплинути на достовірність, точність і надійність представлених результатів.

Показник «списування» (*Sp*). Списування – «виконання письмових робіт із залученням зовнішніх джерел інформації, крім дозволених для використання, зокрема під час оцінювання результатів навчання».

Процедура списування здобувачем вбачає одержання дзеркальної копії начального, наукового чи іншого матеріалу. Виходячи з логіки очевидних речей для оцінки списування (*Sp*) раціонально використати аналогічний показник оцінювання академічного плагіату (*PI*) (4):

$$Sp \approx PI \quad (4)$$

Тільки на відміну від академічного плагіату (*PI*) здобувач вищої освіти мав за мету одержати копію тексту. Чи слід зараховувати здобувачеві вищої освіти до списання переписана лекція співкурсника? З досвіду підготовки курсантів однозначно ні. Тому, що курсанти окрім навчання виконують пряму посадову функцію виконання службових обов'язків в добовому наряді. В день

чергування вони звільняються з поважної причини від відвідування лекційних та інших видів занять. Якщо лабораторні роботи вони обов'язково повинні відпрацювати, то лекційний матеріал, як правило переписуються з зошитів тих курсантів, які найкращим чином конспектують лекції. Щодо студентів може застосовуватися норма поважної причини. Інша річ, якщо здобувач вищої освіти мав за мету списати однаковий варіант рішення практичного або лабораторного варіанту, то тоді однозначно варто застосовувати критерії оцінки списування (Sp). Критерії оцінювання рівня показника списування подано в табл. 5.

Таблиця 5 Критерії оцінювання рівня показника списування (Sp).

Критерій списування	Лінгвістичний опис рівня показника списування
$0 \leq Sp \leq 0,25$	Дуже низький рівень списування
$0,25 < Sp \leq 0,5$	Низький рівень списування
$0,5 < Sp \leq 0,75$	Середній рівень списування
$0,75 < Sp \leq 0,9$	Високий рівень списування
$0,9 < Sp \leq 1$	Дуже високий рівень списування

Показник «обман» (Ob). Обман – «надання завідомо неправдивої інформації стосовно власної освітньої (наукової, творчої) діяльності чи організації освітнього процесу; формами обману є, зокрема, академічний плагіат, самоплагіат, фабрикація, фальсифікація та списування».

Критерії оцінювання рівня показника обману подано в табл. 6.

Таблиця 6. Критерії оцінювання рівня показника обману (Ob)

Критерій обману	Лінгвістичний опис рівня показника обману
$Ob = 0$	Відсутні ознаки обману
$Ob = 1$	Присутні ознаки обману

Показник «хабарництво» (Ha). Хабарництво – «надання (отримання) учасником освітнього процесу чи пропозиція щодо надання (отримання) коштів, майна, послуг, пільг чи будь-яких інших благ матеріального або нематеріального характеру з метою отримання неправомірної переваги в освітньому процесі». До хабарництва можуть бути зараховані одержання, провокування або пропонування неправомірної вигоди за отримання позитивної або вищої оцінки при складанні будь-якого виду поточного та підсумкового контролю, а також будь-яких інших переваг у навчальній, дослідницькій чи трудовій діяльності; примусові благодійні внески та примусова праця здобувачів освіти та/або їх батьків; примусове надання освітніх послуг (примусове репетиторство); деякі випадки конфлікту інтересів тощо [1].

Критерії оцінювання рівня показника хабарництва подано в табл. 7.

Таблиця 7. Критерії оцінювання рівня показника хабарництва (Ha)

Критерій хабарництва	Лінгвістичний опис рівня показника хабарництва
$Ha = 0$	Відсутні ознаки хабарництва
$Ha = 1$	Присутні ознаки хабарництва

Показник «необ'єктивне оцінювання» (NO). Необ'єктивне оцінювання – «свідоме завищення або заниження оцінки результатів навчання здобувачів освіти» [1].

Критерії оцінювання рівня показника необ'єктивного оцінювання подано в табл. 8.

Таблиця 8. Критерії оцінювання рівня показника необ'єктивного оцінювання (NO)

Критерій необ'єктивного оцінювання	Лінгвістичний опис рівня показника необ'єктивного оцінювання
$NO = 0$	Відсутні ознаки необ'єктивного оцінювання
$NO = 1$	Присутні ознаки необ'єктивного оцінювання

Показник «хибне співавторство» (HSa). До неправдиве співавторства, як форми обману, належить:

- приписування співавторства особам, які не брали кваліфікованої участі у дослідженні та підготовці публікації (зокрема, це стосується зарахування до авторів керівників установ і підрозділів, які безпосередньо не брали участі у виконанні роботи, працівників чи студентів, які здійснювали лише технічну допомогу, тощо);

• не включення до співавторів осіб, які брали активну кваліфіковану участь у дослідженні та підготовці публікації, зокрема у постановці цілей та завдань роботи, формулюванні її висновків, розробці алгоритмів, аналізі результатів експериментів та розрахунків, написанні тексту тощо.

Критерії оцінювання рівня показника хибного співавторства подано в табл. 9.

Таблиця 9. Критерії оцінювання рівня показника хибного співавторства (HSa)

Критерій хибного співавторства	Лінгвістичний опис рівня показника хибного співавторства
$HSa = 0$	Відсутні ознаки хибного співавторства
$HSa = 1$	Присутні ознаки хибного співавторства

Етап 2. Планування процедур збирання вихідних даних для оцінювання.

Етап 3. Обчислення окремих значень показників академічної доброчесності.

Етап 4. Узагальнене (інтегральне) обчислення академічної доброчесності здобувачів вищої освіти (5):

$$AD = \frac{\sum_{i=1}^n AD_i}{n} \quad (5)$$

AD_i – показники академічної доброчесності;

n – кількість показників академічної доброчесності;

Узагальнене оцінювання академічної доброчесності здобувачів вищої освіти через окремі показники як інтегральний (табл. 10).

Таблиця 10 – Узагальнене оцінювання академічної доброчесності здобувачів вищої освіти через окремі показники

№ П/П	Узагальнений критерій академічної доброчесності	Лінгвістичний опис академічної доброчесності	Еквівалентний рівень моральні властивості особистості (M)	Лінгвістичний опис моральні властивості особистості
1	$0 \leq AD \leq 0,25$	Дуже високий	5	Дуже високий
2	$0,25 < AD \leq 0,5$	Високий	4	Високий
3	$0,5 < AD \leq 0,75$	Середній	3	Середній
4	$0,75 < AD \leq 0,9$	Низький	2	Низький
5	$0,9 < AD \leq 1$	Дуже низький	1	Дуже низький

Обговорення результатів дослідження.

У статті [15] представлено результати виконання проекту, місія якого полягала у підвищенні дослідницької доброчесності шляхом просування дослідницької культури, що уможливило внутрішню мотивацію до проведення досконалих досліджень, а не лише орієнтацію на зовнішні фактори та обмеження. Авторами розглянуто та охарактеризовано поняття та принципи дослідницької доброчесності, категорії її порушень. Ми повністю погоджуємося, що на схильність до академічної доброчесності впливає мотиви (моральні властивості) дослідника. Моральні властивості особистості є рушійною силою людини до якісної професійної діяльності [16].

Автори дослідження [17] академічну нечесність охарактеризували, як феномен академічної культури. Якщо розглядати академічну доброчесність через показник плагіату, то на думку автора [18, с. 125] плагіат – це феномен із нечіткими межами, інтерпретація якого залежить від культурного контексту, та має соціокультурні коріння [19]. Отже, на академічну доброчесність (лояльність), згідно нашого припущення впливає мораль, мотивація, культурологічна складова методології та дослідницької компетентності здобувачів вищої освіти.

За даними [20] більшість респондентів з числа здобувачів вищої освіти заявили, що ніколи не копіювали фрази з інших (приблизно 62%), весь текст (94%) або окремі абзаци (82%). Більше того 40% студентів заявили, що ніколи не використовували ідеї інших людей як свої власні і лише 10% заявили те саме пізніше. Лише небагато здобувачів вищої освіти визнали, що плагіат регулярно (близько 4%), та жоден з них не заявив, що плагіат постійно (0%). Так опитування 1928 студентів та 374 викладачів у 25 ВНЗ України засвідчило, що: 78% студентів складають іспити не самостійно, 67% з них вдаються до списування під час іспиту, 23% студентів вказують на випадки отримання оцінок за послуги або гроші, 90% студентів вдаються до плагіату [21]. Ці результати

можна порівняти з результатами опитування американських студентів, серед яких більш, ніж 75% зізналися, що вдаються до обману в процесі навчальної діяльності, а 68% — визнають факти копіювання матеріалів з Інтернету без відповідних посилань на першоджерела [22]. Найпоширенішою формою прояву академічної нечесності на переконання переважної більшості дослідників є плагіат [23].

Висновки. Розроблена системи показників та критеріїв оцінювання рівня академічної доброчесності дозволяє застосувати підхід від абстрактного до конкретного. Застосувавши її можна визначити числове значення академічної доброчесності. Це в подальших дослідженнях ляже в основу методики оцінювання рівня академічної доброчесності здобувачів вищої освіти. Рішення про притягнення учасника освітнього процесу до академічної відповідальності має базуватися на доказах, а не на припущеннях. Особа, яку звинувачують, повинна мати всю необхідну інформацію стосовно того, у чому саме і на яких підставах її звинувачують. Найбільш популярним на даний час є оцінювання академічної доброчесності науковця через критерій академічного плагіату у науковому тексті. Широко розрекламовані спеціалізовані програми перевірки наукового тексту на академічний плагіат не містять критеріїв оцінювання. Результатом їх використання є загальна пропорція запозиченого тексту до загального обсягу.

Наукова новизна. Вперше запропоновано окремі показники та критерії оцінювання рівня академічної доброчесності здобувачів вищої освіти.

Практичне значення. Важливою умовою забезпечення академічної доброчесності є створення чітких і зрозумілих процедур оцінювання. На основі одержаних показників та критеріїв можна розробити методику оцінювання рівня академічної доброчесності здобувачів вищої освіти.

Перспективи подальших досліджень. Представлене дослідження не вичерпує всіх аспектів зазначеної проблеми. Теоретичні результати, становлять підґрунтя для подальшого обґрунтування методики обчислювання числового значення академічної доброчесності.

Список бібліографічного опису

1. Лист МОН України Керівникам закладів вищої освіти «Щодо рекомендацій з академічної доброчесності для закладів вищої освіти». Рекомендації від 23.10.2018 № 1/9-650. URL: <https://zakon.rada.gov.ua/rada/show/v-650729-18#Text>.
2. Закон України «Про Освіту». URL: <http://zakon0.rada.gov.ua/laws/show/2145-19/page3>.
3. Рекомендації щодо забезпечення принципів академічної доброчесності. Підкомісія 303 «Академічна доброчесність» Науково-методичної комісії 15 з організаційно-методичного забезпечення вищої освіти. К.: МОН України, 2016. 24 с.
4. DiVall M., Schlesselman L. Academic Dishonesty: Whose Fault is it Anyway? URL: <https://www.ncbi.nlm.nih.gov/pmc/articles/PMC4857630>.
5. Busch P., Bilgin A. Student and Staff Understanding and Reaction: Academic Integrity in an Australian University. URL: <https://doi.org/10.1007/s10805-014-9214-2>.
6. Академічна доброчесність: проблеми дотримання та пріоритети поширення серед молодих вчених : кол. моногр. / за заг. ред. Н.Г. Сорокіної, А.С. Артюхова, І.О. Дегтярьової. Дніпро: ДРІДУ НАДУ, 2017. 169 с.
7. Авдеева Н.В., Иванова Е.Н. Методика и методология объективной оценки заимствований в академических текстах. *Открытое образование*. 2020. Т. 24. № 6. С. 22 – 30. URL: DOI: <http://dx.doi.org/10.21686/1818-4243-2020-6-22-30>.
8. Svechkarev V. G., Ashkhamakhov K. I., Dvoynikova E. S., Guchetl I. N. Method of checking works for borrowings in the "Anti-plagiatum.vuz" system for prevention of the fraud. *Business. Education. Law*. 2020. No. 3. Pp. 395–400.
9. Який відсоток схожості можна вважати допустимим для академічних робіт? Академічна доброчесність. Січень 26, 2021. <https://unicheck.com/ua/blog/yakyj-vidsotok-shozhosti-mozhna-vvazhati-dopustymym-dlya-akademichnyh-robit>.
10. Харсеева О.В. Целесообразность и пределы использования автоматизированных систем при проверке научных работ на наличие заимствований. *Современная наука: от плагиата к академической честности: Материалы Всероссийской научно-практической конференции*, (Курск, 12 апреля 2019 г). С. 142 – 148.
11. Додаток 2 до протоколу від 07.06.2022 №9(14). Рекомендації для експертів із розгляду внутрішньої політики і процедур забезпечення академічної доброчесності та зовнішньої оцінки їх якості. Національна агентство із забезпечення якості вищої освіти. URL: <https://naqa.gov.ua/wp-content/uploads/2022/06/3.-Рекомендації-для-експертів-АкД.pdf>
12. Севостьянова И.О. Обзор систем проведения проверки на плагиат: общероссийские и зарубежные. *Научное обозрение. Педагогические науки*. 2017. № 5. С. 162 – 166.
13. Побіженко В.В., Побіженко І.А. Плагіат, як чинник зниження якості освіти. Системи обробки інформації. *Збірник наукових праць Харківського університету Повітряних Сил*. 2011. Випуск 8 (98). С. 310–313.
14. Зарівна Н.О., Поляк О.Б., Горлачук Н. В. Академічна доброчесність – основа для забезпечення якісної освіти. *Академічна доброчесність: виклики сучасності*. Збірник наукових есе учасників дистанційного етапу наукового

- стажування для освітян, (Варшава, 11.05 – 19.06.2020). С. 39 – 42.
15. Meriste H., Parder M.-L., Lõuk K., Simm K., Lilles-Heinsar L., Veski L., Soone M., Juurik M., Sutrop M. (2016). Promoting Integrity as an Integral Dimension of Excellence in Research. URL: <https://printeger.eu/wp-content/uploads/2016/10/D2.3.pdf>.
16. Михалёв А.С. Математическая трехмерная модель специалиста. «Высшая школа»: наукова-метадычны і публіцыстычны часопіс. 2015. № 6. С. 19 – 24.
17. Артьомов П.М., Пак І.В. Академічна нечесність як елемент академічної культури українського студентства: результати емпіричних досліджень. *Вісник ХНУ імені В. Н. Каразіна. Серія «Соціологічні дослідження сучасного суспільства: методологія, теорія, методи»*. 2016. Том 37. С. 234–240. URL: <https://periodicals.karazin.ua/ssms/article/view/8093>.
18. Ковальський Б.М. Фундаментальні цінності академічної доброчесності. *Академічна доброчесність: виклики сучасності*. Збірник наукових есе учасників наукового стажування для освітян, (Варшава, 1 – 13.10.2018). С. 123 – 129.
19. Чернявская В.Е. Плагиат как социокультурный феномен. *Известия Санкт-Петербургского университета экономики и финансов*. 2011. №3. С. 26–31.
20. Fish R., Hura G. Students' perceptions of plagiarism. *Journal of the Scholarship of Teaching and Learning*. 2013. Vol. 13. No. 5. Pp. 33 – 45.
21. Академічна культура українського студентства: основні чинники формування та розвитку. Результати всеукраїнського соціологічного дослідження. URL: <http://fond.sociology.kharkov.ua/images/presentation160715/press-reliz1.pdf>.
22. Blum S.D. My Word!: Plagiarism and College Culture. Cornell University Press, 2009. 240 p.
23. Академічна чесність як основа сталого розвитку університету / Міжнарод. благод. Фонд «Міжнарод. фонд. дослідж. освіт. політики». К.: Таксон, 2016. 234 с.

References

1. Letter of the Ministry of education and science of Ukraine to the heads of higher educational institutions "on recommendations on academic integrity for higher educational institutions". Recommendations from 23.10.2018 No.1/9-650. URL: <https://zakon.rada.gov.ua/rada/show/v-650729-18#Text>. (in Ukrainian).
2. The Law Of Ukraine "On Education". URL: <http://zakon0.rada.gov.ua/laws/show/2145-19/page3>. (in Ukrainian).
3. Recommendations for ensuring the principles of academic integrity. Sub-Commission 303 "Academic integrity" of the Scientific and methodological Commission 15 on organizational and methodological support of Higher Education. Moscow: Ministry of education and science of Ukraine, 2016. 24 p. (in Ukrainian).
4. DiVall M., Schlesselman L. Academic Dishonesty: Whose Fault is it Anyway? URL: <https://www.ncbi.nlm.nih.gov/pmc/articles/PMC4857630>.
5. Busch P., Bilgin A. Student and Staff Understanding and Reaction: Academic Integrity in an Australian University. URL: <https://doi.org/10.1007/s10805-014-9214-2>.
6. Academic integrity: compliance issues and distribution priorities among young scientists: Col. monogr. / for zag. edited by N.G. Sorokina, A.E. Artyukhov, and I.O. Degtyareva. Dnipro: DRYDU NADU, 2017. 169 p. (in Ukrainian).
7. Avdeeva N.V., Ivanova E.N. Methodology and methodology of objective assessment of borrowings in academic texts. Open education. 2020. Vol. 24. No. 6. Pp. 22 – 30. URL: DOI: <http://dx.doi.org/10.21686/1818-4243-2020-6-22-30>. (in Russian).
8. Svechkarev V. G., Ashkhamakhov K. I., Dvoynikova E. S., Guchetl I. N. Method of checking works for borrowings in the "Anti-plagiatum.vuz" system for prevention of the fraud. *Business. Education. Law*. 2020. No. 3. Pp. 395–400.
9. What percentage of similarity can be considered acceptable for academic papers? Academic integrity. January 26, 2021. URL: <https://unichек.com.ua/blog/yakyj-vidsotok-shozhosti-mozhna-vvazhaty-dopustymym-dlya-akademichnyh-robit>. (in Ukrainian).
10. Kharseeva O.V. Expediency and limits of using automated systems when checking scientific papers for the presence of borrowings. Modern Science: from Plagiarism to Academic Honesty: Materials of the All-Russian Scientific and Practical Conference, (Kursk, April 12, 2019). Pp. 142 – 148. (in Russian).
11. Appendix 2 to the Protocol No. 9(14) of 07.06.2022. Recommendations for experts to review internal policies and procedures for ensuring academic integrity and external assessment of their quality. National Agency for quality assurance of Higher Education. URL: <https://naqa.gov.ua/wp-content/uploads/2022/06/3.-Рекомендації-для-експертів-АкД.pdf>. (in Ukrainian).
12. Sevostyanova I.O. Review of systems for checking for plagiarism: all-Russian and foreign. Scientific review. Pedagogical sciences. 2017. No. 5. pp. 162 – 166. (in Russian).
13. Pobizhenko V.V., Pobizhenko I.A. Plagiarism as a factor of reducing the quality of Education. Information processing systems. Collection of scientific papers of Kharkiv Air Force University. 2011. Issue 8 (98). Pp.310–313. (in Ukrainian).
14. Zarivna N.A., Polyak O.B., Gorlachuk N.V. Academic integrity is the basis for ensuring quality education. Academic integrity: challenges of our time. Collection of scientific essays by participants of the remote stage of scientific internship for teachers, (Warsaw, 11.05 – 19.06.2020). Pp.39 – 42. (in Ukrainian).
15. Meriste H., Parder M.-L., Lõuk K., Simm K., Lilles-Heinsar L., Veski L., Soone M., Juurik M., Sutrop M. (2016). Promoting Integrity as an Integral Dimension of Excellence in Research. URL: <https://printeger.eu/wp-content/uploads/2016/10/D2.3.pdf>.
16. Mikhalev A.S. Mathematical three-dimensional model of a specialist. "Higher school": navukova-metadychny i

publitsystychny chasopis. 2015. No. 6. Pp. 19–24. (in Russian).

17. Artemov P.M., Tak I.V. Academic dishonesty as an element of academic culture of Ukrainian students: results of empirical research. Bulletin of V.N. Karazin KHNU. Series "Sociological research of modern society: methodology, theory, methods". 2016. Volume 37. Pp. 234 – 240. URL: <https://periodicals.karazin.ua/ssms/article/view/8093>. (in Ukrainian).

18. Kovalsky B.M. Fundamental values of academic integrity. Academic integrity: challenges of our time. Collection of scientific essays of participants of scientific internships for teachers, (Warsaw, 1 – 13.10.2018). Pp.123 – 129. (in Ukrainian).

19. Chernyavskaya V.E. Plagiarism as a socio-cultural phenomenon. Proceedings of the St. Petersburg University of Economics and Finance. 2011. No. 3. Pp. 26–31. (in Russian).

20. Fish R., Hura G. Students' perceptions of plagiarism. Journal of the Scholarship of Teaching and Learning. 2013. Vol. 13. No. 5. Pp. 33 – 45.

21. Academic culture of Ukrainian students: the main factors of formation and development. Results of the All-Ukrainian sociological research. URL: <http://fond.sociology.kharkov.ua/images/prezentation160715/press-reliz1.pdf>. (in Ukrainian).

22. Blum S.D. My Word!: Plagiarism and College Culture. Cornell University Press, 2009. 240 p.

23. Academic honesty as a basis for sustainable development of the University / Mezhnarod. blagod. Foundation "Mezhnarod. the foundation. research education. politicians". Moscow: Taxon, 2016, 234 p. (in Ukrainian).

DOI: <https://doi.org/10.36910/6775-2524-0560-2022-49-07>

УДК 004.93

Левицька Тетяна Олександрівна, к.т.н., доцент

<https://orcid.org/0000-0003-3359-1313>

Герасимов Дмитро, магістр

ДВНЗ «Приазовський державний технічний університет», м. Маріуполь/Дніпро, Україна

ПРОГНОЗУВАННЯ ТИМЧАСОВИХ РЯДІВ ФІНАНСОВИХ ПОКАЗНИКІВ ЗА ДОПОМОГОЮ БАГАТОШАРОВОЇ НЕЙРОННОЇ МЕРЕЖІ

Левицька Т.О., Герасимов Д.С. Прогнозування тимчасових рядів фінансових показників за допомогою багатошарової нейронної мережі. Робота присвячена моделюванню багатошарової нейронної мережі для прогнозування обсягів продажу груп товарів. Наведено аналіз публікацій про використання нейронних мереж в сфері прогнозування фінансових показників. Описано розроблене програмне забезпечення для проведення моделювання багатошарової нейронної мережі та подальшого прогнозування. Приведено приклади аналізу тимчасового ряду та формування на його основі прогнозу за допомогою двохшарової нейронної мережі.

Ключові слова: багатошарова нейрона мережа, прогнозування обсягу продажів, інформаційна система прогнозування

Levitskaya T.O., Herasymov D.S. Forecasting time series of financial indicators using a multilayer neural network. The work is devoted to modeling a multilayer neural network for forecasting sales volumes of product groups. An analysis of publications on the use of neural networks in the field of forecasting financial indicators is presented. The developed software for modeling a multilayer neural network and further forecasting is described. Examples of time series analysis and the formation of a forecast based on it using a two-layer neural network are given.

Keywords: multilayer neural network, sales volume forecasting, forecasting information system

Постановка проблеми. Сучасний процес управління неможливий без урахування маркетингових закономірностей в умовах діяльності будь-якої організації або підприємства. Обмеженість ринку збуту змушує організації швидко адаптуватися до змін довкілля, що стає важливою конкурентною перевагою підприємств, галузей, окремих регіонів і навіть країн [1-3]. Тому будь яка організація або підприємство має одну з основних цілей – це встановлення максимально можливої пропорційності та планованості у своїй діяльності. Для цього необхідно потрібне всеосяжне і продумане планування та прогнозування діяльності. Найпоширеніша діяльність – це продажі. Прогнозування обсягу продажів – важлива система, що дозволяє планування та управління торговим підприємством. За обсягом товарообігу можна будувати висновки про значимості підприємства на споживчому ринку. Попит – це потреба ринку, яка не обмежена можливостями робочої сили та технологічними потужностями підприємства.

Будь яке підприємство має у своєму розпорядженні інформацію за попередніми значеннями показників продажів та попиту. Тому можливо аналізувати дані минулих періодів і будувати прогнозні значення на майбутні зміни попиту. Прогнозування продажів розглядається як план попиту цільового ринку галузі з накладеними обмеженнями підприємства. Для цього оцінюється товарообіг – економічний показник господарської діяльності торговельного підприємства, що відображає суму продажів у грошах за певний період часу [1-3]. Такі дані описуються тимчасовими рядами, тобто послідовністю значень деяких величин, отриманих у певні моменти часу. Тимчасовий ряд включає два обов'язкові елементи – позначку часу і значення показника ряду, отримане тим чи іншим способом і відповідне зазначеній відмітці часу. Аналіз тимчасових рядів дозволяє вивчити показники у часі. Мета аналізу тимчасових рядів – витягти з цього ряду корисну інформацію, а для цього необхідно побудувати математичну модель процесу. Результатом дослідження прийнято вважати не точний прогноз, а прогноз тенденції розвитку часового ряду [1-3].

Одним з сучасних математичних методів опрацювання тимчасових рядів є апарат штучних нейронних мереж. Нейронні мережі дають додаткові можливості у моделюванні нелінійних явищ та розпізнаванні хаотичної поведінки [4-6]. Найбільш цінна властивість нейронних мереж – здатність навчатися на безлічі вхідних параметрів у тих випадках, коли невідомі закономірності та залежності між вхідними та вихідними даними. В такому разі к ним можна віднести до 80% завдань фінансового аналізу. Таким чином, для прогнозування зміни попиту товару чи послуги для навчання нейронної мережі достатньо подати реальні дані щодо ринку за різні періоди, а потім отримати прогнозоване значення.

Не дивлячись на те, що нейронні мережі добре себе зарекомендували, в даний час не має стандартних методів їх застосування. У нейронній мережі численні чинники взаємодіють дуже складним чином, тому в залежності від галузі застосування необхідно проводити моделювання - евристичний підхід підбору характеристик штучної нейронної мережі. Потрібно правильно вибрати число та вид показників, що характеризують процес. Після цього потрібно вибрати архітектуру мережі. Для використання нейронних мереж з прямим зв'язком необхідно визначити кількість прихованих елементів. Надалі, для знаходження параметрів моделі вибирають критерій помилки та навчальний алгоритм. Потім необхідно використати аналіз та перевірити різні налаштування моделі. Зрештою, необхідно інтерпретувати вихідну інформацію [1-3].

Аналіз останніх досліджень та публікацій. Сучасний стан публікацій щодо використання нейронних мереж для прогнозування економічних показників наступний. В роботі [18] автори провели дослідження для побудови прогнозуальної нейромережевої моделі для визначення динаміки ризику використання банківських установ для легалізації кримінальних коштів. Для дослідження використали багатошаровий перцептрон MLP-архітектури з використанням алгоритму BFGS, радіальна базисна функція RBF-архітектури з використанням алгоритму RBFT. Результати моделювання отримано успішні – коефіцієнт ефективності отримано в межах 0,9–1,0.

Автори роботи [19] провели порівняльний аналіз методу нейронних вимірів та методу сезонної декомпозиції часового ряду на прикладі по прогнозу обсягів продаж філії корпорації «Рошен» в Черновецькій області. Визначено переваги та недоліки використання нейронної мережі. Але зазначено, що «є нейронні мережі є потужним методом моделювання, який дозволяє відтворювати надзвичайно складні залежності».

В дослідженні [20] побудовано прогноз індексу цін на ринку нерухомості на основі застосування моделі ARIMA і нейромережі. Автори порівняли точність прогнозування: сума абсолютних та відносних помилок при прогнозуванні по нейромережі значно менше похибки, що отримується при здійсненні прогнозів за допомогою ARIMA-моделі. На думку авторів, це обумовлено тим, що нейронні мережі розраховані на застосування до рядів зі складною та нелінійною структурою ряду, тоді як модель ARIMA Бокса-Дженкінса призначена для роботи з рядами, що мають більш помітні структурні закономірності.

Стаття [21] також присвячена питанням моделювання нейронних мереж в галузі житлової нерухомості. Автори моделюють багатошарову нейронну мережу та підбирають необхідні показники для вирішення задачі оцінки ринкової вартості житла. Результати моделювання автори пропонують впровадити у систему підтримки прийняття рішень, що дозволить підвищити ефективність процесу оцінки об'єктів.

Робота [23] присвячена обґрунтуванню нейронних мереж як методу прогнозування валютних курсів. Автором вирішено, що нейронні мережі здатні оцінювати динаміку та нелінійність фінансових даних краще, ніж інші відомі методи. Представлені експериментальні результати це підтверджують. Зокрема, як модель нейронної мережі автор використовує багатошаровий перцептрон з одним прихованим шаром. Прогнозування реалізовано за допомогою одноетапного підходу, коли перцептрон перенавчається на кожному кроці експерименту. Досягнутий рівень точності є прийнятним для розробки заходів для управління валютний ризик.

Питанню прогнозування тимчасових рядів фінансової галузі також присвячено інші роботи. Робота [23] присвячена також обґрунтуванню доцільності використання штучних нейронних мереж. Авторами описано проблеми традиційних та класичних систем моделювання прогнозів, теорії нейромереж, питання вдосконалення методів аналізу та збільшення точності прогнозів фондових ринків, моделювання прогнозування фондових ринків з використанням нейронної мережі. Описано наукове обґрунтування методів застосування моделювання прогнозів на прикладі задачі прогнозування динаміки цін на фондовому ринку.

Робота [24] описує добре застосування штучних нейронних мереж для оцінки банкрутства банків, на прикладі банків України. У статті [25] досліджено можливість використання штучних нейронних мереж для прогнозування логістичних витрат. Розглянуто прогнозування на основі інформації попередніх періодів. Автор пропонує використовувати тришарову нейронну мережу прямого зв'язку з навчанням за алгоритмом зворотного поширення. Визначено оптимальну конфігурацію нейронної мережі для використання щомісячної інформації про логістичні витрати. У статті наголошується на актуальності використання підходу на різноманітних машинобудівних підприємствах.

В роботі [26] автори показали актуальність використання нейронних мереж як способу моделювання економічних процесів. Описали стан існуючих програмних засобів для роботи з
© Левицька Т.О., Герасимов Д.С.

нейромережами. Визначили, що «застосування нейронних мереж є досить потужним методом прогнозування, який дозволяє відтворювати досить складні залежності.»

В статті [27] запропоновано використання штучних нейронних мереж для прогнозування ємності фармацевтичного ринку. Автори розробили трьохшарову нейронну мережу з прямою передачею сигналів та показали результати її роботи на прикладі обсягів продажів антигельмінтних лікарських препаратів. Отримані результати мають досить високу точність. Тому автори вважають використання методів нейронних мереж доцільним в галузі фармацевтичного ринку.

Таким чином, аналіз публікацій показав, що прогнозування обсягів продажів в різних галузях з використанням багатошарових нейронних мереж для є ефективним, але передбачає моделювання та налаштування для конкретного завдання.

Метою даної роботи є дослідження і моделювання багатошарової нейронної мережі для прогнозування обсягу продажів для груп товарів.

Виклад основного матеріалу. Дані, що подаються на вхід нейромережі, повинні попередньо оброблятися з метою отримання якісного процесу навчання та відповідно якісних прогнозних значень вихідних змінних. Для проведення дослідження обрано дані обсягу продажів товарів за період з квітня 2021 року по червень 2022 року. Товари сформовано за групами, усього 15 груп товарів. На кожен день з цього періоду маємо обсяг продажів для кожної групи товару. Н рисунку 1 показано приклад розподілу обсягу товарів за кожним днем місяця. Кожен ряд (лінія) відповідає одній групі товару.



Рисунок 1 – Приклад розподілу обсягу продажу товарів за місяць

З аналізу місячних даних визначено, що є закономірність продажів, що пов'язані з вихідними днями. Тому треба враховувати тип дня для подальшого прогнозування. Оскільки має вплив вихідний день, то святковий день теж треба враховувати.

На рисунку 2 показано фрагмент динаміки продажів за період. Поглянувши на графік можна побачити, що пік продажів посідає на грудень порівняно з іншими місяцями. Аналіз даних за період з квітня 2021 року по червень 2022 року показав зміни в обсягу продажів в залежності від місяця.

Тому треба також враховувати місяць, як вхідний параметр для моделювання нейронної мережі.

На рисунку 2 також видно, що данні мають деякі екстремальні значення. Їх треба виділити для покращення майбутнього прогнозу. Таким чином, при моделюванні нейронної мережі на вхід буде подаватись: значення обсягу продажі групи товарів; день; місяць; тиждень; тип дня (звичайний або ні). Значення дати складно подати на вхід нейронної мережі тому замість нього враховується тиждень для визначення порядкового дня. Значення місяця можна подат числовим значенням від 1 до 12. Значення дня – це порядкове значення у тиждні. Тип дня можна кодувати 1 чи 0. Але усі данні будуть в програмі нормалізовано до шкали [0..1].

Для моделювання багатошарової нейронної мережі та подальшого прогнозування обсягів продажу товарів за групами було розроблено програмне забезпечення. Головне вікно програми та вікно налаштування параметрів нейронної мережі показано на рисунку 3.



Рисунок 2 – Фрагмент динаміки обсягу продажів за період

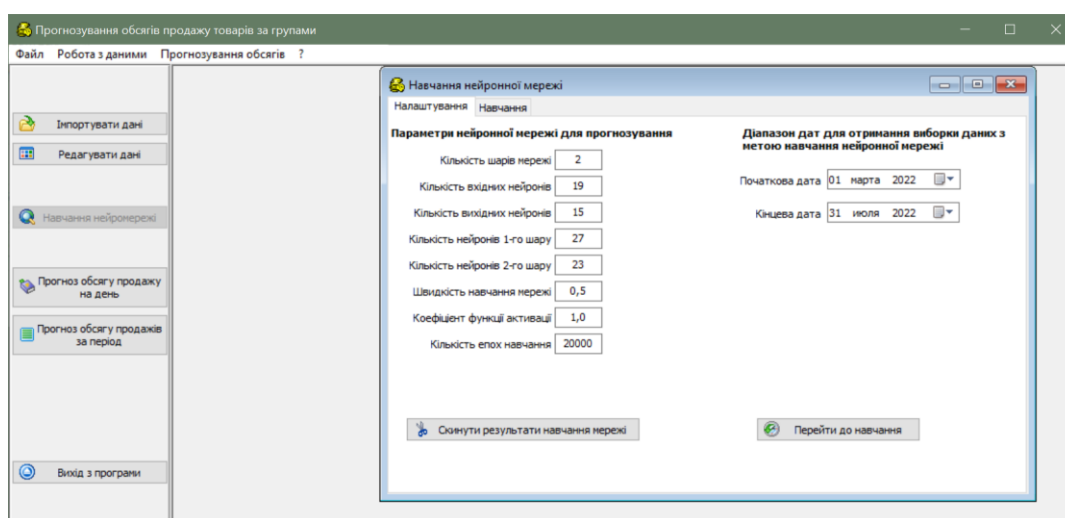


Рисунок 3 – Діаграма розсіювання після навчання.

Програмне забезпечення дозволяє завантажувати нові данні на основі тимчасових рядів продажів організації з розбивкою по днях для подальшого прогнозування; редагувати дані у системі; налаштовувати параметри нейрононої мережі (кількість шарів, кількість входів, кількість виходів тощо); надавати прогнозування обсягів продажу товарів і з урахуванням об'єднань різних груп товарів на день або на необхідний період; надавати отриману інформацію в наочному вигляді; зберігати результати прогнозування у окремий файл.

Для обраного набору вхідних даних було проведено моделювання нейронної мережі та обрання її характеристик. Оскільки немає чітких правил ні для вибору кількості прихованих шарів, ні для вибору кількості нейронів в них. Цей процес має евристичний характер тому поперше експерименти присвячені обранню архітектури багатошарової нейронної мережі. З аналізу публікацій визначено, що для прогнозування числових рядів достатньо використовувати 2 прихованих шара. Треба визначити скільки нейронів необхідно у кожному шарі. Їх не буде менш ніж початкових входів (19) тому можна почати з 21, а далі оцінювати помилку навчання. Щоб не перебирати усі значення перебирали з кроком 2. Кількість нейронів другого шару зазвичай не перевищує кількість першого шару, тому не перебирали їх значення більше ніж у першому шарі.

Результати наведено на рисунку 4, на якому стовбці відповідають кількості нейронів, а лінія показує зміну помилки. Аналіз даних показав, що однакова кількість нейронів у прихованих шарах недоцільна. Також за результатами експерименту визначено декілька наборів, які можна обрати для подальшого дослідження: перший шар -27, другий - 25; перший шар -31, другий - 25; перший шар -31, другий - 27.

На вплив навчання нейронної мережі також має кількість епох навчання. Тому другий експеримент призначено підбору цього параметру. При кількості епох навчання 2000 усі набори показали гарні результати. Коли визначено оптимальні параметри нейронної мережі можна виконувати прогнозування. Приклад формування прогнозу показано на рисунку 5.

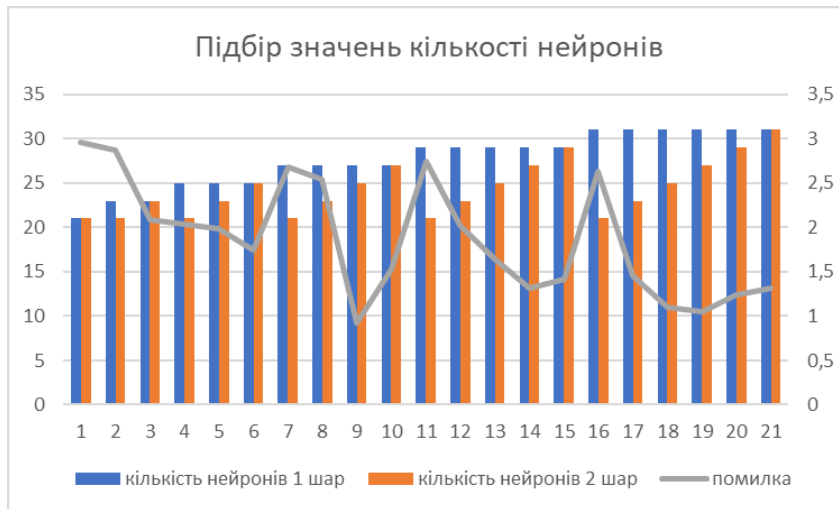


Рисунок 4 – Підбір значень кількості нейронів

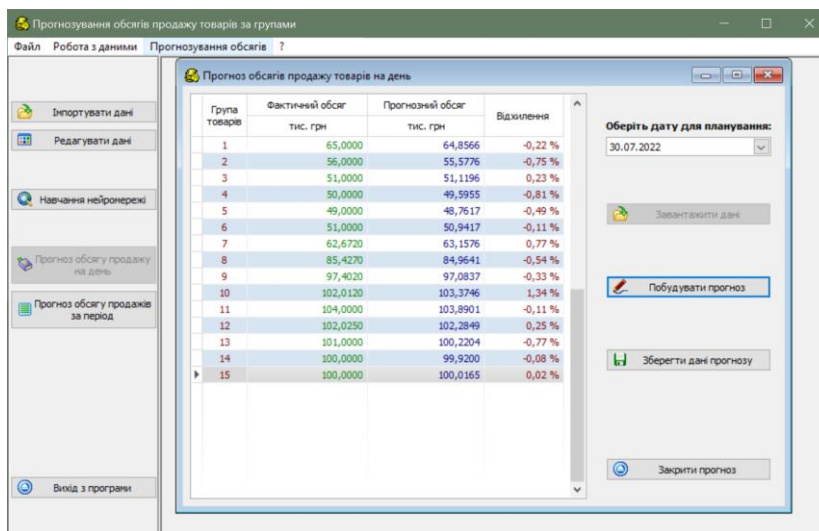


Рисунок 5 – Вікно «Прогноз обсягу продажу на день»

Висновки та напрямок подальших досліджень. Штучні нейронні мережі з успіхом справляються з вирішенням завдання та побудови прогнозів, тому що досліджувані між собою дані з вхідними та вихідними значеннями пов'язані на рівні абстракції та відсутня строга класифікація аналізованої моделі. Важлива відмінна особливість застосування нейронних мереж у вирішенні завдання прогнозування це створення власної, наперед невідомої моделі на підставі наданих даних.

В роботі запропоновано програмне забезпечення, що забезпечує ретельне дослідження обробки послідовності з акцентом на прогнозуванні часових рядів обсягів продаж із використанням моделі багатошарової нейронної мережі з двома прихованими шарами.

Розроблена інформаційна система є корисним інструментом для будь-якого спеціаліста в галузі маркетингу, який використовує послідовності та хоче передбачити подальші значення часового ряду. Розроблену інформаційну систему можна використовувати у таких сферах як фінанси, економіка, сільське господарство, прогнозування погоди.

Використання система прогнозування продажів суттєво спростить бізнес-процеси організації та дозволить внести коригування у складання плану закупівель щодо сезонності товару та залишків від продажу попереднього періоду.

Список бібліографічного опису

1. Дема Д.І. Фінансовий ринок : навч. посіб. / Житомир : ЖНАЕУ, 2017. 448 с.
2. Мних С. В. Економічний аналіз: підруч. / Київ: Знання, 2011. 630 с.
3. Бачевський Б. Є. Потенціал і розвиток підприємства: навч. посіб. / Київ: Центр учбової літератури, 2009. 400 с.
4. Субботін С. О. Нейронні мережі : теорія та практика: навч. посіб.– Житомир: Вид. О.О.Євенок, 2020. – 184 с.
5. Тимошук П. В. Штучні нейронні мережі Навчальний посібник.// Львів: Видавництво Львівської політехніки. – 2011. – 444 с.
6. Леонов, С., Кузьменко, О., Боженко, В., Мурсалов, М. Прогнозування ризику використання фінансових посередників для легалізації кримінальних доходів. *Financial and Credit Activity Problems of Theory and Practice*, 4(35), 2021. С. 191–201. <https://doi.org/10.18371/fcaptr.v4i35.222015>
7. Лукьянова В.В., Мацюк Н.О., Порівняння методу нейронних мереж і методу декомпозиції часових рядів у прогнозуванні обсягів продажу торговельного підприємства. *Науковий вісник Чернівецького університету*. 2012. Випуск 623-626. *Економіка*. С. 332-338.
8. Горчакова И. А., Ворваровская Н. Н. Прогнозирование индекса цен на рынке недвижимости средствами ARIMA-модели и нейронных сетей. *Вісник Хмельницького національного університету*. 2010. №4. С. 74–86.
9. Левицька Т. А., Железов Д. А. Моделирование нейронных сетей для оценки рыночной стоимости объекта жилой недвижимости. *Комп'ютерно-інтегровані технології: освіта, наука, виробництво*. 2020. № 41. С. 59-64.
10. Галешук С. Штучні нейронні мережі у прогнозуванні валютного ринку. *Вісник Київського національного торговельно-економічного університету*. 2016. Випуск № 3. С. 101–114.
11. Левицька Т.А., Романов К.Г. Моделирование прогнозування фондових ринків з використанням нейромереж. *Вісник Приазовського державного технічного університету*. Серія: Технічні науки. 2017. Випуск 35. С. 226-230.
12. Гадетька З. М. Оцінка ризику банкрутства банків України сучасним методом штучних нейронних мереж. *Економіка і суспільство*. 2019. Випуск № 20. С. 753-761.
13. Ковшик В. І. Використання штучних нейронних мереж для прогнозування логістичних витрат машинобудівних підприємств. *Східна Європа: економіка, бізнес та управління*. – Дніпро : ДВНЗ «Придніпровська державна академія будівництва та архітектури», 2016. № 4. С. 430–435.
14. Ставицький О. В., Мозолівська М. О. Використання нейронних мереж для прогнозування у фінансовій сфері. *Актуальні проблеми економіки та управління*. 2017. №19. С. 35–42.
15. Мнушко М., Попова Ю. В., Тіманюк І. В.. Багатошарова штучна нейронна мережа в прогнозуванні ємності фармацевтичного ринку. *Клиническая информатика и телемедицина*. 2009. Т. 5, Випуск 6. С. 69-74.

References

1. Dema D.I. Financial market: education. manual / Zhytomyr: ZhNAEU, 2017. 448 p.
2. Mnykh E.V. Economic analysis: in hand. / Kyiv: Znannia, 2011. 630 p.
3. Bachevskiy B. E. The potential and development of the enterprise: education. manual / Kyiv: Center for Educational Literature, 2009. 400 p.
4. Subbotin S. O. Neural networks: theory and practice: teaching. guide.– Zhytomyr: Ed. O.O. Evenok, 2020. - 184 p.
5. Tymoshchuk P. V. Artificial neural networks Study guide.// Lviv: Publishing House of Lviv Polytechnic. - 2011. - 444 p.
6. Lyeonov S., Kuzmenko O., Bozhenko V., Mursalov M., Zeynalov Z., Huseynova A. Forecasting the risk of money laundering through financial intermediaries. *Financial and Credit Activity Problems of Theory and Practice*, 4(35), 2021. pp. 191–201. <https://doi.org/10.18371/fcaptr.v4i35.222015>
7. Lukyanova V., Matsiuk N. Comparison of neural networks and time-series decomposition method for sales forecasting trading company. *Scientific Bulletin of Chernivtsi University*. 2012. No.623-626. Pp. C. 332-338.
8. Gorchakova I. A., Vorvarovskaya N. N. Forecasting the price index in the real estate market using the ARIMA model and neural networks. *Bulletin of Khmelnytsky National University*. 2010. №4. pp. 74–86.
9. Levitskaya T.A., Zhelezov D.A. Modeling neural networks for assessing the market value of a residential property. *Computer-integrated technologies: education, science, production*. 2020. No 41. pp. 59-64
10. Galeshchuk S. Pieces of neural networks in forecasting the currency market. *Bulletin of the Kiev National Trade and Economic University*. 2016. Issue No. 3. P. 101–114
11. Levitskaya T.A., Romanov K.G. Modelling stock markets forecasting using neural networks. No. 35 2017: Reporter of the Priazovskiy State Technical University. Section: Technical sciences (2017). No 35. Pp 226-230
12. Gadetska Z. Assessment of risk of bankruptcy of banks of ukraine by a modern method of artificial neural networks. *Economy and Society*, 2019. No 20. pp. 753-761.
13. Kovshik V.I. Selection of piece-wise neural networks for predicting the logistics of machine business enterprises. *Skhidna Evropa: economy, business and management*. – Dnipro: DVNZ “Pridniprovskaya State Academy of Life and Architecture”. 2016. No. 4. pp. 430–435.
14. Stavitsky O. V., Mozolevska M. O. Using neural for forecasting in the financial sector. *Actual problems of economy and management*. 2017. No. 19. pp. 35–42.
15. Mnushko Z. M., Popova Yu. V., Timanyuk I. V. Multi-layer artificial neural network in predicting the capacity of the pharmaceutical market. *Clinical informatics and telemedicine*./ 2009. Т. 5, No 6. pp. 69-74.

DOI: <https://doi.org/10.36910/6775-2524-0560-2022-49-08>

УДК 339.9:311.172

Ніколаєвський Олександр Юрійович, к.т.н., доцент

<https://orcid.org/0000-0002-0786-5432>

ПВНЗ «Європейський університет», м.Київ, Україна

БІЗНЕС-ІНФОРМАЦІЙНІ ТЕХНОЛОГІЇ ЯК ГОЛОВНИЙ АСПЕКТ КОНКУРЕНТНОЇ ПЕРЕВАГИ СУЧАСНОГО БІЗНЕСУ

Ніколаєвський О.Ю. Бізнес-інформаційні технології як головний аспект конкурентної переваги сучасного бізнесу. У статті розкрито бізнес-інформаційні технології, як головний аспект конкурентної переваги сучасного бізнесу. Використання інноваційних інформаційних технологій дозволяє підприємству значно швидше розвиватися та отримувати більший прибуток. Наголошено, що для того, щоб вижити в сучасних умовах, бізнес підприємства повинні змінити вже існуючу структуру та здійснювати узгодженість між бізнес стратегіями та ІТ-технологіями. В умовах сьогодення скорочення часу стає основою отримання конкурентних переваг. Бізнес-стратегії, засновані на скороченні часу приносять максимальні конкурентні переваги. Зазначається, що в сучасних умовах розвитку світових ринків ключовим фактором, що визначає конкурентоспроможність підприємства, є наявність у нього унікального інформаційного ресурсу, який ґрунтується на бізнес стратегії підприємства та враховує основні показники сфери діяльності бізнесу. Тільки наявність інформаційного ресурсу не є джерелом конкурентної переваги. Володіння та розпорядження унікальним інформаційним ресурсом надає можливості бізнес підприємствам удосконалювати використання традиційних факторів виробництва та економічних ресурсів, або комбінувати їх з новими ресурсами, що забезпечує більшу результативність порівняно з конкурентами. На рівні підприємства впровадження бізнес-інформаційних технологій дає можливість швидшого збирання та аналізу інформації для прийняття стратегічних управлінських рішень. Інформаційні технології як інструмент допомагають знизити витрати, збільшити прибутковість підприємства, підвищити продуктивність праці за умови повної узгодженості зі стратегіями підприємства. Підкреслено, що використання бізнес-інформаційних технологій дає підприємству суттєву перевагу перед конкурентами, дозволяючи ефективніше розподіляти ресурси компанії, керувати персоналом, скорочувати витрати тощо. Завдяки інвестиціям в бізнес-інформаційні технології підприємства мають можливість змінити свій бізнес і в результаті збільшити частку ринку, зайняти нову нішу, налагодити тіснішу співпрацю з клієнтами та постачальниками.

Ключові слова: бізнес, інформаційні технології, управління підприємством, безпека, ІТ-стратегія, бізнес-модель, ризики, великі дані.

Nikolaievskiy O. Business information technologies as the main aspect of the competitive advantage of modern business. The article reveals business information technologies as the main aspect of the competitive advantage of modern business. The use of innovative information technologies allows the enterprise to develop much faster and earn more profit. In order to survive in modern conditions, business enterprises must change the already existing structure and achieve consistency between business strategies and IT technologies. In today's conditions, time reduction becomes the basis for obtaining competitive advantages. Business strategies based on time reduction bring maximum competitive advantages. It is noted that in the current conditions of the development of world markets, the key factor determining the competitiveness of an enterprise is the availability of a unique information resource, which is based on the enterprise's business strategy and takes into account the main indicators of the business sphere. Only the availability of an information resource is not a source of competitive advantage. Possession and management of a unique information resource provides opportunities for business enterprises to improve the use of traditional factors of production and economic resources, or to combine them with new resources, which ensures greater effectiveness compared to competitors. At the enterprise level, the implementation of business information technologies enables faster collection and analysis of information for making strategic management decisions. Information technology as a tool helps to reduce costs, increase the profitability of the enterprise, and increase labor productivity, provided that it is fully aligned with the strategies of the enterprise. It is emphasized that the use of business information technologies gives the enterprise a significant advantage over competitors, allowing more efficient allocation of company resources, personnel management, cost reduction, etc. Thanks to investments in business information technologies, enterprises have the opportunity to change their business and, as a result, increase their market share, occupy a new niche, establish closer cooperation with customers and suppliers.

Key words: business, information technologies, enterprise management, security, IT strategy, business model, risks, big data.

Вступ та постановка проблеми. Технологічні зміни сучасності породили нові способи, за допомогою яких підприємства можуть створювати бізнес-моделі направлені на ефективне здійснення діяльності. Для суб'єктів сучасного бізнесу важливо бути в курсі нових інформаційних технологій і трендів. У той же час, для того, щоб інформаційні технології (ІТ) додавали цінності суб'єкту господарювання, важливо досягти стратегічного узгодження між ІТ та бізнес-стратегіями підприємства. Феномен інформатизації широко увійшов у бізнес сферу тим, самим сприяючи розвитку сучасного бізнесу. Масові інформаційні транзакції та виробничі дані генеруються різними системами інформатизації на підприємстві виводячи бізнес ринок на нову ланку. Враховуючи розвиток сучасних підприємств як комерційного так і некомерційного характеру, варто наголосити, що важливі виробничі дані в інформаційній системі підприємства можуть

зберігатися лише протягом одного місяця. У разі виникнення потреби на довший час зберігання, підприємство несе масові витрати на зберігання. Крім того, для роботи з великою кількістю інформації видалення необхідних даних є дуже важливим для управління виробництвом підприємства. Однак традиційні технології управління даними та інтелектуального аналізу даних, які масово використовуються, не здатні ефективно зберігати та аналізувати великі обсяги даних [1], вони потребують модернізації. У процесі управління бізнес-інформацією ефективно зберігання та використання великих даних може ефективно знизити вартість операцій інформаційних технологій підприємства [2]. Крім того, працівники підприємства (аналітики, менеджери, керівники) можуть проводити глибокий аналіз даних, формувати кількісні рішення та технічну підтримку для управління підприємством, покращувати ефективність виробництва та продажів і, зрештою, покращувати економічну ефективність загалом.

У зв'язку зі стрімким збільшенням обсягу бізнес-інформаційних даних існує нагальна потреба використовувати технологію обробки великих даних у сучасному управлінні бізнес-інформацією, щоб виявити потенційно придатну для використання інформацію, яка може не тільки забезпечити підтримку даних для керівництва підприємства під час прийняття рішень, але й надати кращі послуги споживачам продукції.

Головною метою інформаційних технологій виступає підвищення цінності бізнесу через інвестиції як в інформацію, так і в технології, а також узгодження бізнесу з ІТ-стратегіями на двох рівнях. По-перше, підприємства повинні отримати вигоду від технологічних розробок, включивши їх у бізнес-модель. По-друге, технологічна архітектура підприємств має забезпечувати реалізацію їх загальної бізнес-стратегії.

Аналіз останніх досліджень і публікацій. Формулювання наукової думки в окреслі глибокої інтеграції, інтеграції вітчизняної електронної комерції та логістичної індустрії базується на новітніх технологіях в інформаційній сфері, які поступово поєднуються в дослідженнях інформатизації логістики [3, 4]. Простежується значна кількість перехресних досліджень вітчизняних логістичних інформаційних технологій та технологій хмарних обчислень [5]. Розвиток бізнес підприємств залежить від розвитку електронної комерції та сприяє розвитку останньої [6, 7]. Враховуючи принципи реалізації оригінальної індустріальної системи, логістика, комерційний потік та інформаційний потік поєднанні зі структурою сучасного підприємства і розподілом товарів, які разом формують соціалізовану систему логістики та розподілу [8].

Технологія великих даних використовувалася багатьма дослідниками в управлінні бізнес-інформацією [9]. Машинне навчання, в силу своєї масштабності дії, може допомогти керівництву обробляти комерційні дані для прийняття суджень і рішень щодо ключових питань операцій, оптимізації інформаційних систем логістики, оптимізації послуг, економії місця та контролю над запасами [10]. У розвитку бізнесу ключові технології, такі як хмарні обчислення та аналітика великих даних, можуть бути використані для кращого управління сучасною бізнес-інформацією [11]. Однак дослідникам, які працюють у сфері комп'ютерних наук у цій міждисциплінарній галузі дослідження, бракує знань про управління бізнес-довірою. Тому конкретного застосування когнітивної науки та технології великих даних в управлінні бізнес-інформацією досягти проблематично [12]. Дослідницький фундамент інформаційних технологій в управлінні бізнес-інформацією є відносно слабким, що призводить до повільного розвитку існуючих хмарних обчислень і технологій аналізу великих даних в управлінні бізнес-інформацією.

Із зарубіжних авторів варто відзначити такі роботи як: Shen, Ching-Cheng & Yeh, Chien-Chi & Lin, Chun-Nan [13], Otuz, Evans [14], Patrisia, Dina & Linda, Muthia & Abror, Abror [15], Ikhwana, Andri & Dianti, Sasi [16], Luo, Dean & Shahbaz, Muhammad & Qureshi, Muhammad & Anis, Maira & Mahboob, Farhan & Kazouz, Nayfa & Mao, Jie [17], Muazu, Umar & Abdulmalik, Sambo [18] та інші.

Однак, незважаючи на масштабність наукових досліджень за окресленою тематикою, питання розкриття бізнес-інформаційних технологій, як головного аспекту конкурентної переваги сучасного бізнесу залишається відкритим та потребує детального опрацювання.

Постановка завдання. Розкрити бізнес-інформаційні технології, як головний аспект конкурентної переваги сучасного бізнесу.

Викладення основного матеріалу дослідження. Впровадження ефективних бізнес-інформаційних технологій для досягнення конкурентної переваги сучасного бізнесу, має бути делеговано керівництву підприємства. Стратегічні бізнес-цілі суб'єкта господарювання, яким виступає підприємство, досягаються засобами як основних бізнес-припущень, так і бізнес імперативами. Основні бізнес-припущення представляють загальні цілі, які не є унікальними для

конкретних підприємств сучасного бізнесу та впровадження базових ІТ рішень і внутрішнього контролю, що достатньо для ефективного управління діяльністю.

Основні бізнес-припущення стосуються способу управління діяльністю суб'єкта господарювання, впровадження має важливе значення для того, щоб суб'єкт господарювання міг ефективно та результативно виконувати свої основні повсякденні завдання в конкретному бізнес-середовищі. Приклади базових бізнес-припущень включають, серед іншого, те, що всі суб'єкти господарювання орієнтовані на отримання прибутку, здійснюють транзакції в усіх бізнес-процесах, дотримуються загальних законів і чинних нормативних актів, застосовних до всіх суб'єктів господарювання, і враховують потік грошових коштів і безперервність бізнесу. Базові ІТ-рішення та внутрішній контроль впроваджуються для того, щоб керувати основними бізнес-припущеннями, однак це не є узгодженням бізнесу та ІТ.

Бізнес-імперативи виходять за рамки базових бізнес-припущень і є тими важливими рушійними силами, які мають бути впроваджені в організації та виконані на максимальному рівні, щоб організація досягла своїх стратегічних цілей. Бізнес-імперативи діють як ключові рушійні сили суб'єкта, і їх успішне виконання дасть суб'єкту конкурентну перевагу в його специфічному середовищі. Отже, бізнес-імперативи залежать від контексту організації та визначають технологічну архітектуру, управління якою є головною умовою.

Діяльність організації здійснюється з урахуванням різних чинників як внутрішнього так і зовнішнього характеру. Внутрішні чинники можна узагальнити у відповідності до бізнес-моделі підприємства, на яку, у свою чергу, постійно впливають зовнішні сили, що постійно змінюються, такі як конкуренція, соціальні, правові чи технологічні зміни, що оточують суб'єкт господарювання, це робить бізнес-модель кожного суб'єкта господарювання унікальною.

Бізнес-модель – це інструмент, який містить елементи суб'єкта господарювання та зв'язки між цими елементами. Він забезпечує бізнес-контекст і механізми досягнення стратегічних цілей суб'єкта господарювання і, зрештою, представляє унікальну логіку заробітку суб'єкта господарювання у формі його відносин із клієнтами, ціннісних пропозицій та ціннісних мереж взаємодії.

Управління ІТ має вирішальне значення, щоб гарантувати, що впроваджені ІТ-рішення забезпечують заплановану цінність для суб'єкта господарювання, а також щоб ІТ підтримували стратегії та цілі суб'єкта господарювання. Одним із способів досягнення ефективного управління ІТ є впровадження структури управління. Існує багато систем управління ІТ, кожна з яких має різну мету.

Впровадження бізнес-інформаційних технологій допомагає організаціям досягати цілей управління ІТ і відіграє вирішальну роль в узгодженні цілей бізнесу та ІТ. Мета стимулюючих процесів і практик управління може бути узгоджена з бізнес-імперативами. Успішне узгодження визначить бізнес-імперативи, які служать як бізнес-цілями, так і цілями ІТ-відділу організації, і можуть стати основою для подолання розриву в ІТ, щоб остаточно досягти узгодженості між бізнесом та ІТ на рівні технологічної архітектури. Стратегічне узгодження, тобто узгодження між бізнесом та ІТ-цілями, є однією з основних сфер управління ІТ і, по суті, може розглядатися як ключ до досягнення ІТ-управління. Стратегічне узгодження дозволить суб'єкту господарювання отримати бажану цінність від своїх інвестицій, пов'язаних з ІТ.

Для того, щоб успішно визначити спосіб, у який технологічна архітектура суб'єкта має бути спроектована, щоб забезпечити досягнення загальних бізнес-цілей суб'єкта, бізнес-цілі необхідно перетворити на цілі ІТ-відділу. Потім ці цілі повинні стати основою ІТ-стратегії, щоб сприяти успішному узгодженню з бізнес-стратегією. Для успішного виконання цього процесу необхідна чітка комунікація між вищим керівництвом та ІТ-фахівцями. Однак між вищим керівництвом та ІТ-спеціалістами часто виникають непорозуміння внаслідок нерозуміння ІТ-спеціалістами бізнес-цілей і важливості управління ІТ, а також нерозуміння вищим керівництвом ІТ. Визначення вичерпного переліку бізнес-імперативів, що мають відношення до суб'єкта господарювання, і визначення того, як технологічну архітектуру суб'єкта господарювання можна адаптувати для підтримки цих бізнес-імперативів, полегшить спілкування між вищим керівництвом та ІТ-спеціалістами.

Бізнес-імперативи для успішного впровадження бізнес-інформаційних технологій:

– висока продуктивність: наприклад, у середовищі масового ринку висока пропусканна здатність продуктів або послуг є важливим показником економічної ефективності;

- орієнтація на клієнта: суб'єкти господарювання аналізують клієнтів, щоб визначити, як слід розробляти продукти чи адаптувати послуги. Одним із способів досягнення цього є цілеспрямований маркетинг по відношенню до клієнтів, що відповідає їхнім конкретним вимогам і потенційна адаптація продуктів чи послуг для задоволення потреб клієнтів;
- низькі витрати: зменшення операційних витрат і/або витрат на продукцію може призвести до більшої норми прибутку або продукції та послуги можуть бути надані за нижчою вартістю, ніж у конкурентів суб'єкта господарювання. Це робить продукти або послуги більш доступними для клієнтської бази;
- різноманітність продуктів або напрямків діяльності: певні суб'єкти господарювання залежать від різних потоків доходу різноманітних сегментів споживачів, щоб залишатися прибутковими, наприклад ті, хто надає сезонні продукти або послуги, і суб'єкти, які надають продукцію з мінімальними витратами або без прибутку, надаючи при цьому додаткові безкоштовні продукти або послуги, які є більш прибутковими;
- продуктивність та ефективність: певні суб'єкти господарювання, наприклад ті, що працюють в обробній промисловості, повинні продуктивно та ефективно керувати своїми операціями, тому, що проста допомога клієнтам у «виконанні роботи» може бути важливою частиною бізнес-моделі суб'єкта господарювання;
- скорочення часу доставки/циклу: суб'єктам господарювання необхідно мінімізувати час від етапу розробки нового продукту до його доставки на ринок, а також скоротити час, витрачений на дії, що не додають вартості, щоб забезпечити доступність продуктів;
- орієнтація на продукт: суб'єкти зосереджуються на можливостях, навичках і наявних ресурсах всередині суб'єкта, щоб визначити, як необхідно розробляти продукти чи послуги;
- інновації: суб'єкти господарювання, які прагнуть бути інноваційними лідерами (також їх називають першими), повинні першими, серед іншого, виробляти нові продукти, надавати інноваційні послуги, застосовувати певні процеси або розробляти спеціальні стандарти;
- швидкий послідовник: стратегія отримання прибутку швидкого послідовника полягає у використанні продуктів, послуг, стандартів або процесів, розроблених першими, щоб отримати від них вигоду;
- швидка адаптивність: суб'єкти господарювання, а також їхні працівники повинні бути більш винахідливими, адаптованими та толерантними до невизначеності, щоб функціонувати на оптимальному рівні в сучасних умовах, що швидко змінюються;
- розподілені філії та процеси: наявність кількох філій для охоплення ширшої клієнтської бази, можливо розподіленої по широкій географічній території, має вирішальне значення для деяких організацій, наприклад для роздрібних точок. Це може включати власні магазини суб'єкта господарювання або партнерські магазини та інші канали, як-от веб-сайти, що належать партнерам;
- розподілені/гібридні проектні групи: у деяких галузях необхідно, щоб члени команди, залучені до одного проекту, працювали з різних місць. Члени гібридних проектних команд працюють над спільною метою та працюють переважно віртуально та лише час від часу спілкуються особисто;
- низький рівень необхідних/доступних навичок: напівкваліфіковані або некваліфіковані працівники, заробітна плата сприяє нижчій базі витрат, будуть найняті для виконання рутинних завдань в організаціях, які прагнуть отримати низькі витрати на заробітну плату;
- зменшення часу простою/надійності систем: у середовищі електронної комерції простої системи можуть бути шкідливими для успіху організації. В інших середовищах, таких як роздрібна торгівля чи виробництво, системи, необхідні для роботи систем торгових точок і виробничого обладнання, є обов'язковими для успіху підприємства;
- висококваліфікована робоча сила/децентралізація повноважень: моральний дух працівників підвищується, коли працівникам дозволяється приймати рішення без нагляду чи схвалення керівництва, що скорочує час циклу, покращує гнучкість процесу та підвищує результативність і ефективність процесів;
- робота з будь-якого місця: деякі галузі вимагають, щоб її працівники працювали з різних місць. Таким чином, співробітники повинні мати можливість доступу та обміну інформацією з різних місць;

– співпраця/інтеграція: інтеграція непов'язаних додатків для безперервного виконання наскрізних бізнес-процесів, обмежуючи людське втручання, дозволить співробітникам зосередитися на основній діяльності бізнесу;

– простота використання: процеси та системи, такі як торговельні термінали в роздрібних магазинах, мають бути простими у використанні та/або однаковими для всіх місць, наприклад у галузях, які стикаються з некваліфікованою робочою силою, або підприємствах із географічно розподіленими філіями. Платформи електронної комерції також повинні бути зручними для користувача, оскільки клієнти повинні мати можливість переміщатися по платформі без допомоги самої організації;

– самообслуговування: надання клієнтам можливості допомогти собі, особливо під час виконання монотонних, однозначних завдань, стало більш популярним серед споживачів, ніж обслуговування на місці, оскільки це зручніше, і клієнти відчувають більший контроль над обслуговуванням, яке вони здійснюють. Це також може підвищити продуктивність в організації та заощадити витрати, пов'язані з наданням послуг;

– автоматизація: автоматизація та використання таких технологій, як великі дані та когнітивна аналітика, призведуть до меншого втручання людини, що збільшить важливість рішень, прийнятих невеликою кількістю працівників, залучених до процесу прийняття рішень. Дві переваги автоматизації, це зниження собівартості продукції та підвищення якості продукції;

– персональна допомога: певні організації покладаються на людську взаємодію, щоб допомогти своїм клієнтам під час або після транзакцій продажу чи надання послуг. Ця взаємодія може мати форму, серед іншого, взаємодії віч-на-віч, наприклад, у торгових терміналах, кол-центрах або електронною поштою. У деяких випадках цілеспрямована особиста допомога є центральною для бізнес-моделі суб'єкта господарювання, наприклад, призначення приватного банкіра клієнту банку;

– надійність інформації: інтелектуальні ресурси, включаючи, серед іншого, бази даних клієнтів і власні знання, стають критичними елементами в бізнес-моделях певних організацій. Суб'єкти можуть залежати від актуальної інформації в режимі реального часу, щоб допомогти їм у процесах прийняття рішень і операціях. Інвестиційні суб'єкти, наприклад, не можуть дозволити собі приймати рішення щодо інвестування на основі застарілої інформації, тоді як роздрібні онлайн-магазини повинні оновлювати наявність товарних запасів, щоб адекватно керувати очікуваннями клієнтів;

– навички використання важелів: знання є найважливішим активом і рушійною силою економічного внеску в епоху інформації. У середовищі, де обмін знаннями є надзвичайно важливим, потрібно використовувати навички та знання, щоб працювати максимально ефективно та результативно. Окрім обміну знаннями між працівниками, певні організації покладаються на спільноти користувачів, щоб допомагати один одному у вирішенні проблем і навіть спільному створенні контенту;

– високоефективні команди: використання високоефективних команд є важливим для виконання складних проектів;

– мобільний доступ клієнтів: суб'єкти господарювання повинні надати клієнтам доступ до інформації про продукти та послуги суб'єкта господарювання з їхніх мобільних пристроїв;

– додана вартість, надійність та якість продуктів і послуг: продажні ціни можуть бути підвищені в обмін на ексклюзивні продукти або послуги вищої якості;

– керовані процеси: управління бізнес-процесами забезпечує необхідну координацію між бізнес-процесами та призведе до підвищення ефективності процесів і підвищення рівня продуктивності підприємства.

Менеджери з інформаційних технологій можуть впроваджувати системи та технології з використанням певної архітектури для досягнення конкретного бізнес-імперативу, що служить і ІТ, і бізнес-ціллю.

Розробка ІТ-архітектури відповідно до бізнес-імперативів забезпечить досягнення бізнес-імперативів як цілей ІТ-відділу. Це, у свою чергу, підтримуватиме бізнес-цілі суб'єкта господарювання. Рисунок 1 ілюструє процес, якого має дотримуватися керівництво, щоб успішно досягти узгодження бізнесу та ІТ на рівні технологічної архітектури.

Першим кроком у використанні бізнес-імперативів, визначених у цьому дослідженні як ціль для ІТ-відділу, і проектування технологічної архітектури відповідно до бізнес-імперативів, є

визначення цілей, що відповідають бізнес-цілям і стратегії суб'єкта. Вони стануть основою для ідентифікації тих бізнес-імперативів, які стосуються конкретної організації. Кожен суб'єкт повинен мати максимум п'ять-сім бізнес-імперативів.

Другим кроком для вищого керівництва є визначення практичних оперативних заходів, які мають бути впроваджені з точки зору бізнесу, щоб реалізувати бізнес-імперативи, визначені як ключові рушійні сили їхньої організації.

Після встановлення оперативних бізнес-заходів вище керівництво та ІТ-фахівці можуть визначити вплив на технологічну архітектуру організації шляхом оцінки того, як ІТ можуть підтримувати ці оперативні бізнес-заходи (етап 2), необхідні для реалізації бізнес-імперативів. Цей процес гарантує, що розробка технологічної архітектури суб'єкта втілює в життя його бізнес-імперативи та, зрештою, забезпечує узгодженість між стратегічними цілями бізнесу та його ІТ-стратегією з метою усунення ІТ-розриву.

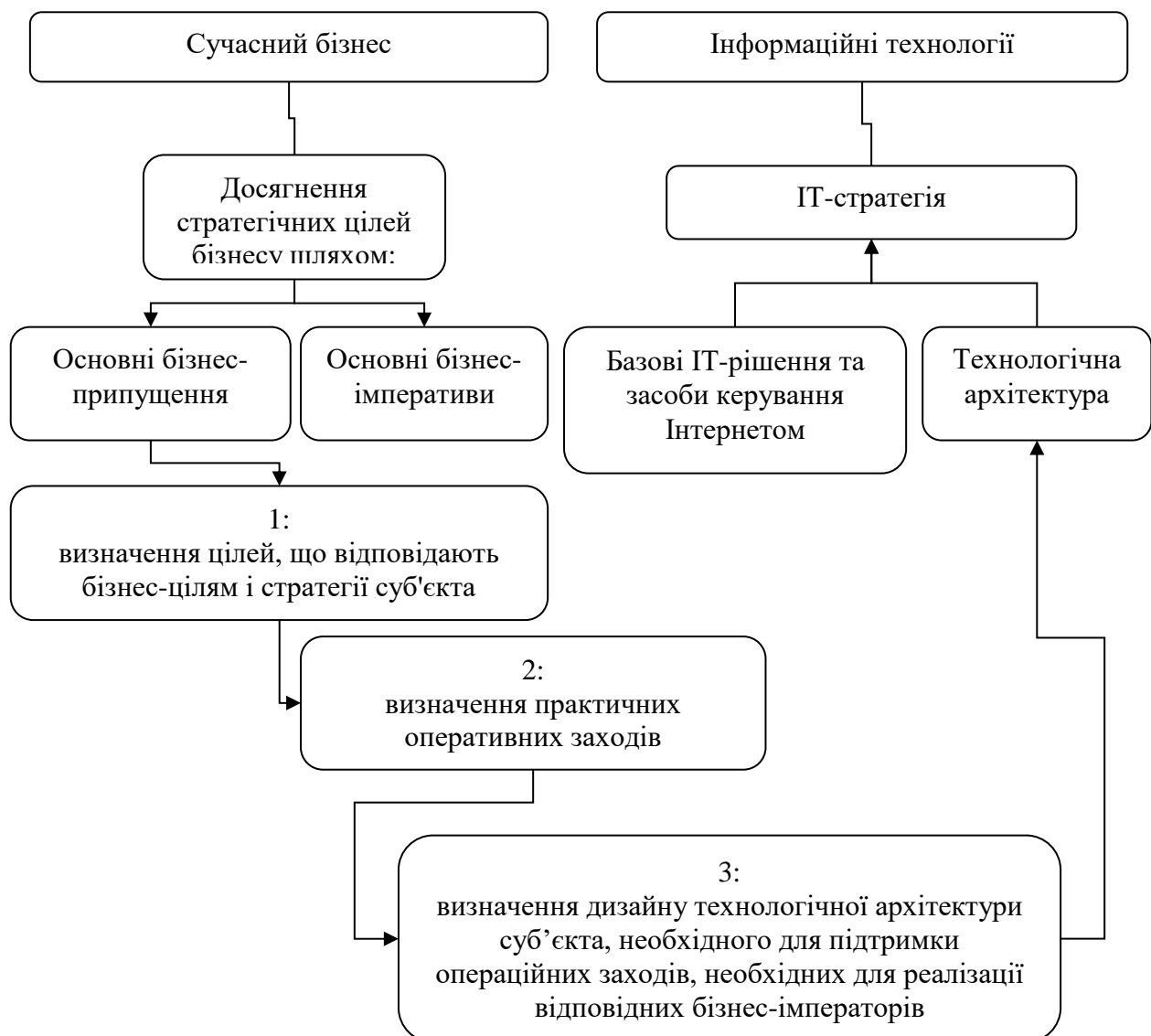


Рисунок 1 – Процес узгодження сучасного бізнесу та інформаційних технологій

Беззаперечно, що функціонування бізнес-інформаційних технологій ґрунтується на великих даних. В умовах сьогодення, для зберігання великих даних використовується сховище даних гібридного типу, яке є складною системою зберігання інформації, а модель аутсорсингу багатокористувацького сховища даних стала основним сценарієм застосування. Однак результати аналізу безпеки даних показують, що безпека поступово стала помітною перешкодою для розвитку розподілених систем зберігання та хмарних додатків для зберігання [19]. На даний момент лише 20% користувачів готові розміщувати приватні дані в хмарі або розподіленій системі зберігання, а 50% користувачів хочуть зберігати в хмарі лише резервні копії даних і дані для

© Ніколаєвський О.Ю.

аварійного відновлення. Тому створення ефективної моделі безпеки даних для багаторівневих гібридних розподілених систем зберігання стало новим викликом для сучасних дослідників.

У багаторівневій гібридній розподіленій системі зберігання є дві фундаментальні проблеми:

1) оптимістичне припущення про довірений домен розподілених систем зберігання призводить до ігнорування атак і загроз зсередини системи зберігання;

2) складний механізм керування зберіганням інформації багаторівневої гібридної системи.

Комплексне керування даними потребує відповідного механізму захисту даних для спільної роботи. Безпека даних у сховищі даних гібридного типу гарантується кількома рівнями ключів. Ця схема не тільки підвищує безпеку даних, але й знижує вартість зв'язку під час використання та обслуговування ключів.

Контроль конфіденційності даних стосується двох ключових проблем:

1) структурного дизайну файлів даних;

2) керування ключами в середовищі розподілених ключів.

Для структурного проектування варто враховувати, що файл даних логічно розділений на дві частини у вузлі зберігання даних: файл метаданих і файл даних. Файл метаданих (mod-файл) зберігає атрибути, пов'язані з безпекою даних і керуванням зберіганням, наприклад інформацію про доступ, пов'язаний список кореневого хешу, популярність файлу даних і розташування вузла зберігання. Файл даних (d-файл) зберігає зашифровані дані файлу. По-друге, керування ключами сховища даних гібридного типу використовує ієрархічну схему керування, яка не тільки підвищує безпеку даних, але й знижує вартість зв'язку в процесі використання та обслуговування ключів.

Модель безпеки сховища даних гібридного типу має реалізувати ефективну інтеграцію площини управління сховищем і площини контролю безпеки даних; необхідно реалізувати спільне управління та синхронізацію даних двох площин. Спільне управління реалізується за допомогою синхронізації даних за подіями.

Масовий розподіл даних сприяє підвищенню якості управління бізнес-інформацією. Переваги механізму масового розподілу ґрунтуються на:

– можливість швидко збирати дані в реальному часі, хоча дані в операційному середовищі надзвичайно складні;

– розрідженість високорозмірних даних;

– поєднання даних із багатьох різнорідних джерел. Основними типами даних є неструктуровані та напівструктуровані потокові дані, які сприяють зберігання та обробці.

Дані в бізнес галузі дуже складні, що призводить до серйозних проблем, таких як шум і надлишкові дані. Для вирішення вищезазначених проблем необхідно застосовувати рекомендаційний механізм дистрибуційних послуг.

В умовах масивних бізнес-даних розрідженість спільних рекомендаційних систем більш очевидна, а оскільки типи даних стають складнішими, проблема надлишкових і зашумлених даних стає серйознішою. Обробка поточкових даних у реальному часі також висуває нові вимоги до обчислювальної продуктивності рекомендованих систем. Щоб покращити якість роботи із розрідженими даними великої розмірності, необхідно зменшити чутливість надлишкових і зашумлених даних та зменшити обчислювальну складність алгоритму.

Базуючись на перевагах користувача та характеристиках ресурсів, варто наголосити на моделі управління бізнес-інформацією, що характеризується векторами тегів які, у першу чергу, повинні бути персоналізованими, необхідно оцінювати та обмежувати вагу міток.

Врахування видів та значень потоків великих даних, що циркулюють у рамках суб'єкта господарювання, покращена комунікація та ідея роботи для досягнення спільної мети зменшать IT-розрив і забезпечать узгодження бізнесу та IT на стратегічному рівні. Після того, як технологічна архітектура буде розроблена відповідно до бізнес-цілей організації, IT-фахівці зможуть створювати, налаштовувати, конфігурувати, експлуатувати та підтримувати окремі компоненти технологічної архітектури таким чином, щоб недоліки в IT було подолано на операційному рівні.

Висновки. У роботі розкрито бізнес-інформаційні технології, як головний аспект конкурентної переваги сучасного бізнесу. Було проведено дослідження процесу формування бізнес IT-стратегії на підприємстві, визначено основні складові впровадження інформаційних технологій, принципи аналізу великих даних. Схема паралельної реалізації рекомендаційного алгоритму великих даних має певну універсальність і може бути застосована до різних типових

алгоритмів машинного навчання та аналізу даних в умовах впровадження бізнес-інформаційних технологій на базі сучасного бізнесу. Система управління бізнесом представлена у цьому дослідженні може бути використана для управління бізнес-інформацією сучасних підприємств.

Перспективами подальшого дослідження є розробка алгоритму впровадження інформаційних технологій на підприємстві із застосуванням штучного інтелекту.

Список бібліографічного опису

1. Самойленко Л. Б. Можливості та проблеми застосування технологій big data вітчизняними компаніями. Ефективна економіка. 2018. № 1. – URL: <http://www.economy.nayka.com.ua/?op=1&z=6066> (дата звернення: 21.11.2022).
2. Інформаційні технології в бізнесі. Частина 1: Навч. посіб. / [Шевчук І.Б., Старух А.І., Васків О.М. та ін.]; за заг. ред. І.Б. Шевчук. Львів: Видавництво ННВК «АТБ», 2020. 455 с.
3. Підгорний М. В., Лукьянченко О. Ю. Дослідження інформатизації логістичних процесів як комплексу завдань програмування життєвого циклу автотранспортного підприємства / Вчені записки ТНУ імені В.І. Вернадського. Серія: Технічні науки, 2021. № 32(71)-2. С. 200-205. DOI: 10.32838/2663-5941/2021.2-2/31
4. Підгорний М.В. Веретюк С.М. Системний підхід до управління життєвим циклом інформаційно-комунікаційної технології. Системи управління, навігації та зв'язку. Полтава, 2021. Вип. 1(63) С. 100–103.
5. Буслів П.В. Моделі та методи диференціації даних консолідованої інформації для систем підтримки рішень. – Кваліфікаційна наукова праця на правах рукопису. Дисертація на здобуття наукового ступеня кандидата технічних наук за спеціальністю 05.13.06 – інформаційні технології. –Черкаський державний технологічний університет, Черкаси, 2021. 184 с.
6. Боковець В.В., Давидюк Л.П. Електронна торгівля її значення для розвитку бізнесу. Вісник Хмельницького національного ун-ту. Серія: Економічні науки. Хмельницький: Хмельницький національний університет. 2021. №1. С.210-214.
7. Ковтун Е. О. Електронна торгівля в умовах міжнародної економічної діяльності / Вінницький торговельно-економічний інститут ДТЕУ, 2021. № 2(298). С. 228-229. DOI: 10.31891/2307-5740-2021-298-5(2)-38
8. Коцко Т.А. Розвиток підприємства на засадах концепції реверсивної логістики: система мотивів та обмежень / Бізнес, інновації, менеджмент: проблеми та перспективи: зб. тез доп. II Міжнар. наук.-практ. конф., 22 квіт. 2021 р. – Київ: КПІ ім. Ігоря Сікорського, Вид-во «Політехніка», 2021. – С. 264-265.
9. Панченко О. Д. Big Data як основа інформаційного забезпечення реалізації аналітичної функції управлінського обліку в рослинництві. Бізнес-навігатор. 2020. № 2 (58). С. 115–119.
10. Ліщинська Л.Б., Добровольська Н.В. Перспективні програмні інструменти для аналізу даних у бізнесі / Вісник Хмельницького національного університету. 2022. №1. С.78-83.
11. Пономаренко І. В., Телеус А. В. Бізнес-аналітика як ефективний інструмент обробки даних. Проблеми інноваційно-інвестиційного розвитку. 2020. № 23. С. 64-70.
12. Ліщинська Л.Б. Впровадження Business Intelligence для ефективного управління підприємством. І Науково-технічна конференція підрозділів Вінницького національного технічного університету (2021): збірник доповідей (Вінниця, 10-12 березня 2021). Вінниця: ВНТУ, 2021. URL: <https://conferences.vntu.edu.ua/index.php/all-fitki/all-fitki-2021/paper/view/12330/10310>.
13. Shen, Ching-Cheng & Yeh, Chien-Chi & Lin, Chun-Nan. (2022). Using the perspective of business information technology technicians to explore how information technology affects business competitive advantage. Technological Forecasting and Social Change. 184. 121973. 10.1016/j.techfore.2022.121973.
14. Otuza, Evans. (2022). Rethinking Business Information as Veritable Catalyst for Organizational Competitive Advantage. Information Impact: Journal of Information and Knowledge Management. 12. 137-152. 10.4314/ijikm.v12i2.11.
15. Patrisia, Dina & Linda, Muthia & Abror, Abror. (2022). Creation of competitive advantage in improving the business performance of banking companies. Jurnal Siasat Bisnis. 26. 121-137. 10.20885/jsb.vol26.iss2.art1.
16. Ikhwana, Andri & Dianti, Sasi. (2022). The Influence of Information Technology and SCM on Competitive Advantage to Improve MSMEs Performance. International Journal of Computer and Information System (IJCIS). 3. 6-11. 10.29040/ijcis.v3i1.54.
17. Luo, Dean & Shahbaz, Muhammad & Qureshi, Muhammad & Anis, Maira & Mahboob, Farhan & Kazouz, Hayfa & Mao, Jie. (2022). How maritime logistic SMEs lead and gain competitive advantage by applying information technology?. Frontiers in Psychology. 13. 10.3389/fpsyg.2022.955145.
18. Muazu, Umar & Abdulmalik, Sambo. (2021). Information technolugu capabilities and competitive advantage: a review. International Journal of Technology and Systems. 6. 1. 10.47604/ijts.1206.
19. Драб Ю., Ящук В. Основні підходи до побудови системи управління інформаційною безпекою / Інформаційна безпека та інформаційні технології: збірник тез доповідей V Всеукраїнської науково-практичної конференції молодих учених, студентів і курсантів, м. Львів, 26 листопада 2021 року. Львів, ЛДУ БЖД, 2021. С. 29-32.

References

1. Samoilenko L. B. Mozhlivosti ta problemy zastosuvannia tekhnolohii big data vitchyznianymy kompaniiamy. Efektyvna ekonomika. 2018. № 1. – URL: <http://www.economy.nayka.com.ua/?op=1&z=6066> (data zvernennia: 21.11.2022).
2. Informatsiini tekhnolohii v biznesi. Chastyna 1: Navch. posib. / [Shevchuk I.B., Starukh A.I., Vaskiv O.M. ta in.]; za zah. red. I.B. Shevchuk. Lviv: Vydavnytstvo NNVK «ATB», 2020. 455 s.

3. Pidhorneyi M. V., Lukianchenko O. Yu. Doslidzhennia informatyzatsii lohistychnykh protsesiv yak kompleksu zavdan prohramuvannia zhyttievoho tsykladu avtotransportnoho pidpriemstva / Vcheni zapysky TNU imeni V.I. Vernadskoho. Seriya: Tekhnichni nauky, 2021. № 32(71)-2. S. 200-205. DOI: 10.32838/2663-5941/2021.2-2/31
4. Pidhorneyi M.V. Veretiuk S.M. Systemnyi pidkhid do upravlinnia zhyttievym tsyklom informatsiino-komunikatsiinoi tekhnologii. Systemy upravlinnia, navihatsii ta zviazku. Poltava, 2021. Vyp. 1(63) S. 100–103.
5. Buslov P.V. Modeli ta metody dyferentsiatsii danykh konsolidovanoi informatsii dlia system pidtrymky rishen. – Kvalifikatsiina naukova pratsia na pravakh rukopysu. Dysertatsiia na zdobuttia naukovoho stupenia kandydata tekhnichnykh nauk za spetsialnistiu 05.13.06 – informatsiini tekhnologii. –Cherkaskyi derzhavnyi tekhnolohichnyi universytet, Cherkasy, 2021. 184 s.
6. Bokovets V.V., Davydiuk L.P. Elektronna torhivlia yii znachennia dlia rozvytku biznesu. Visnyk Khmelnytskoho natsionalnoho un-tu. Seriya: Ekonomichni nauky. Khmelnytskyi: Khmelnytskyi natsionalnyi universytet. 2021. №1. S.210-214.
7. Kovtun E. O. Elektronna torhivlia v umovakh mizhnarodnoi ekonomichnoi diialnosti / Vinnytskyi torhovelno-ekonomichnyi instytut DTEU, 2021. № 2(298). S. 228-229. DOI: 10.31891/2307-5740-2021-298-5(2)-38
8. Kotsko T.A. Rozvytok pidpriemstva na zasadakh kontseptsii reversyvnoi lohistyky: systema motyviv ta obmezhen / Biznes, innovatsii, menedzhment: problemy ta perspektyvy: zb. tez dop. II Mizhnar. nauk.-prakt. konf., 22 kvit. 2021 r. – Kyiv: KPI im. Ihoria Sikorskoho, Vyd-vo «Politekhnik», 2021. – S. 264-265.
9. Panchenko O. D. Big Data yak osnova informatsiinoho zabezpechennia realizatsii analitychnoi funktsii upravlynskoho obliku v roslynnytsvi. Biznes-navihator. 2020. № 2 (58). S. 115–119.
10. Lishchynska L.B., Dobrovolska N.V. Perspektivni prohramni instrumenty dlia analizu danykh u biznesi / Visnyk Khmelnytskoho natsionalnoho universytetu. 2022. №1. S.78-83.
11. Ponomarenko I. V., Teleus A. V. Biznes-analytika yak efektyvnyi instrument obrobky danykh. Problemy innovatsiino-investytsiinoho rozvytku. 2020. № 23. S. 64-70.
12. Lishchynska L.B. Vprovadzhennia Business Intelligence dlia efektyvnoho upravlinnia pidprymstvom. L Naukovo-tekhnichna konferentsiia pidrozdiliv Vinnytskoho natsionalnoho tekhnichnoho universytetu (2021): zbirnyk dopovidei (Vinnytsia, 10-12 bereznia 2021). Vinnytsia: VNTU, 2021. URL: <https://conferences.vntu.edu.ua/index.php/all-fitki/all-fitki-2021/paper/view/12330/10310>.
13. Shen, Ching-Cheng & Yeh, Chien-Chi & Lin, Chun-Nan. (2022). Using the perspective of business information technology technicians to explore how information technology affects business competitive advantage. *Technological Forecasting and Social Change*. 184. 121973. 10.1016/j.techfore.2022.121973.
14. Otuza, Evans. (2022). Rethinking Business Information as Veritable Catalyst for Organizational Competitive Advantage. *Information Impact: Journal of Information and Knowledge Management*. 12. 137-152. 10.4314/ijjkm.v12i2.11.
15. Patrisia, Dina & Linda, Muthia & Abror, Abror. (2022). Creation of competitive advantage in improving the business performance of banking companies. *Jurnal Siasat Bisnis*. 26. 121-137. 10.20885/jsb.vol26.iss2.art1.
16. Ikhwana, Andri & Dianti, Sasi. (2022). The Influence of Information Technology and SCM on Competitive Advantage to Improve MSMEs Performance. *International Journal of Computer and Information System (IJCIS)*. 3. 6-11. 10.29040/ijcis.v3i1.54.
17. Luo, Dean & Shahbaz, Muhammad & Qureshi, Muhammad & Anis, Maira & Mahboob, Farhan & Kazouz, Hayfa & Mao, Jie. (2022). How maritime logistic SMEs lead and gain competitive advantage by applying information technology?. *Frontiers in Psychology*. 13. 10.3389/fpsyg.2022.955145.
18. Muazu, Umar & Abdulmalik, Sambo. (2021). Information technology capabilities and competitive advantage: a review. *International Journal of Technology and Systems*. 6. 1. 10.47604/ijts.1206.
19. Drab Yu., Yashchuk V. Osnovni pidkhody do pobudovy systemy upravlinnia informatsiinoiu bezpekoiu / Informatsiina bezpeka ta informatsiini tekhnologii: zbirnyk tez dopovidei V Vseukrainskoi naukovo-praktychnoi konferentsii molodykh uchenykh, studentiv i kursantiv, m. Lviv, 26 lystopada 2021 roku. Lviv, LDU BZhD, 2021. S. 29-32.

DOI: <https://doi.org/10.36910/6775-2524-0560-2022-49-09>

УДК 004.41

Поплавська Ганна Володимирівна¹, викладач

<https://orcid.org/0000-0002-1185-8868>

Сачук Юрій Володимирович², к.т.н., доцент

<https://orcid.org/0000-0002-1317-1103>

¹ Комунальний заклад вищої освіти «Луцький педагогічний коледж» Волинської обласної ради, м. Луцьк, Україна

² Луцький національний технічний університет, м. Луцьк, Україна

МЕТОДИЧНІ АСПЕКТИ НАВЧАННЯ СТУДЕНТІВ РОЗРОБКИ ДОДАТКУ ДЛЯ ПЕРЕВІРКИ ЗНАТЬ УЧНІВ ІЗ ВИКОРИСТАННЯМ LASARUS

Поплавська Г. В., Сачук Ю.В. **Методичні аспекти навчання студентів розробки додатку для перевірки знань учнів із використанням Lazarus.** У статті розкрито особливості створення додатку для перевірки знань учнів із використанням мови програмування Lazarus. На прикладі створення тестового додатку, розробленого у середовищі програмування Lazarus, розглянуто основні етапи проектування, об'єкти та алгоритмічні конструкції середовища.

Ключові слова: Lazarus, тестування, проєкт, тест, перевірка знань, програмування.

Poplavska G., Sachuk Yu. Methodical aspects of teaching students to develop an application for testing students' knowledge using Lazarus. The article describes the features of creating an application for testing students' knowledge using the Lazarus programming language. On the example of creating a test application developed in the Lazarus programming environment, the main stages of design, objects and algorithmic designs of the environment are considered.

Keywords: Lazarus, testing, project, test, knowledge test, programming.

Постановка проблеми. Перевірка знань, вмінь та навичок учнів у засвоєнні тем з певних предметів, особливо початкової школи, стала вагомим складовим навчання. Результат навчання великою мірою залежить від методів контролю знань. Адже, контроль знань – це засіб управління навчальною діяльністю учнів. Для того, щоб з функцією перевірки реалізувалася і функція навчання, необхідно створити певні умови та використовувати певні засоби об'єктивної перевірки знань. Для підвищення ефективності навчального процесу, засоби перевірки знань, умінь і навичок учнів початкової школи мають зацікавити здобувачів освіти своїм інтерфейсом, правильно та коректно сформульованими запитаннями.

Аналіз досліджень. Питання застосування тестування учнів вивчається уже багато років. Вагомим дослідженням в цій галузі внесли вчені В. В. Божкова, Мединська С. І., Савчин М. В. Питанням дослідження використання ІКТ- технологій тестування серед учнів займалися Н. Дичківська, О. Сікорський, Н. Щетина, Ляшенко О. І., Лукіна Т. О. [9]

Формулювання цілей статті. Мета статті – розкрити особливості розробки додатку для перевірки знань учнів із використанням Lazarus.

Виклад основного матеріалу. Важливою складовою ланкою навчання є перевірка знань, вмінь та навичок учнів, на основі якої можна виявити переваги та недоліки знань учнів, що в свою чергу призводить до оптимального підбору форм та методів для подання матеріалу вчителем-предметником. Сьогоднішнє стрімке реформування в освіті посприяло використанню ширшим колом вчителів засобів тестування, як ефективного засобу визначення рівня навчальних досягнень учнів. Цей засіб уже багато років зарекомендував себе, як ефективний в теоретичному та практичному плані. На сучасному етапі розвитку освіти тестування набуває визначального характеру в процесі з'ясування компетентності учнів.

Дискусії фахівців з приводу доцільності чи недоцільності введення тестування у навчальні програми, щоб виявити якість освіти, ведуться й досі. Однак, зовнішнє незалежне оцінювання у тестовій формі - це наше сьогоднішнє і, найвірогідніше, майбутнє, тому варто робити «ставку» саме на цей засіб вимірювання рівня компетентності здобувача освіти.

Приступаючи до розкриття мети статті, доцільно спочатку коротко охарактеризувати Lazarus як мову програмування. Lazarus – мова візуального програмування, це візуальний компілятор, який працює у середовищі Windows на базі мови Object Pascal.

Lazarus відноситься до систем візуального програмування, які називають також системами швидкої розробки додатків [3]. Побудова додатків в Lazarus передбачає два взаємопов'язані кроки:

- побудова інтерфейсу програми;

- програмне проектування.

Стадії й етапи розробки тестової програми в середовищі Lazarus:

Перший етап - Аналіз задачі, визначення засобів та методів її вирішення .

Другий етап - Проектування загального алгоритму роботи програми.

Третій етап - Розробка функціональних алгоритмів роботи програми, побудова інтерфейсу .

Четвертий етап - Робробка програмного забезпечення .

П'ятий етап - Тестування програми .

Шостий етап - Завершення та збереження проєкту.

Побудований інтерфейс користувача встановлює взаємодію між користувачем та додатком. Зовнішній вигляд проєкту будується методом розміщення на формі компонентів, які називають об'єктами. Щоб відтворити дію роботи запланованого додатку, що розробляється, використовують вікно модуля(коду). При побудові інтерфейсу програми використовують вікно форми, вікно інспектора об'єктів та головне вікно (рис. 1).

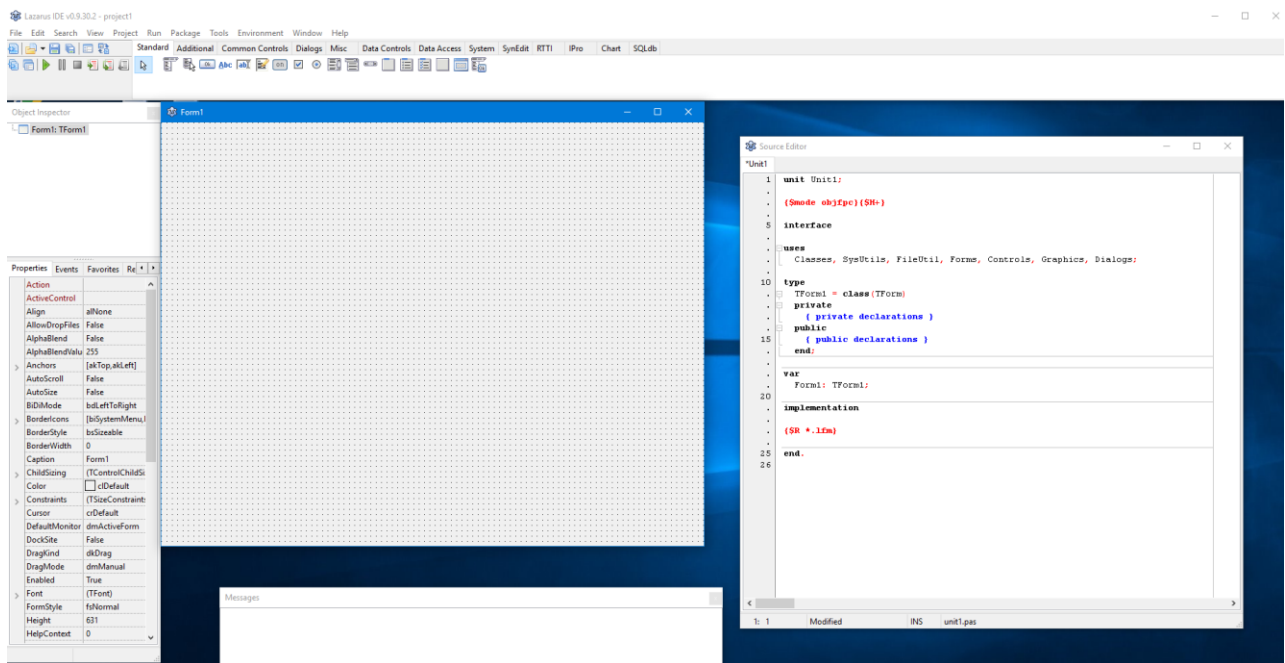


Рис. 1. Інтерфейс візуального середовища Lazarus

Вікна форми та коду тісно пов'язані між собою. Коли ми робимо якісь зміни із формою (базовим об'єктом будь-якого проєкту), то це призводить до автоматичної зміни коду програми. Тому спочатку треба сконструювати форму за тими запитами, які стоять перед розробником, розміщуючи на ній усі компоненти додатку, а потім переходити до написання коду програми, в яких поетапно відображати дії компонентів.

Коли конструюємо проєкти, то розміщуємо на формі компоненти, які підходять для відображення даних проєкту. Усі компоненти у Lazarus є об'єктами. Кожен об'єкт може виконувати певну дію, реагуючи на певну подію. Подіями є будь-які дії користувача та операційної системи щодо кожного компонента окремо.

У Lazarus компоненти – це об'єкти, що реагують на події. Компонент, розміщений на формі, як і будь-який інший об'єкт, відноситься до певного класу. Усі компоненти мають властивість Name – назву компонента. Lazarus дає кожному компоненту загальне ім'я Назва_компонента_N, де N змінюється, починаючи з 1. Найкраще змінювати ім'я, щоб воно відображало призначення даного компонента у програмі. Усі візуальні об'єкти мають різні властивості, що розташовані у Інспекторі об'єктів. Властивості можуть бути як прості, так і складені. Значення кожної властивості можна обирати із списку, що пропонується, або задавати власноруч змінюючи у вікні Інспектора об'єктів. Також можна надавати значення властивостей програмно в програмно в редакторі коду. [5]

Форма є полотном для побудови додатків, нагадує звичайне вікно Windows. На ній розташовані стандартні елементи вікна: кнопки меню, розкриття, згортання, закриття та рядок

заголовка. На формі за допомогою миші розташовуються компоненти. Форма є об'єктом класу TForm.

Форми, що відображають різноманітні повідомлення та запитують від користувача введення будь-якої інформації, називають діалоговими вікнами. Деякі форми дають можливість здійснити двосторонню передачу інформації, але головним завданням діалогових вікон є передача інформації від користувача до програми. Діалоги можна відкрити тільки в модальному режимі та на час передачі інформації.

Коли показана модальна форма, всі повідомлення, що надходять обробляються модальною формою. Вибір способу створення діалогу залежить від завдання, що розв'язується.

Головні компоненти Lazarus, які слід використати при розробці тестової програми, знаходяться на вкладках Standard та CommonControls знаходяться стандартні та додаткові компоненти палітри, а саме: Main Menu – головне меню; Button – стандартна кнопка; Label – мітка; Edit – однорядковий редактор; CheckBox -прапорець; RadioButton – залежний перемикач; RadioGroup – група залежних перемикачів; PageControl – панель.

При розробці програм часто виникає завдання об'єднати або згрупувати різні елементи управління. Об'єднання елементів виконується за допомогою компонентів –контейнерів. Контейнер є візуальним компонентом, на поверхні якого можна розміщувати інші компоненти. Форма програми є основним контейнером, з якого зазвичай і починається проектування інтерфейсу. Форма є власник всіх розміщених на ній компонентів.

При створенні графічних зображень використовуються графічний компонент Image - графічне зображення.

Компонент Image використовується для відображення зображення.

Основна властивість – Picture, що визначає зображення, яке озміщується всередині компонента зображення. Зображення з графічного файлу можна завантажувати на етапі проектування програми та при його виконанні. Потрібно враховувати, що зображення, підключене під час проектування, значно збільшує обсяг файлу програми, що виконується. Щоб уникнути цього, рекомендується завантажувати великі зображення динамічно.

Розглянемо основні можливості Lazarus як засобу розробки контролюючої програми на прикладі лабораторного практикуму з «Основ програмування та технології створення програмних продуктів».

ПРАКТИЧНЕ ЗАНЯТТЯ

Модуль VI. Програмування у середовищі розробки Lazarus.

Тема Створення контролюючої програми

Мета: формування простих навичок створення проектів з використанням декількох форм

Завдання:

- *студенти повинні знати:* інтерфейс програми, призначення, властивості, події для форм; поняття модальної та немодальної форм;
- *студенти повинні вміти:* створювати проекти з використанням декількох форм, застосовуючи при цьому окремі шаблони форм

Хід виконання:

Завдання. Створити програму-тест для учнів початкової школи, яка містить 6 запитань:

1. Скільки пір року ви знаєте? (Для введення відповіді використовуйте компонент Edit)
2. Які місяці є зимовими?
 - Січень, лютий, грудень
 - Березень, квітень, травень
 - Червень, липень, серпень
 - Вересень, жовтень, листопад
3. У якому місяці приходить Святий Миколай?
 - Грудень
 - Січень
 - Лютий
 - Березень
4. В яку пору року з'являються проліски?
 - Зима
 - Весна

- Літо
 - Осінь
5. Скільки днів у високосному році? (Для введення відповіді використайте компонент Edit)
6. Який місяць має 29 днів?
- Березень
 - Жовтень
 - Липень
 - Лютий

Алгоритм розробки проекту

1. Створити папку D:\Прізвище\Pract_test.
2. Змінюємо назву і колір форми і вставляємо компонент *CommonControls* → *PageControl*. Розтягуємо рамку цього компонента і з контекстного меню обираємо *Add Page* (Додати сторінку) (стільки разів, скільки запитань у тестах).
3. Змінимо назви вкладок: 1 питання, 2 питання, 3 питання...
4. Переходимо на вкладку *1 питання*. Розміщуємо на цій вкладці для написання запитання елемент *Standard* → *Label*, для введення відповіді *Standard* → *Edit*. Розміщуємо ще на закладці кнопку *Standard* → *Button1*. Задаємо заголовок кнопки *Відповісти на запитання*. Аналогічно робимо із 5 питанням.
5. Переходимо на вкладку *2 питання*. Розміщуємо на цій вкладці для написання запитання елемент *Standard* → *Label*, для варіантів відповідей *Standard* → *RadioGroup*. Справа від *RadioGroup* вставляємо зображення згідно теми запитання.
6. Аналогічно заповнюємо інші вкладки з запитаннями: кожна вкладка повинна містити компоненти *Label*, *Standard*, *RadioGroup*.
7. Створюємо головне меню *MainMenu* тесту (рис. 2).

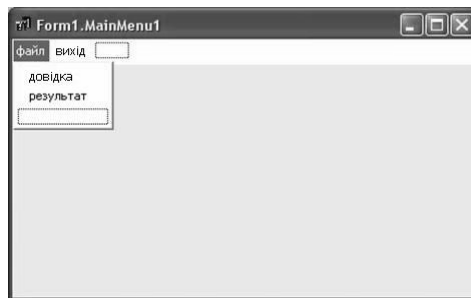


Рис. 2 Створення меню

8. Дизайн практично завершено. Зберігаємо та компілюємо проект (рис. 3).

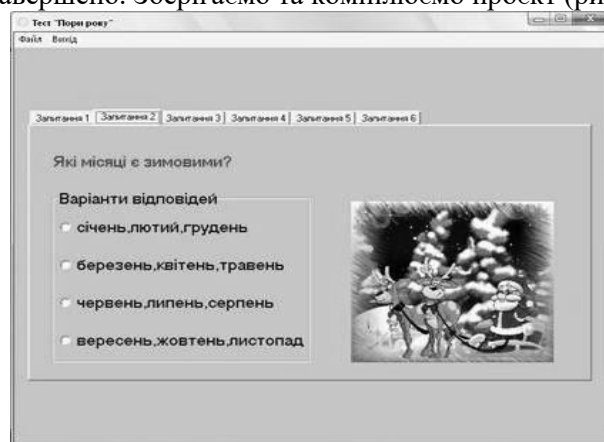


Рис. 3 Дизайн проекту

9. Для кнопки *Вихід* головного меню програмуємо процедуру *close* для виходу з програми.
10. Створюємо форму для довідки *NewForm*. Змінюємо її назву на *Довідка*. На формі надрукуємо інструкцію через компонент *Label*, також вставляємо картинку (рис. 4).

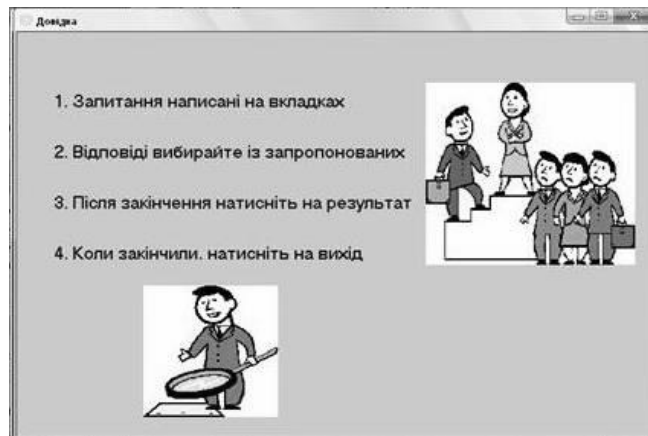


Рис. 4 Форма довідки

11. Створимо процедуру, яка при натисканні на пункт *Допомога* головного меню буде відкривати вікно з довідкою: *Form2.Showmodal*.
12. Для того, щоб використовувати при роботі програми другу форму, в *Unit1* добавляємо *uses Unit2*.
13. Підрахунок балів має здійснюватись при натисканні на пункт *Результат*. При цьому потрібно передбачити необхідність вибору хоча б одного варіанту з запропонованих, наприклад, виведення повідомлення:

```
if RadioGroup1.ItemIndex=-1  
then ShowMessage ('Потрібно вибрати відповідь');
```

Для 1 питання зробити процедуру для кнопки *OnClick*, в якій прописати фрагмент:

```
If StrToInt(Edit1.text)=4 then ball:=ball+1;
```

Аналогічно 5 запитання.

14. Передбачити виведення повідомлення з оцінкою за 12-бальною шкалою оцінювання.
Фрагмент лістингу програми:

```
interface  
uses  
Unit2, Classes, SysUtils, FileUtil, Forms, Controls, Graphics, Dialogs, ComCtrls, StdCtrls;  
type  
{ TForm1 }  
TForm1 = class(TForm)  
Button1: TButton;  
PageControl1: TPageControl;  
TabSheet1: TTabSheet;  
procedure Button1Click(Sender: TObject);  
private  
{ private declarations }  
public  
{ public declarations }  
end;  
var  
Form1: TForm1;  
Ball: integer;  
  
implementation  
{ $R *.lfm }  
{ TForm1 }  
  
procedure TForm1.N4Click(Sender: TObject);  
begin
```

```

close;
end;
procedure TForm1.N2Click(Sender: TObject);
begin
  Form2.ShowModal;
end;
procedure TForm1.N3Click(Sender: TObject);
begin
  Ball:=0;
  if RadioGroup2.ItemIndex=-1 then ShowMessage ('Потрібно вибрати відповідь') else
  if RadioGroup2.ItemIndex=0 then Ball:=Ball+1;
  if RadioGroup3.ItemIndex=-1 then ShowMessage ('Потрібно вибрати відповідь') else
  if RadioGroup3.ItemIndex=1 then Ball:=Ball+1;
  if RadioGroup4.ItemIndex=-1 then ShowMessage ('Потрібно вибрати відповідь') else
  if RadioGroup4.ItemIndex=3 then Ball:=Ball+1;
  if RadioGroup6.ItemIndex=-1 then ShowMessage ('Потрібно вибрати відповідь') else
  if RadioGroup6.ItemIndex=1 then Ball:=Ball+1;
  ShowMessage ('Ваш результат'+IntToStr(Ball));
end;
end.

```

Висновки та перспективи подальшого дослідження. У роботі розкрито особливості розробки додатків для перевірки знань учнів із використанням мови програмування Lazarus. Розроблені додатки у середовищі програмування Lazarus мають як перспективи, так і недоліки. Такі додатки можуть бути використані як приклад показу можливості створювати контрольні програми для учнів школи будь-якого віку та з будь-якого предмета. Недоліком таких додатків є те, що вони «одноразові», тобто створивши будь-який тест, не можна його корегувати, удосконалювати, додавати чи змінювати запитання чи відповіді. Але з іншого боку, зробивши один, два чи більше тестів, запам'ятавши програмний код та навчившись створювати форму, на створення такого тесту витрачається 15-30 хвилин.

Список бібліографічного опису

1. Алексеев С. Р. Самовчитель з програмування на Free Pascal та Lazarus. Донецьк : ДонНТУ, Технопарк ДонНТУ УНІТЕХ, 2011. 503 с.
2. Бублик В.В. Об'єктно-орієнтоване програмування. К.: ІТкнига, 2015. 624 с.
3. Гуриков С. Р. Програмування в середовищі Lazarus для школярів та студентів : навч. посібник. К. : Форум : НДЦ ІНФР, 2016. 336 с.
4. Белов В. В. Програмування в Delphi : процедурне, об'єктивно орієнтоване, візуальне: Навч. посібник для вузів. К.: Гаряча лінія, Телеком, 2014. 240 с.
5. Мансуров К. Т. Основи програмування у середовищі Lazarus. К.: Palmarium Academic Publishing, 2013. 772 с.
6. Завадський І. О., Заболотний Р. І. Основи візуального програмування. К.: Вид. група ВHV. 2007. 272 с.
7. Семакін І. Н. Основи програмування : підручник. К.: Вища школа, 2002. 432 с.
8. Информатика : підруч. для 9-го кл. загальноосвіт. Навч. закл. Київ. : Генеза, 2017. 288 с.
9. URL: <https://naurok.com.ua/stattya-vikoristannya-suchasnih-testovih-tehnologiy-dlya-perevirki-znan-uchniv-naurokah-273884.html> (дата звернення: 10.12. 2022)

References

1. Alekseev E.R. Free Pascal and Lazarus programming tutorial. Donetsk: DonNTU, Technopark of DonNTU UNITECH, 2011. 503 p.
2. Bubyk V.V. Object-oriented programming. K.: IT book, 2015. 624 p.
3. Gurikov S. R. Programming in the Lazarus environment for schoolchildren and students: teaching. manual. K.: Forum: NDC INFR, 2016. 336 p.
4. Belov V.V. Programming in Delphi: procedural, objective-oriented, visual: Education. a guide for universities. K.: Hotline, Telecom, 2014. 240 p.
5. Mansurov K. T. Fundamentals of programming in the Lazarus environment. K.: Palmarium Academic Publishing, 2013. 772 p.
6. Zavadskiy I.O., Zabolotnyi R.I. Fundamentals of visual programming K.: Ed. BHV group. 2007. 272 p.
7. Semakin I. N. Fundamentals of programming: a textbook. K.: Higher School, 2002. 432 p.
8. Informatics: tutor. for the 9th grade general education Education closing Kyiv. : Geneza, 2017. 288 p.
9. URL: <https://naurok.com.ua/stattya-vikoristannya-suchasnih-testovih-tehnologiy-dlya-perevirki-znan-uchniv-naurokah-273884.html> (date of application: 10.12. 2022)

DOI: <https://doi.org/10.36910/6775-2524-0560-2022-49-10>

УДК 681.518:62-50

Свиридюк Катерина Аатоліївна, магістр

<https://orcid.org/0000-0003-0138-3931>

Хрестинець Максим Олександрович, магістр

Черняшук Наталія Леонідівна, д.пед.н., професор

<https://orcid.org/0000-0002-3178-8377>

Хрестинець Наталія Анатоліївна, к.т.н., ст.викл.

<https://orcid.org/0000-0002-4836-7632>

Луцький національний технічний університет, м. Луцьк, Україна

ВИЗНАЧЕННЯ ВПЛИВУ ЕРГОНОМІКИ ТА ПОШУКОВОЇ ОПТИМІЗАЦІЇ НА ФАКТОРИ РАНЖУВАННЯ САЙТУ

Свиридюк К.А., Хрестинець М.О., Черняшук Н.Л., Хрестинець Н.А. **Визначення впливу ергономіки та пошукової оптимізації на фактори ранжування сайту.** Статтю присвячено аналізу пошукової оптимізації сайту з позицій ергономіки, основ юзабіліті, основ інтернет-маркетингу та ключових факторів ранжування.

Поставлена задача визначення впливу UX-дизайну на релевантність сайту реалізується за допомогою евристик Нільсена, а також за допомогою французького типу юзабіліті і набору евристик Бастьєна і Скапіна. Розглянуто позитивні та негативні сторони адаптивної верстки, обґрунтовано теоретично ріст коефіцієнту конверсій.

Евристичний підхід ґрунтується на формуванні записів, категорій та сторінок сайту, що побудований на CMS WordPress, та на побудові класичного семантичного ядра. Досліджено теоретично, що фактори ранжування сайту складають сигнали, які пошукові системи використовують для сортування результатів пошуку відповідно до їх якості та відповідності конкретному запиту. Кожен сигнал може або не може використовуватися Google для цілей рейтингу або може бути трохи меншим чи більш важливим залежно від пошукового запиту чи результатів пошуку.

Ключові слова: ергономіка, пошукова оптимізація, SEO, ранжування, евристики Нільсена, адаптивна верстка.

Svyrydiuk K.A., Khrystynets M.O., Chernyashchuk N.L., Khrystynets N.A. **Determining the impact of ergonomics and search engine optimization on site ranking factors.** The article is devoted to the analysis of search engine optimization of the site from the standpoint of ergonomics, the basics of usability, the basics of Internet marketing and key ranking factors. The task of determining the impact of UX design on the relevance of the site is implemented using Nielsen's heuristics, as well as using the French type of usability and a set of heuristics by Bastien and Scapin. The positive and negative sides of the adaptive layout are considered, the growth of the conversion rate is theoretically substantiated. The heuristic approach is based on the formation of records, categories and pages of a site built on CMS WordPress and on the construction of a classic semantic core. It has been theoretically studied that site ranking factors are signals that search engines use to sort search results according to their quality and relevance to a specific query. Each signal may or may not be used by Google for ranking purposes or may be slightly less or more important depending on the search query or search results.

Keywords: ergonomics, search optimization, SEO, ranking, Nielsen heuristics, adaptive layout.

Постановка проблеми. Оптимізація є важливою та фундаментальною проблемою для вирішення питань покращення результату в різних наукових дисциплінах. Взаємодія з багатьма людьми з усіх верств населення життя через формати інтернет-даних можна визначити як цифрове медіа в дуже простому вигляді, де інформація зберігається в електронний формат, такий як текст, зображення, аудіо або відео-контент. Унікальність цифрових медіа полягає в тому, що є можливість легко надсилати та копіювати дані через цифровий код. Щоб вижити в конкурентній боротьбі, популярною і актуальною є стратегія SEO – оптимізація веб-додатку для покращення видимості веб-сайту в пошуковій системі Google і зацікавлення відвідувачів, або користувачів веб-сайту. Метою роботи є реалізація пошукової оптимізації веб-додатку на платформі WordPress.

Для досягнення мети необхідно провести аналіз останніх досліджень і публікацій з питань просування сайтів, дослідити параметри впливу сервіс-аналітиків, експертних оцінок, вебметричних даних, семантичного ядра на оцінку індексування веб-додатку та технічні характеристики хостингу і забезпечити повноцінне функціонування файлового клієнту та бази даних, а також реалізувати внутрішню та зовнішню оптимізацію веб-додатку і дослідити індексацію сайту та швидкість його ранжування.

Аналіз досліджень. Пошукова оптимізація, або SEO, дозволяє зрозуміти, як споживачі шукають і знаходять інформацію про бренд і його конкурентів в Інтернеті. Використовуючи стратегії SEO у маркетингових дослідженнях, можна підвищити видимість і рейтинг веб-сайту. Наведені статистичні дані (рисунок 1) допоможуть точно налаштувати вашу стратегію SEO, збільшити відвідуваність веб-сайту та підвищити кількість конверсій.

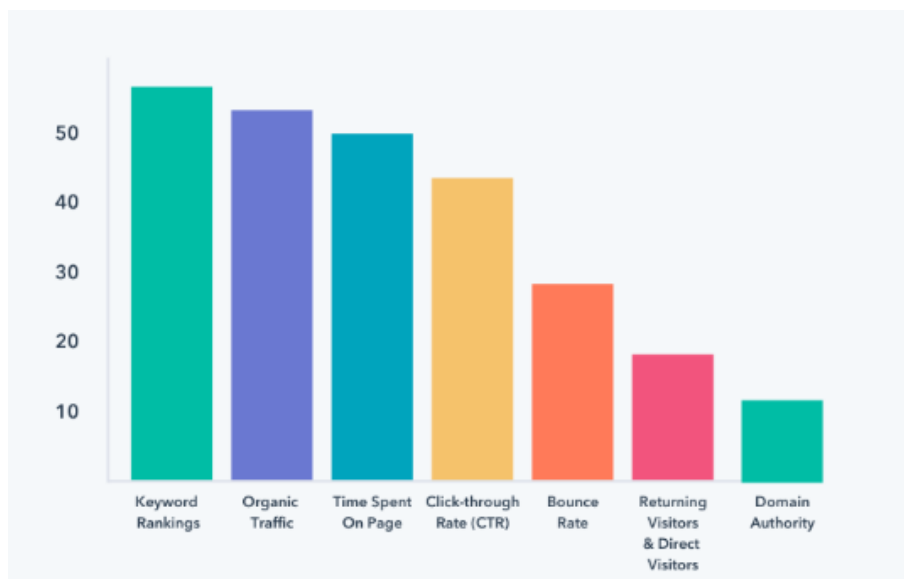


Рис. 1 – Вплив факторів оптимізації на стан світового маркетингу

Огляд сучасних публікацій щодо напрямку SEO показав наступне. Автори [1,2] підтверджують черговість впливу факторів оптимізації, що наведені у п.1.1 кваліфікаційної роботи та зазначають, що «оптимізацію вмісту для SEO можна здійснити кількома способами. Виявлення правильних ключових слів/ключових фраз і рівномірне вплетення їх у контент є одним із найефективніших способів оптимізації контенту. Отже, якщо визначено 5-10 ключових слів/ключових фраз, які важливі для інформації, а також є найбільш використовуваними підказками в пошуковій системі; переконайтеся, що ці ключові слова / ключові фрази використовуються більше одного разу у своєму вмісті, і вони рівномірно розподілені по вмісту».

Відомий інтернет-ресурс Searchenginejournal [3] опублікував дані аналітики про те, що вплив машинного навчання та штучного інтелекту значною мірою впливає на успішну пошукову оптимізацію. Згідно з дослідженнями, практики SEO агентства передбачають очікування того, що машинне навчання (Machine learning - ML) і штучний інтелект (Artificial intelligence - AI) спричинять найбільші зміни в SEO. Екс-працівник Google Маріса Майер [4] стверджує, що «один із способів, як Google намагався боротися із загальним зниженням якості, це доповнення свого індексу трильйона веб-сторінок деяким власним вмістом». Якщо поставити просте запитання про кулінарію чи вік якогось політика чи актора, або навіть про те, який підкаст найкращий, можна побачити те, що Майер називає «вбудованим результатом» або те, що Google називає «виданим фрагментом» – фрагмент тексту, який відповідає на запитання прямо на сторінці результатів пошуку, без необхідності натискати посилання».

Виклад основного матеріалу й обґрунтування отриманих результатів. Ергономіка веб-сайту в першу чергу залежить від ясності, ієрархії, простоти та узгодженості, необхідних для оптимізації взаємодії з користувачем. Необхідно дотримуватись цих способів з основ ергономіки, щоб отримати вищі позиції в результатах пошукової системи, а також покращувати активність користувачів на сайті, щоб вони залишалися на ньому та переходили до дій.

Важливим розділом/частиною ергономіки є usability (зручність використання) сайту. Оскільки багато користувачів формують думку про бренд виключно на основі досвіду роботи з веб-сайтом компанії, цінність UX (usability extended) стає очевидною, бо саме зручність використання веб-сайту є надважливою.

Якщо веб-сайт можна використовувати, це означає, що користувачі можуть досягати своїх цілей: шукати товари, сторінки, теми на блогах тощо. Дизайн і розробка веб-сайту зосереджені на взаємодії з користувачем, щоб користувачам було легко виконувати бажані дії. Ефективність UX електронної комерції залежить від сотень елементів, які впливають на те, наскільки легко знайти та вибрати продукти, знайти політику доставки та повернення, налаштувати обліковий запис і завершити покупку.

Вагомість використання UX залежить від характеру сайту та мети веб-додатку, яка стоїть за ним. Але, загалом, веб-сайти надають добре організовану, корисну інформацію, просту для навігації, у добре структурованому форматі. Якщо розробник дотримується стандартів зручності

використання веб-сайту, щоб зробити сайт зручнішим для користувачів, то коефіцієнт конверсії сайту покращиться.

У світі UX-дизайну існує ряд інструментів, якими повинен користуватися кожен: тестування користувачів, виведення макету сайту, евристичні сітки (або евристики) тощо.

Важливими і обґрунтованими є евристики – списки критеріїв, визначених для аналізу інтерфейсів. Використовуючи визнану евристику, можна покращити існуючі інтерфейси, вказавши на елементи, які не відповідають сітці, і які, отже, є областями для вдосконалення. Також можна створювати інтерфейси, які сприяють оптимальній ергономіці та взаємодії з користувачем, ставлячи під сумнів їхню адекватність евристикам під час створення екранів. Найвідоміша сітка критеріїв для UX-дизайнерів – це сітка Нільсена. Сітка спрямована на покращення взаємодії з користувачем і, безсумнівно, є найбільш використовуваною сьогодні. Але його популярність багато в чому пов'язана з основним способом вивчення UX. У дизайні UX у світі все таки домінує англосаксонське бачення. Це нормально, оскільки UX народився в США в 90-х. Але для досягнення тих самих цілей у світі розвивалися й інші правила та способи, зокрема французький тип юзабіліті. Відома значна кількість евристик, у кваліфікаційній роботі використаємо набір із 24 евристик, які згруповані в шість ергономічних критеріїв, які були створені шляхом інтеграції ергономічних критеріїв, запропонованих Бастьеном і Скапіном з десяти евристик Нільсена і Моліха [5]:

- керівництво користувача;
- робоче навантаження;
- адаптивність і контроль;
- управління помилками;
- послідовність і стандарти;
- сумісність.

Для адаптивної мобільної версії не потрібно оптимізувати ні десктопну, ні мобільну версію сайту. Незалежно від того, який пристрій використовує користувач, запити надходять на той самий сервер, який надсилає користувачеві той самий HTML-код. Вміст ресурсу підлаштовується під роздільну здатність екрану за допомогою CSS. Це дозволяє зробити функціональність і дизайн ресурсу під будь-який пристрій. Перевагами адаптивної мобільної верстки є єдиний URL для всіх версій сайту, відсутність зайвих елементів на сторінці, а також коректне відображення сторінок на всіх пристроях.

У той же час адаптивна верстка має ряд недоліків. Зокрема, необхідність скорочення текстів при оптимізації сайту під мобільні пристрої може привести до того, що якщо на сайті багато відео, великоформатних зображень або анімації, це може вплинути на швидкість завантаження сторінки.

Згідно з дослідженнями відомого веб-розробника та автора багатьох книг та публікацій Тіма Кадлека [6-7], коефіцієнт конверсії майже втричі вищий на комп'ютері, порівняно зі смартфоном. Зважаючи на це, ми можемо сказати, що адаптивний веб-дизайн можна визнати переможцем, якщо ваш бізнес сильно покладається на користувачів комп'ютерів. Однак факт залишається фактом: глобальні продажі ПК падають. Незважаючи на те, що кількість ноутбуків і настільних комп'ютерів залишається високою – у 2018 році очікується понад 260 мільйонів поставок – ринок мобільних пристроїв стрімко зростає та починає переважати в бізнесі. Дослідження показують [8], що незабаром на смартфони припадатиме 80% глобального використання Інтернету. У той же час Google повідомляє, що 61% користувачів навряд чи повернуться на мобільний сайт, на який мали проблеми з доступом. Інші статистичні дані також демонструють ту саму тенденцію: У 2017 році роздрібні продажі електронної комерції сягнули 2,3 трильйона доларів США, що на 23,2% більше, ніж у минулому році. Мобільна електронна комерція може заробити 3,5 трильйона доларів у 2022 році, а потім становитиме майже три чверті всіх продажів електронної комерції. Споживачі, які здійснюють покупки в Інтернеті за допомогою своїх мобільних пристроїв, як правило, витрачають вдвічі більше через цифрові канали, ніж ті, хто не купує на смартфонах.

Знання ключових факторів ранжування є важливим для реалізації сайту в мережі Інтернет. Хоча деякі сигнали рейтингу все ще оповиті таємницею Google, багато з них добре встановлені та підтверджені офіційними особами Google. Фактори ранжування сайту (рисунок 2) складають сигнали, які пошукові системи використовують для сортування результатів пошуку відповідно до

їх якості та відповідності конкретному запиту. Google використовує сотні сигналів ранжування з багатьма змінними. Кожен сигнал може або не може використовуватися Google для цілей рейтингу або може бути трохи меншим чи більш важливим залежно від пошукового запиту чи результатів пошуку. Є думки [9], що Google має електронну таблицю з усіма сигналами ранжування, і вони можуть просто сортувати їх за потребою. Існує ряд досліджень, які намагаються довести (або спростувати) існування деяких факторів ранжування [10], просто налаштувавши або внісши зміни на веб-сайті, які корелювали б зі змінами його рейтингу в Пошуку Google.



Рис. 2 – Фактори ранжування сайту

Веб-сторінки знаходяться краулерами шляхом індексування. Тобто, процес відвідування пошуковими системами та аналізу нових і оновлених веб-сторінок перед додаванням їх до свого індексу (бази даних) веб-сторінок формує індексацію.

Індексацію можна виконати трьома способами: роботою сканерів, за допомогою карти сайту, або за запитом вручну. Системи керування вмістом часто налаштовані таким чином, щоб дозволити індексацію щойно опублікованих сторінок за умовчанням

Висновки та перспективи подальшого дослідження. Аналіз наукових досліджень і публікацій показав, що близько 81 відсоток онлайн-покупців використовують Інтернет для пошуку, перш ніж прийняти рішення про покупку, що свідчить про актуальність веб-розробок. Виявлено, що важливо витратити час і ресурси на дизайн веб-сайту і не довірятимуть бренду та не купуватимуть продукт, якщо вважатимуть сайт заплутаним або марним, тобто, аргументовано створення зручного для користувача usability сайту. Дослідження показують, що незабаром на смартфони припадатиме 80% глобального використання Інтернету, тобто, мобільна адаптація сайту є необхідною для повноцінного функціонування.

Проілюстрована послідовність побудови моделі ранжування сайту може використовуватись фахівцями у галузі: SEO-спеціалістами та девелоперами. Також, рекомендації оптимізації веб-додатків, що запропоновані в роботі, мають навчальну практичну цінність для вивчення дисциплін веб-дизайну, веб-розробки та основ SEO.

Список бібліографічного опису

1. Kanika Sachdeva. Understanding SEO [Electronic resource] // OMEMY Tutorials, October, 2021. – Mode of access: https://www.researchgate.net/publication/356412343_Understanding_SEO
2. Kanika Sachdeva. Grow your Small Business with Digital Marketing [Electronic resource] // OMEMY Tutorials, October, 2021. – Mode of access: https://www.researchgate.net/publication/357091678_Grow_your_Small_Business_with_Digital_Marketing_How_Digital_Marketing_helps_to_Grow_Business
3. Searchenginejournal [Electronic resource] – Mode of access: <https://www.searchenginejournal.com>

4. Marissa mauer at Google [Electronic resource] // Melissa Thomas -- Mode of access: <https://www.emerald.com/insight/content/doi/10.1108/case.darden.2016.000186/full/html>
5. Bastien & Scapin's Criteria, an Alternative to Nielsen's // Published in UX Planet [Electronic resource] // Mode of access: <https://uxplanet.org/bastien-scapins-criteria-an-alternative-to-nielsen-s-5b36e6937b73/>
6. Kadlec, Tim. Implementing Responsive Design: Building sites for an anywhere, everywhere web / Tim Kadlec// New Riders. Voices That Matter – 2012. – 422p.
7. Bendell C., Kadlec T., Weiss Y. and oth. High Performance Images: Shrink, Load, and Deliver Images for Speed / Colin Bendell, Tim Kadlec, Yoav Weiss, Guy Podjarny, Nick Doyle, Mike McCall // Published by O'Reilly Media, Inc – 2017. – 307p.
8. Kadlec, Tim. Praxiswissen Responsive Webdesign / Tim Kadlec // Published by O'Reilly Media, Inc – 2013. – 187p.
9. Сайт The 2020 State of Business Communication [Електронний ресурс]. – Режим доступу : <https://www.nextiva.com/x/businesscommunication-report/>
10. Безкоштовний курс "SEO Outreach Specialist" [Електронний ресурс]. – 2018. – Режим доступу до ресурсу: <https://unistudy.org.ua/seo-outreach-specialist/>

References

1. Kanika Sachdeva. Understanding SEO [Electronic resource] // OMEMY Tutorials, October, 2021. – Mode of access: https://www.researchgate.net/publication/356412343_Understanding_SEO
2. Kanika Sachdeva. Grow your Small Business with Digital Marketing [Electronic resource] // OMEMY Tutorials, October, 2021. – Mode of access: https://www.researchgate.net/publication/357091678_Grow_your_Small_Business_with_Digital_Marketing_How_Digital_Marketing_helps_to_Grow_Business
3. Searchenginejournal [Electronic resource] – Mode of access: <https://www.searchenginejournal.com>
4. Marissa mauer at Google [Electronic resource] // Melissa Thomas -- Mode of access: <https://www.emerald.com/insight/content/doi/10.1108/case.darden.2016.000186/full/html>
5. Bastien & Scapin's Criteria, an Alternative to Nielsen's // Published in UX Planet [Electronic resource] // Mode of access: <https://uxplanet.org/bastien-scapins-criteria-an-alternative-to-nielsen-s-5b36e6937b73/>
6. Kadlec, Tim. Implementing Responsive Design: Building sites for an anywhere, everywhere web / Tim Kadlec// New Riders. Voices That Matter – 2012. – 422p.
7. Bendell C., Kadlec T., Weiss Y. and oth. High Performance Images: Shrink, Load, and Deliver Images for Speed / Colin Bendell, Tim Kadlec, Yoav Weiss, Guy Podjarny, Nick Doyle, Mike McCall // Published by O'Reilly Media, Inc – 2017. – 307p.
8. Kadlec, Tim. Praxiswissen Responsive Webdesign / Tim Kadlec // Published by O'Reilly Media, Inc – 2013. – 187p.
9. Сайт The 2020 State of Business Communication [Електронний ресурс]. – Режим доступу : <https://www.nextiva.com/x/businesscommunication-report/>
10. Безкоштовний курс "SEO Outreach Specialist" [Електронний ресурс]. – 2018. – Режим доступу до ресурсу: <https://unistudy.org.ua/seo-outreach-specialist/>

DOI: <https://doi.org/10.36910/6775-2524-0560-2022-49-11>

УДК 004.04

Фещенко Іван Олександрович, магістрант

<https://orcid.org/0000-0003-2884-4822>

Дадиверін Віталій Валерійович, магістрант

<https://orcid.org/0000-0001-5121-2263>

Потапова Катерина Романівна, к.т.н., доцент

<https://orcid.org/0000-0002-3347-6350>

Наливайчук Микола Васильович, к.т.н., ст. викладач

<https://orcid.org/0000-0002-8942-9844>

Національний технічний університет України «Київський політехнічний інститут імені Ігоря Сікорського»

ТЕХНОЛОГІЯ СУПЕРСЕМПЛІНГУ ДЛЯ ПОКРАЩЕННЯ ПРОДУКТИВНОСТІ РЕНДЕРИНГУ ЗОБРАЖЕННЯ МЕТОДОМ ТРАСУВАННЯ ПРОМЕНІВ У РЕАЛЬНОМУ ЧАСІ

Фещенко І. О., Дадиверін В. В., Потапова К.Р., Наливайчук М.В. Адаптування технології суперсемплінгу для збільшення продуктивності рендерингу зображення методом трасування променів у реальному часі. У статті представлено варіант адаптації технології суперсемплінгу, або надлишкової вибірки згладжування, для збільшення продуктивності рендерингу зображення методом трасування променів у реальному часі. Запропоновано модифікований підхід з різною роздільною здатністю. Замість того, щоб дотримуватися жорсткої структури сітки квадродрев, яка властива текстурам і збереженим пікселям, для збору зразків використовується трикутна сітка. Ця трикутна сітка фактично накладається на традиційний формат текстури, який використовується для збору зразків у офсетному візерунку, а потім повертається до традиційної сітки для відображення на екрані. Ми вважаємо, що використання цієї трикутної вибірки потребуватиме меншої кількості вибірок, покращуючи ефективність алгоритму без погіршення якості кінцевої візуалізації. Представлено результати тестування адаптованого алгоритму відносно немодифікованого методу рендерингу.

Ключові слова: суперсемплінг, адаптивний семплінг, трасування променів, рендеринг, світло, роздільна здатність, графічний прискорювач.

Feshchenko I.O., Dadyverin V.V., Potapova K.R., Nalyvaichuk M.V. Adaptation of supersampling technology to increase the performance of real-time ray tracing image rendering. The article presents an option for adapting the supersampling technology, or oversampling smoothing, to increase the performance of image rendering using real-time ray tracing. A modified approach with different resolutions is proposed. Instead of following the rigid quad tree structure inherent in textures and stored pixels, a triangular grid is used to collect samples. This triangular mesh is actually superimposed on the traditional texture format used to collect the samples in the offset pattern and then returned to the traditional mesh for display on the screen. We believe that using this triangular sample will require fewer samples, improving the efficiency of the algorithm without compromising the quality of the final rendering. The results of testing the adapted algorithm relative to the unmodified rendering method are presented.

Keywords: supersampling, adaptive sampling, ray tracing, rendering, light, resolution, graphics accelerator.

Постановка проблеми.

Головним видом подання інформації користувачу використовуючи електронні обчислювальні пристрої є графічне відображення на дисплеї. Одною з основних цілей сучасної комп'ютерної графіки є генерування зображення, якість якого близька до фотореалістичної. Метод трасування променів є перспективним варіантом розвитку рендерингу, але достатньо ресурсозатратним. Проблеми, які вирішує даний метод порівняно до інших:

- якісне та реалістичне результуюче зображення;
- відсутність апроксимації при рендерингу гладких об'єктів;
- можливість паралелізації обчислень на рівні алгоритму;
- мала залежність складності алгоритму від складності схеми.

Аналіз існуючих рішень.

Існує кілька підходів, які були розроблені для зменшення або усунення проблем із накладанням спектрів у сценах із трасуванням променів. Більшість, однак, створює високу вартість продуктивності через необхідність великої кількості променів на піксель, як у моделях розподіленого трасування променів. Однією з таких областей дослідження є техніка, відома як адаптивна супервибірка. Завдяки цьому процесу пікселі можуть бути ефективно розділені за

потреби залежно від швидкості зміни між пікселями та наступними субпікселями. Потім можна використовувати промені для визначення конкретного кольору субпікселів, результати яких можна об'єднати разом, щоб відтворити справжній колір повного пікселя. Цей зв'язок також може бути спрямований у зворотному напрямку, у процесі, що називається недостатньою дискретизацією, дозволяючи одному променю вміщувати колір кількох пікселів у регіонах із невеликими змінами. Разом ці методи разом називаються адаптивною вибіркою. Один із способів реалізації адаптивної вибірки передбачає використання кількох сіток різної роздільної здатності, починаючи від більш грубих варіантів цільової роздільної здатності до все більш тонких кратних цільової роздільної здатності. Застосування цього підходу до адаптивної вибірки з різною роздільною здатністю дозволяє покращити згладжування зображень, що відтворюються, із покращеним часом відтворення порівняно з візуалізаціями з простою вибіркою та супервибіркою.

Постановка завдання.

Метою роботи є дослідження можливості оптимізації методу трасування променів для відображення графічної сцени задля підвищення продуктивності.

Для досягнення мети вирішуються наступні наукові завдання:

- розглянути загальний алгоритм трасування променів та виконати його опис;
- розглянути алгоритм адаптивної вибірки;
- розглянути метод почергового рендерингу у текстурі з різною роздільною здатністю;
- дослідити можливість використання трикутної сітки для зменшення зразків текстурі і оцінити ефективність даного підходу.

Виклад основного матеріалу дослідження.

Трасування променів – це техніка рендерингу, яка відстежує шлях світла у вигляді пікселів на площині зображення та імітує ефект зіткнення з об'єктами в сцені. Цей метод дає змогу досягти більшої реалістичності, ніж звичайні методи рендерингу, але вона вимагає великих обчислювальних витрат і створює велике навантаження на графічний прискорювач. З цієї причини трасування променів найбільше підходить для відносно тривалого нехтувано малого часу рендерингу, наприклад, для нерухомих комп'ютерних зображень і візуальних ефектів у кіно й телебаченні, але, як правило, не підходить для застосунків реального часу, де важлива швидкість покадрового рендерингу, наприклад, для відеоігор.

Однак, останніми роками апаратне прискорення для трасування променів у реальному часі стало стандартним для комерційних відеокарт, і графічні АРІ наслідували цей приклад, даючи змогу додавати цей метод до програмного забезпечення, такого як ігри, де важлива продуктивність у реальному часі.

Принципи алгоритму трасування променів засновані на фізичних законах поширення світла. Таким чином, кожен об'єкт у сцені має матеріал із такими властивостями:

- Випромінювання енергії (англ. *emittance*) – кількість і довжина хвилі світла, що випромінюється об'єктом.
- Шорсткість (англ. *roughness*) – наскільки сильно розсіюються промені світла під час потрапляння на об'єкт.
- Відбивна здатність (англ. *reflectance*) – кількість і довжина хвилі світла, відбитого об'єктом.
- Прозорість (англ. *transparency*) – відношення кількості світла, що проходить через об'єкт, до кількості відбитого світла.
- Заломлення (англ. *refraction*) – міра заломлення світла.

Узагальнена схема трасування променів представлена на рисунку 1 і може бути описана наступними етапами:

- Кидання променя з кожного пікселя.
- Перевірка чи було перетинання об'єкта променем. Якщо так – див. пункт 3, якщо ні – див. пункт 5.
- Обчислення впливу кольору об'єкта на колір пікселя.
- Якщо кількість ітерацій не перевищена, то обчислити новий напрямок променя та повернутися до пункту 1, інакше закінчити роботу алгоритму.
- Обчислення впливу кольору фоновго зображення на колір пікселя та закінчити роботу алгоритму.

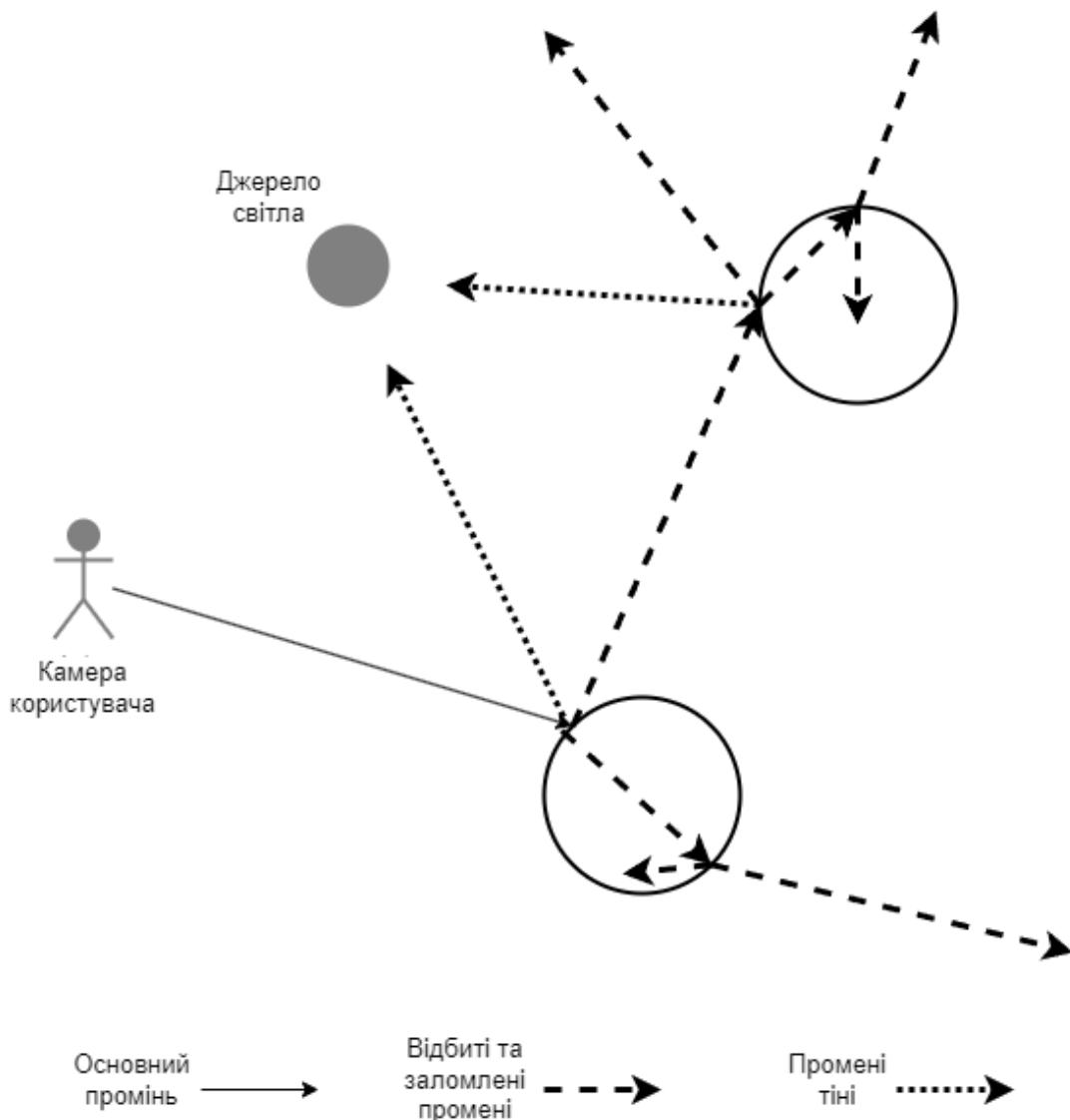


Рис. 1. Схематичне зображення алгоритму рейтрейсингу

Розглянемо метод адаптивної вибірки (англ. adaptive sampling), яку використаємо як один з елементів оптимізації для підвищення продуктивності. Адаптивна вибірка – це алгоритм, що складається з двох процесів, пов'язаних рівнем деталізації необхідним в сцені. На ділянках, де мало деталей і відносно незначні зміни піксель до пікселя, було б достатньо використовувати один промінь для представлення більших блоків пікселів. І навпаки, на ділянках, де спостерігається більша швидкість зміни між пікселями, більше одного променя на піксель може знадобитися для точного визначення правильного кольору пікселя. На рисунку 2 зображено приклад виділення таких ділянок. Хоча геометрія на сцені спрощена, є достатня кількість деталей для перегляду, що також ускладнено шаховим візерунком текстури підлоги. У цій сцені ми можемо виділити всі регіони, де необхідно визначити супервибірку, відповідним зеленим кольором. Примітно, що виділення лежать уздовж країв об'єктів або країв шахової дошки. Це ж зображення можна використовувати, щоб зрозуміти, де можлива недостатня вибірка. Зверніть увагу, що більшість композицій об'єктів, які не відображаються активно інші об'єкти на сцені залишаються еквівалентного або дуже схожого кольору. Зважаючи на це, немає потреби витратитися велику кількість променів, щоб визначити, що один піксель неба справді такого ж кольору, як сусідній піксель. Натомість один промінь можна використовувати для визначення кольору цілої області пікселів на отриманому зображенні.

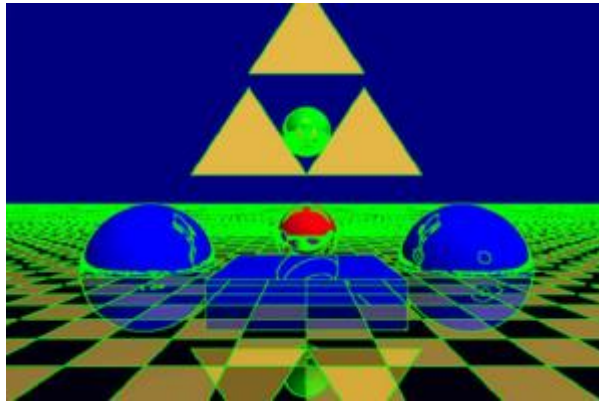


Рис. 2. Виділення зон для суперсемплінгу

Рендеринг у декілька текстур з різною роздільною здатністю

У запропонованому підході використовується підхід із різною роздільною здатністю, щоб визначити, коли кидати більше променів. Основна передумова передбачає спочатку рендеринг сцени на низькій роздільній здатності, що є менше на деяку ступінь двійки за цільову роздільну здатність. Ця роздільна здатність представляє стан із недостатньою дискретизацією візуалізації сцени, де один промінь представляє кілька пікселів візуалізації з цільовою роздільною здатністю. Інформація про сцену збирається з візуалізації з заданою роздільною здатністю, включаючи інформації від того, де і як промінь потрапив на найближчий об'єкт, до кольору отриманого на результуючому зображенні. Ці дані зберігаються в групі окремих текстур, кожна з яких відповідає певній роздільній здатності. Ці текстурні файли потім масштабуються, щоб відповідати наступним сценам з вищою роздільною здатністю. На цьому новому рівні роздільної здатності рендерер тестує текстури попередньої роздільної здатності щоб визначити, чи знімати нові промені з поточною роздільною здатністю. Це виконано шляхом порівняння значень текстур, що відповідають поточному пікселю, зі значеннями сусідніх пікселів. Якщо існує значна різниця між поточним пікселем і сусідніми, то передбачається, що всередині відбувається щось і нові промені мають кидатися у цей локальний блок вже з поточною роздільною здатністю рендерингу. І навпаки, якщо є невелика різниця між поточним пікселем і його сусідами, тоді немає потреби кидати нові промені, а результати візуалізації з нижчою роздільною здатністю можна бути прийняті як валідні і для поточної роздільної здатності. Цей процес триває, доки не буде досягнуто цільової роздільної здатності, після чого буде виконано всю недостатню вибірку. Отримані візуалізації потім можна зменшити до цільової роздільної здатності екрана, щоб зменшити вплив потенціальних дефектів.

Вибір сусідніх пікселів

Загалом, коли ми говоримо про піксельне оточення, ми маємо на увазі діапазон пікселів навколо даного цільового центрального пікселя, який програма «рендерить». Околиці з чотирьох пікселів описують сусідів, які знаходяться безпосередньо над/під і з боків цільового пікселя. А околиці восьми описують усі пікселі, які оточують центр, і так далі. Розмір даного околиці залежить від того, наскільки великі зміни діапазону можуть відбутися, але все ще впливають на вибірку цільового пікселя. Було виявлено, що збільшення розміру околиці пікселів не обов'язково корелює зі збільшенням точності. Навпаки, це може викликати помилкові спрацьовування, які вимагають детального вивчення, навіть якщо джерела змін знаходяться далеко від цільового пікселя. У випадку цієї статті для порівняння було обрано околиці, що складаються з восьми пікселів, що оточують цільовий піксель. Навіть якщо встановлено належний розмір околиці, необхідно визначити управління відмінностями з цільовим пікселем. Мета полягає в тому, щоб отримати нормалізоване значення, яке є результатом порівняння між цільовим пікселем і всіма його сусідніми пікселями.

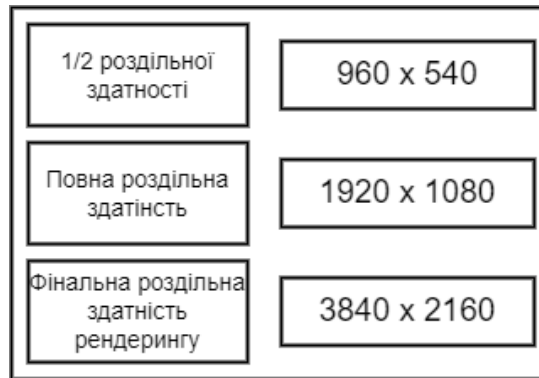


Рис. 3. Схематичне зображення поступового збільшення роздільної здатності рендерингу

Для початку порівняння між цільовим пікселем і його сусідами описується як різниця між очікуваним значенням для пікселя від інтерполяції між протилежними сусідами та фактичною інформацією всередині пікселя. Цей процес, показаний на рисунку 4, виявився більш ефективним, ніж порівняння даних цільового пікселя з кожним сусідом окремо. Потім настає проблема нормалізації відмінностей між кожним із порівнянь до єдиного корисного значення. Двома поширеними методами є нормалізація L1 і L2. Нормалізація L1 просто бере суму абсолютних різниць, тоді як нормалізація L2 бере суму квадратів кожної різниці, перш ніж брати корінь із цього значення. За допомогою тестування визначено, що взяття середнього значення L1 нормалізації відмінностей дало найбільш розумну метрику для того, чи потрібні нові зразки променів. Таким чином, для кожної пари сусідніх пікселів різницю між їх інтерпольованим значенням і цільовим пікселем додавали до суми, яку потім усереднювали за кількістю сусідів. Потім цю різницю можна порівняти з довільним порогом. Якщо нормалізована різниця більша за порогове значення, це означає, що щось цікаве відбувається в околицях пікселів, і потрібні подальші зразки променів із поточною роздільною здатністю. В іншому випадку дані з текстур з нижчою роздільною здатністю можуть бути взяті та використані для поточної роздільної здатності.

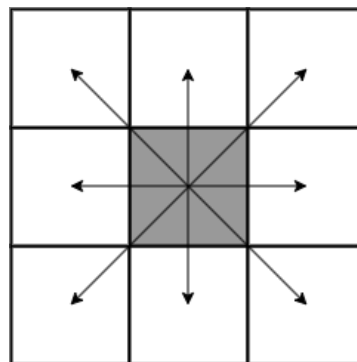


Рис. 4. Порівняння пікселів

Триадаптивна вибірка

Основна мета запропонованого методу полягає в тому, щоб змінити спосіб кидання променів на екран таким чином, щоб для досягнення того самого рівня інформації було необхідно менше зразків текстури. За стандартного підходу з різною роздільною здатністю один промінь потрапляє в центр кожного пікселя. Завдяки зміщенню напрямку променів на половину пікселя для кожного другого рядка створюється трикутна сітка. Ці дані все ще зберігаються у 2D об'єкт текстури, тому концептуально він записується та зчитується з квадратної сітки, як зазвичай. Метою цього процесу є зміна розподілу сусідніх пікселів. Під час аналізу розташування цільового пікселя є шість сусідів, з якими його можна порівняти. З областю розподілу між чотирма та вісьмома сусідами традиційного квадратного «семплера» ми припускаємо, що можна досягти подібного рівня якості порівняно з восьми піксельними околицями. Якщо це так, тоді виключення двох зразків із процесу має отримати незначний приріст продуктивності. Безпосередній недолік цієї системи полягає в тому, що, зміщуючи напрямки променів для кожного другого рядка, створюючи трикутну сітку, виникне очевидне спотворення в остаточному рендері, який відображається на екрані, який читається з квадратної сітки. Однак вирішення цього питання є

простим, оскільки програма повинна просто відобразити трикутну сітку назад у квадратну сітку. Це відображення можна здійснити шляхом інтерполяції відповідного кольору пікселя з навколишніх зразків. Тому для кожного пікселя в рядку зсуву необхідно інтерполювати між двома вибірками з обох боків або додати додаткові вибірки зверху та знизу пікселя, щоб отримати точний колір.

Продуктивність

Тестування кожного варіанту методу рендерингу відбувалося на однакових апаратних застосуваннях та графічних сценах. Для тесту буда взята за основу сцена корнелівської коробки. Як можна бачити на рисунку 5, класичний алгоритм рендерингу має продуктивність в, приблизно, 105 кадрів за секунду в середньому. У той самий час методи з використанням адаптивної вибірки показують приріст у приблизно 40%. Якщо порівнювати квадро- і триадаптивну вибірку, яка була представлена у статті, то маємо приріст у 5 кадрів у середньому. Це не є критичною різницею, але це доводить робоздатність модифікації.

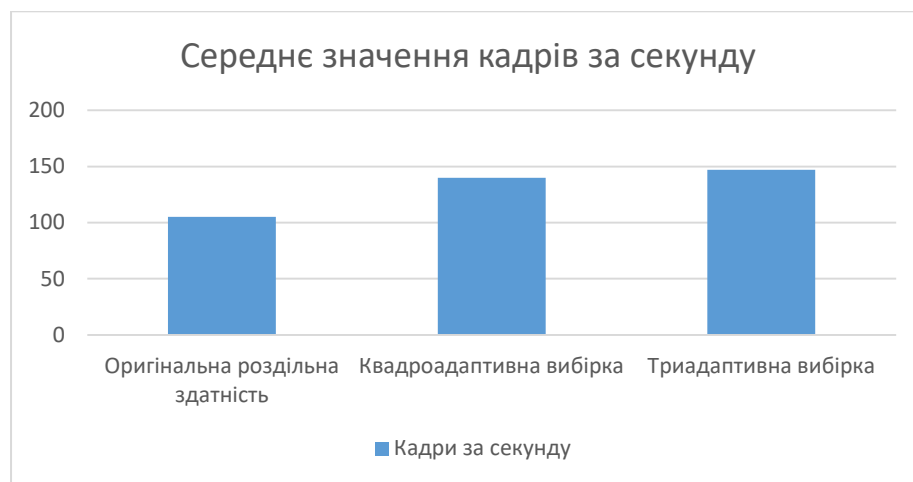


Рис. 5. Середнє значення кадрів за секунду при тестуванні

Висновки

У цій статті досліджено і запропоновано метод адаптування технології суперсемплінгу для збільшення продуктивності рендерингу зображення методом трасування променів у реальному часі. Такий підхід дозволить підвищити продуктивність систем рендерингу, які використовують алгоритм трасування променів, що було показано під час тестування.

Список бібліографічного опису

1. Дженетті, Дж. Трасування променів з адаптивною наддискретизацією в просторі об'єктів / Дженетті, Дж., Д. Гордон., 1993
2. Хачісука, Т. Багатовимірною адаптивна вибірка та реконструкція для трасування променів / Хачісука, Т., В. Ярош, Р. П. Вайстроффер., 2008. – (ACM Trans. Graph.).
3. Метт П. Фізично засноване рендеринг: від теорії до впровадження / П. Метт, Дж. Венцель, Г. Грег., 2016
4. NVIDIA TURING GPU ARCHITECTURE [Електронний ресурс]. – 2018. – Режим доступу до ресурсу: <https://images.nvidia.com/aem-dam/en-zz/Solutions/design-visualization/technologies/turing-architecture/NVIDIA-Turing-Architecture-Whitepaper.pdf>.
5. Фещенко І. О., Дадиверін В. В., Потапова К. Р. ОСОБЛИВОСТІ АПАРАТНО-ПРИСКОРЕНОГО АЛГОРИТМУ РЕНДЕРИНГУ НА ОСНОВІ ТРАСУВАННЯ ПРОМЕНІВ. MODERN RESEARCH IN WORLD SCIENCE, Lviv, 4-6 September 2022, 264-269c.

References

1. Genetti, J. Ray Tracing With Adaptive Supersampling in Object Space / Genetti, J., D. Gordon., 1993
2. Hachisuka, T. Multidimensional adaptive sampling and reconstruction for ray tracing / Hachisuka, T., W. Jarosz, R. P. Weistroffer., 2008. – (ACM Trans. Graph.).
3. Matt P. Physically Based Rendering: From Theory to Implementation / P. Matt, J. Wenzel, H. Greg., 2016.
4. NVIDIA TURING GPU ARCHITECTURE [E-source]. – 2018. – Resource access mode: <https://images.nvidia.com/aem-dam/en-zz/Solutions/design-visualization/technologies/turing-architecture/NVIDIA-Turing-Architecture-Whitepaper.pdf>.
5. Feshchenko I. O., Dadiverin V. V., Potapova K. R. FEATURES OF THE HARDWARE-ACCELERATED RENDERING ALGORITHM BASED ON RAY TRACING. MODERN RESEARCH IN WORLD SCIENCE, Lviv, September 4-6, 2022, pp. 264-269.

DOI: <https://doi.org/10.36910/6775-2524-0560-2022-49-12>

УДК 004.932

Бурбан Олександр Вікторович², викладач<https://orcid.org/0000-0001-6562-3936>**Поліщук Микола Миколайович**¹, к.т.н., доцент<https://orcid.org/0000-0002-1218-5925>**Бортник Катериня Яківна**¹, к.т.н., доцент<https://orcid.org/0000-0001-5282-099x>¹ Луцький національний технічний університет, м. Луцьк, Україна² Відокремлений структурний підрозділ «Волинський фаховий коледж Національного університету харчових технологій», м. Луцьк, Україна

СИСТЕМА ГОЛОСОВОГО КЕРУВАННЯ ОСВІТЛЕННЯМ НА БАЗІ WEMOS D1 MINI

Бурбан О.В., Поліщук М.М., Бортник К.Я. Система голосового керування освітленням на базі Wemos D1 Mini. Розроблено систему голосового керування освітленням на базі Wemos D1 Mini. Розроблено та описано електричну схему пристрою, а також прикладне програмне забезпечення для операційної системи Android. Дана система дозволяє керувати увімкненням та вимкненням кількох приладів освітлення із використанням голосового помічника Google Assistant.

Ключові слова: розумний будинок, голосове керування, освітлення, Google Assistant.

Burban O., Polishchuk M., Bortnyk K.Ya. System of voice control lighting based on Wemos D1 Mini. A voice control lighting system based on Wemos D1 Mini was developed. The electrical diagram of the device as well as the application software for the Android operating system was developed and described. This control system allows you to control the switching on and off of several lighting devices using the Google Assistant voice assistant.

Keywords: smart home, voice control, lighting, Google Assistant.

Постановка наукової проблеми. Невпинний розвиток сучасної техніки зумовлює постійне вдосконалення побутових приладів. Такі прилади повсякденного життя як чайник, праска, освітлення, тощо, окрім своїх звичайних функцій можуть отримувати різноманітні додаткові можливості пов'язані із їх керуванням. Вони можуть керуватись та налаштовуватись людиною, або аналізуючи певні навколишні показники автоматично реагувати на різноманітні зміни зовнішніх чинників. Побутові прилади із таким функціоналом дуже часто є елементами розумних будинків. Розумні прилади можуть працювати як окремих самостійний елемент, або бути складовою системи розумного будинку. Від ефективності роботи системи керування розумними приладами на пряму залежить ефективність виконання ними завдань за основним призначенням. Тому створення та дослідження роботи апаратно-програмних комплексів призначених для керування елементами розумного будинку є досить актуальним. Завданням даного наукового дослідження було створення системи для голосового керування кількома освітлювальними приладами за допомогою смартфона.

Аналіз досліджень. На сьогоднішній день існує безліч методів та засобів для реалізації керування розумними пристроями. Здебільшого такі рішення ґрунтуються на загальних засадах Інтернету речей, а саме: повсюдно поширеній комунікаційній інфраструктурі, глобальній ідентифікації усіх об'єктів, а також можливостей об'єктів отримувати і відправляти дані за допомогою персональної мережі чи мережі Інтернет [1].

На даний час широко застосовуються блоки контролю навколишнього середовища (ECU) які дозволяють керувати приладами у власному будинку чи на роботі [2]. Керування побутовими приладами може відбуватись як у межах локальної мережі, так і через глобальну мережу. Тобто користувачу для зміни параметрів роботи певного пристрою достатньо мати доступ до необхідної мережі та відповідну діалогову систему, яка буде відповідати за взаємодію із користувачем.

На сьогодні існують мультимодальні діалогові системи, які обробляють по кілька комбінованих режимів введення користувача, таких як мова, зображення, відео, дотик, ручні жести, погляд, а також рух голови та тіла [3]. При створенні програмних рішень для використання діалогової системи можна використовувати API персональних голосових помічників, таких як Google Assistant, Amazon Alexa, тощо. Використання таких діалогових систем дозволяє розробляти прикладні програмні продукти з високою ефективністю роботи програмного застосування та мінімальними затратами часу. Ефективність таких голосових систем керування значною мірою

буде залежати від ефективності перетворення голосових сигналів в цифровий код. На правильність таких перетворень можуть впливати якість та властивості первинних датчиків, тобто мікрофонів які сприймають звук, АЦП які оцифровують отриманий сигнал, а також параметри використаного звукового сигналу: тембр, висота та гучність голосу, тощо [4]. На сьогоднішній день у літературі є обмежена кількість досліджень впливу таких технологічних та фізичних параметрів на ефективність роботи керуючої системи. Тому створення системи голосового керування освітленням з метою дослідження її ефективності є досить актуальною.

Виклад основного матеріалу й обґрунтування отриманих результатів дослідження. Найбільш зручним та поширеним способом керування розумними пристроями є використання мобільних пристроїв, які працюють під управлінням операційних систем Android або IOS. Тому в загальному випадку створювана система керування повинна складатись з двох елементів: керуючої частини та виконавчої. Керуюча частина являє собою мобільний телефон з можливістю під'єднання до мережі WiFi та відповідне програмне забезпечення. Виконавча частина повинна складатись з мікроконтролерного пристрою з можливістю підключення до WiFi мережі та ключів, які будуть замикати та розмикати електричні кола навантаження.

У якості мікроконтролерного пристрою, який буде безпосередньо керувати електричними ключами було вибрано Wi-Fi модуль WeMos D1 mini (рис. 1). Плата WeMos D1 mini – це аналог плати NodeMCU v3 виконаний на базі мікросхеми ESP8266 версії ESP-12F. Основною перевагою цього модуля є те, що для роботи WeMos D1 mini не потрібен зовнішній мікроконтролер або інший керуючий пристрій, оскільки крім Wi-Fi модуля ESP-12F у ньому вбудований 32-бітний мікроконтролер з тактовою частотою 80 МГц, а також чіп флеш-пам'яті на 4МБ. Також він має менші габарити у порівнянні з іншими аналогами.

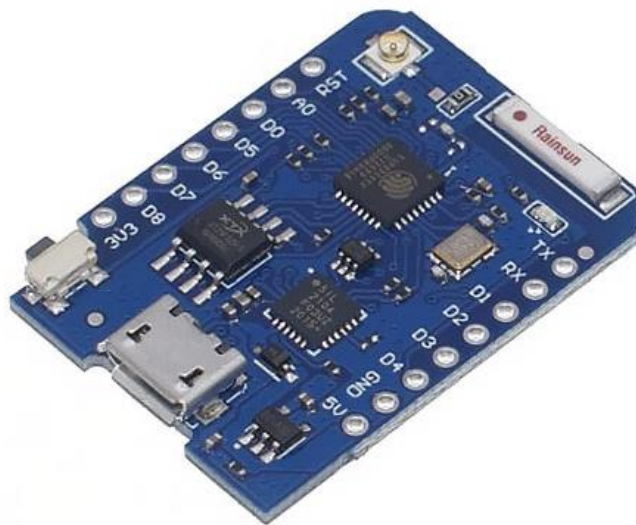


Рис. 1. Зовнішній вигляд WeMos D1 mini

Модуль WeMos D1 mini є досить часто використовуваним елементом при створенні різноманітних пристроїв для Інтернету речей, а також систем віддаленого моніторингу або керування, різних автономних датчиках тощо. Дана плата підтримує кілька варіантів роботи з Wi-Fi мережами. Вона може бути як клієнтом Wi-Fi мережі, так і сама створювати Wi-Fi точку доступу. Тому використання WeMos D1 mini, у порівнянні із використанням окремої ESP8266, істотно спрощує роботу при конструюванні, оскільки в даному модулі вже містяться чимало вбудованих необхідних компонентів, таких як USB-UART перетворювач, лінійний стабілізатор напруги, а також контакти вже розведені до роз'ємів із стандартним кроком 2.54 мм. Призначення портів WeMos D1 mini зображено на рисунку 2.

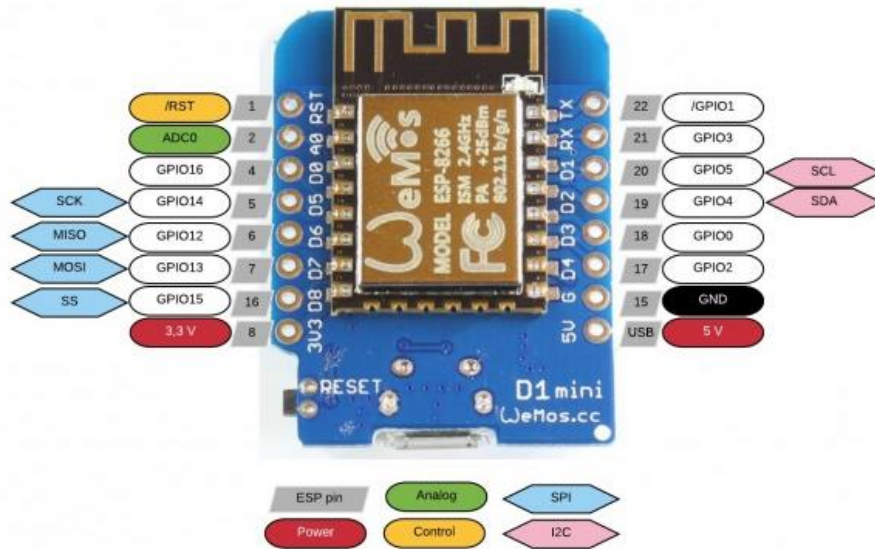
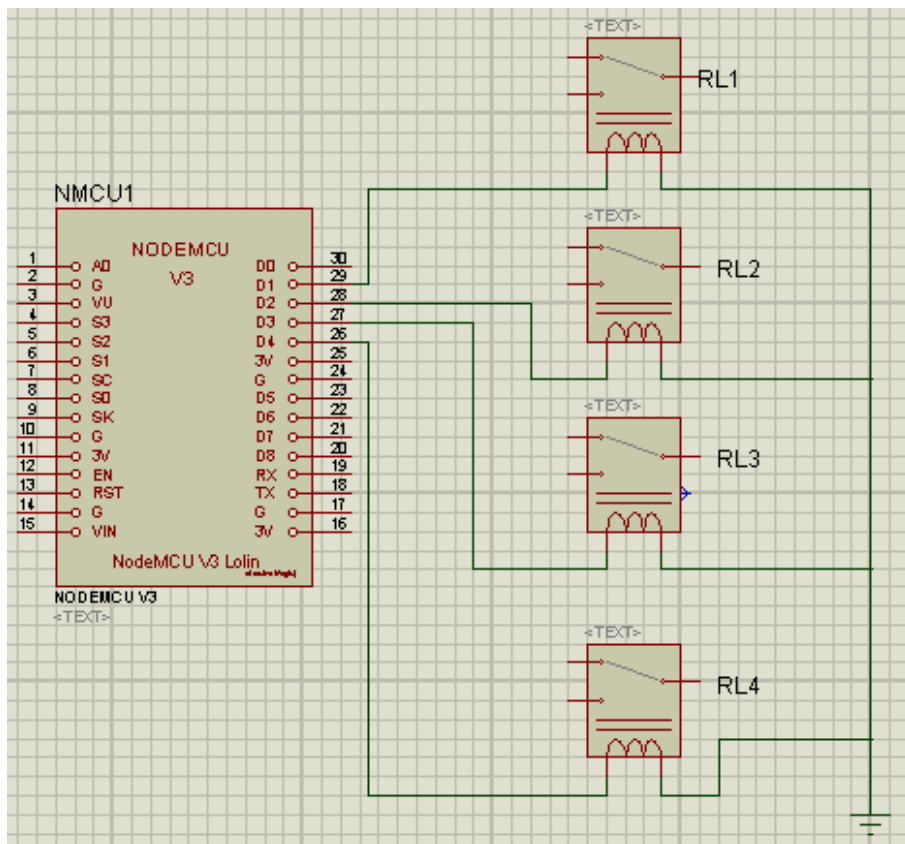
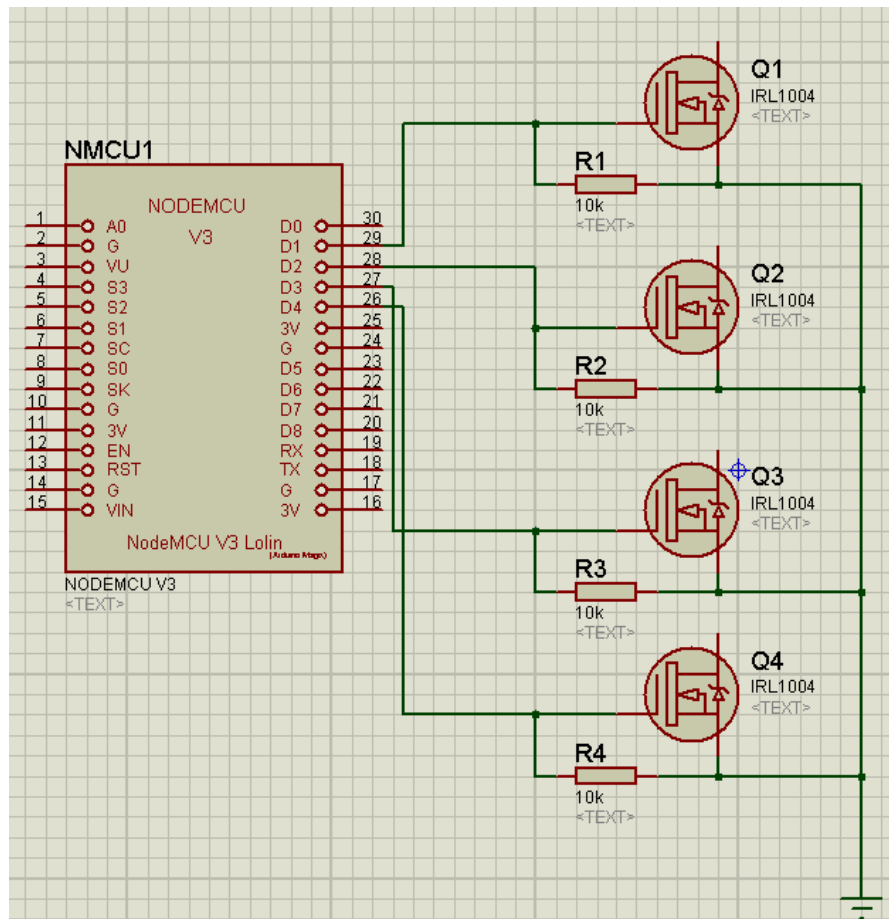


Рис. 2. Призначення портів WeMos D1 mini

При створенні даного комплексу було визначено, що він повинен керувати чотирма споживачами електричної енергії. Для комутації кіл навантаження розглядались два можливих випадки. Перший – це живлення побутових приладів від електричної мережі із напругою 220 В за допомогою масиву реле із низькою напругою керування. Другий – це живлення світлодіодних стрічок, або світлодіодних матриць за допомогою польових транзисторів. Схеми підключень зображені на рисунку 3.



а)



б)

Рис. 3. Схеми підключення споживачів: а) підключення за допомогою електромагнітних реле, б) підключення за допомогою польових транзисторів

Прикладне програмне забезпечення для даного проекту розроблялось за допомогою сервісу MIT App Inventor. Даний сервіс - це інтегроване середовище розробки веб-додатків, Він був заснований компанією Google, а на теперішній час підтримується Массачусетським технологічним інститутом. Дане середовище дозволяє створювати прикладне програмне забезпечення для двох операційних систем: Android та iOS. Воно є безкоштовний і має відкритий вихідний код, та поширюється за подвійною ліцензією: Creative Commons Attribution ShareAlike 3.0 Unported та Apache 2.0. MIT App Inventor використовує інтуїтивний графічний інтерфейс користувача, дуже подібний до мов програмування Scratch та StarLogo. Такі середовища розробки дозволяють користувачам створювати програми для пристроїв які працюють на операційних системах Android, або IOS шляхом перетягування візуальних об'єктів. [5]

Хоча даний сервіс створювався в освітніх цілях, проте на даний час зважаючи на його зростаючі можливості він все частіше використовується у випадках, коли необхідно швидко розробити програмний продукт з мінімальним функціоналом для конкретних задач [6]. Використання даного сервісу при написанні нашого застосунку також було обумовлено саме простотою створення програмного продукту, що значно зменшило затрати часу у порівнянні з написанням програмного коду у інших середовищах розробки.

Програмний продукт повинен був забезпечувати можливість керувати кількома електричними ключами як за допомогою голосових команд з використанням сервісу Google Assistant, так і шляхом натискання керуючих кнопок у мобільного додатку. У разі успішного виконання команди додаток повинен проінформувати голосовим повідомленням про успішне виконання відповідної команди.

Для кожного електричного ключа на екрані навпроти їх ідентифікаторів розміщено по дві кнопки: «ON» та «OFF». Для виклику голосового асистента на робочому полі розміщено елемент «Speech Text». Результат розробки графічного макету програми зображено на рисунку 4.

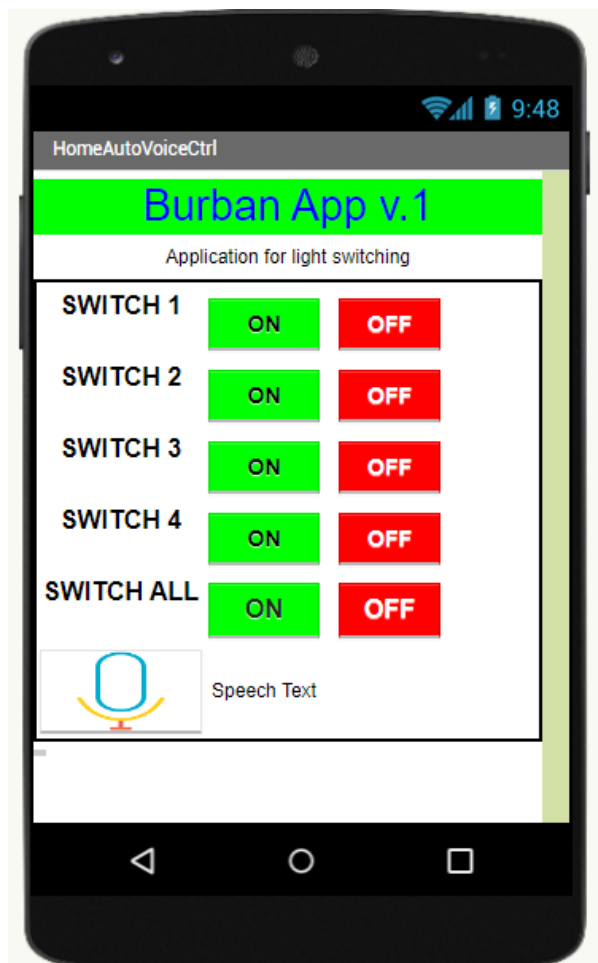
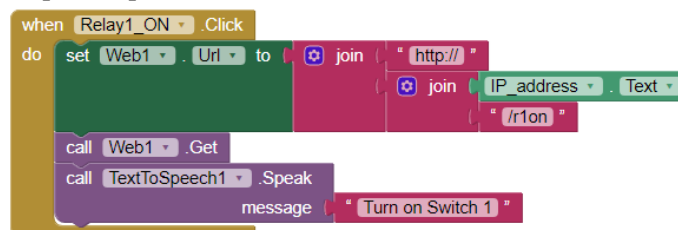


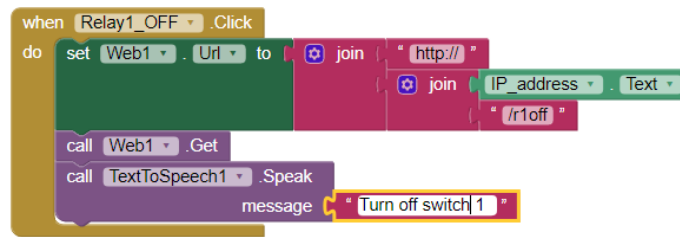
Рис. 4. Графічний макет програми керування

Даний макет передбачає керування чотирма споживачами з ідентифікаторами «SWITCH 1», «SWITCH 2», «SWITCH 3» та «SWITCH 4». Вмикання «SWITCH 1» відбувається шляхом натискання кнопки «ON» або введенням голосової команди «Turn on switch one». Відповідно для вимикання «SWITCH 1» необхідно натиснути кнопки «OFF» або ввести голосову команду «Turn of switch one». Вмикання та вимикання «SWITCH 2», «SWITCH 3», «SWITCH 4» проводиться аналогічно. Для введення голосових команд використовується кнопка «Speech Text». При її натисканні викликається голосовий асистент Google Assistant, який проводить розпізнання голосу та дозволяє отримати значення відповідної команди.

На рисунку 5,а та 5,б зображено вигляд фрагментів програми у MIT App Inventor для команд увімкнення та вимкнення першого реле відповідно.



а)



б)

Рис. 5. Фрагменти програмних блоків для увімкнення та вимкнення споживачів.

Висновки та перспективи подальшого дослідження. У даній роботі було спроектовано та виготовлено просту систему голосового керування освітленням. Вона дозволяє керувати чотирма електричними приладами за допомогою мобільного телефону та наявної WiFi мережі. Хоча дана система проектувалась для керування світловими приладами, проте її можна використовувати для будь-яких електричних приладів. Також при потребі дану систему можна модернізувати, для зміни кількості керованих пристроїв, або ускладнення поведінки пристроїв в залежності від конкретно поставленої задачі. Можливе також застосування таймерів чи зміни споживаної потужності для кожного пристрою окремо. Такі системи можна використовувати як елементи розумного будинку. Також дану систему можна використовувати в наукових цілях для дослідження параметрів роботи систем віддаленого керування.

Список бібліографічного опису

1. Zeinab K. A. M.; Elmustafa, S. A. A. (2017) Internet of things applications, challenges and related future technologies. World Scientific News. V 67, №2 P 126-148.
2. Noda K. (2018). Google Home: smart speaker as environmental control unit. Disability and Rehabilitation: Assistive Technology. №7. P. 674-675, doi: 10.1080/17483107.2017.1369589
3. Kępuska, V., Bohouta, G. (2018) Next-generation of virtual personal assistants (Microsoft Cortana, Apple Siri, Amazon Alexa and Google Home) 2018 IEEE 8th Annual Computing and Communication Workshop and Conference (CCWC). P.99 – 103, doi: 10.1109/CCWC.2018.8301638.
4. Бурбан О. В., Поліщук М. М. (2022) Дослідження роботи систем автоматичного вмикання світла із звуковим керуванням. Комплексне забезпечення якості технологічних процесів та систем (КЗЯТПС – 2022) : матеріали тез доповідей XII Міжнародної науково-практичної конференції. Національний університет «Чернігівська політехніка». С. 258–259.
5. Hardesty, L. (August 19, 2010). "The MIT roots of Google's new software". MIT News Office.
6. Pokress, Shaileen Crawford, and José Juan Dominguez Veiga. (2013) MIT App Inventor: Enabling personal mobile computing. arXiv preprint arXiv:1310.2830.

References

1. Zeinab K. A. M.; Elmustafa, S. A. A. (2017) Internet of things applications, challenges and related future technologies. World Scientific News. V 67, №2 P 126-148.
2. Noda K. (2018) Google Home: smart speaker as environmental control unit. Disability and Rehabilitation: Assistive Technology. №7. P. 674-675, doi: 10.1080/17483107.2017.1369589
3. Kępuska, V., Bohouta, G. (2018) Next-generation of virtual personal assistants (Microsoft Cortana, Apple Siri, Amazon Alexa and Google Home) 2018 IEEE 8th Annual Computing and Communication Workshop and Conference (CCWC). P.99 – 103, doi: 10.1109/CCWC.2018.8301638.
4. Burban O.V., Polishchuk M.M. (2022) Study of the operation of automatic light switching systems with sound control. Comprehensive quality assurance of technological processes and systems (KZYATPS - 2022): materials of abstracts of reports of the XII International Scientific and Practical Conference. Chernihiv Polytechnic National University. P. 258–259.
5. Hardesty, L. (August 19, 2010). "The MIT roots of Google's new software". MIT News Office.
6. Pokress, Shaileen Crawford, and José Juan Dominguez Veiga. (2013) MIT App Inventor: Enabling personal mobile computing. arXiv preprint arXiv:1310.2830.

DOI: <https://doi.org/10.36910/6775-2524-0560-2022-49-13>

УДК 378.37:004

Поліщук Микола Миколайович, к.т.н., доцент

<https://orcid.org/0000-0002-1218-5925>

Саварин Павло Вікторович, к.пед.н., доцент

<https://orcid.org/0000-0001-7140-7068>

Фуркало Сергій Юрійович, магістр

Луцький національний технічний університет, м. Луцьк, Україна

МЕТОДИ WEBHOOKS ТА LONG POLLING ДЛЯ РОЗМІЩЕННЯ TELEGRAM-БОТА

Поліщук М.М., Саварин П.В., Фуркало С.Ю. Методи Webhooks та Long Polling для розміщення Telegram-бота. У статті розглянуто особливості розміщення Telegram-бота на прикладі методів Webhooks та Long Polling. Розкрито механізм та специфіку їх роботи. Описано випадки їх використання.

Ключові слова: Telegram, бот, Webhooks, Long Polling.

Polishchuk M., Savaryn P., Furkalo S. Webhooks and Long Polling methods for hosting a Telegram bot. The article examines the features of placing a Telegram bot using the Webhooks and Long Polling methods as an example. The mechanism and specifics of their work are disclosed. Cases of their use are described.

Keywords: Telegram, bot, Webhooks, Long Polling.

Постановка наукової проблеми. Сьогодні більшість людей використовує месенджери для віддаленої комунікації зі знайомим, друзями чи по роботі. Проте, сучасні месенджери дозволяють не тільки обмінюватись повідомленнями чи здійснювати онлайн дзвінки, а й слугують засобом автоматизації. Для цього компаніями-розробниками месенджерів була забезпечена можливість створення чат-ботів. Функціонал таких ботів може бути надзвичайно широким та різноманітним, а головною перевагою є автономність та швидкодія. Для забезпечення даних характеристик у розробника постає необхідність аналізу методів розміщення та їх використання в залежності від поставлених вимог та майбутнього функціоналу бота.

Найбільш поширеними в Україні месенджерами, що мають можливість створення ботів є Telegram, Viber та Instagram. Серед них найбільш широку документацію та функціонал має саме Telegram. Кінцевою метою будь-якої інтеграції API є ефективний обмін даними між програмами, щоб забезпечити велику цінність для користувачів. Щоб полегшити це, інтеграція повинна надати метод виявлення змін, подій, які відбуваються в додатку кінцевої точки. В даний час два найпопулярніші інструменти управління подіями – це Webhooks та Long Polling. Саме вибір одного з них вирішує принцип роботи боту та швидкість його роботи, а також зручність обслуговування. У даній статті розглянемо дані методи, принцип та особливості їх функціонування.

Аналіз досліджень. На сьогоднішній день існує два основних методи обробки подій при розробці Telegram-бота: Webhooks та Long Polling. Здебільшого вибір конкретного методу залежить від специфікації сервера, що використовується для розміщення. Основною технічною особливістю методу Webhooks є необхідність активного SSL-сертифікату та доменного імені. Це пов'язано з тим, що всі запити будуть надходити та проходити обробку саме з їх використанням.

На даний час широко застосовуються обидва методи при створенні чат-ботів на базі Telegram. Кожен із них має свою специфіку та випадки використання. Проте, більш вживаним все ж можна назвати саме Long Polling, оскільки він є більш простим у використанні.

На сьогоднішній день у літературі є обмежена кількість досліджень особливостей використання даних методів. Книга Metthias Biehl «Webhooks – Events for RESTfull APIs» [1] є чудовим представником наукової роботи у даному напрямку. Автор добре розкрив специфіку використання Webhooks та інших методів при роботі з API. Дана робота також вміщує інформацію і про Polling та Long Polling, які також будуть оглянуті в даній статті.

Вебхук набагато ефективніший за опитування з точки зору ресурсів і комунікації. Zapier провів дослідження 30 мільйонів запитів на опитування, зроблених через їхній сервіс, і виявив, що 98,5% опитувань були витрачені даремно, і вони витратили в 66 разів більше ресурсів на опитування [2].

Завжди свіжі дані. Сама природа Webhook і той факт, що вони зазвичай викликаються подіями, означає, що вони надають вам інформацію в режимі реального часу. У зв'язку з цим, якщо ви

хочете, щоб інформація була максимально наближеною до реального часу, вам слід вибрати використання Webhook замість опитування (Long Polling) [3].

Webhooks перевершують опитування з точки зору релевантності даних, ефективності зв'язку та вартості інфраструктури.

Виклад основного матеріалу й обґрунтування отриманих результатів дослідження.

Отримання оновлень з серверів Telegram може відбуватись двома способами. Вони називаються Webhooks та Long Polling [4]. Вибір одного з цих методів можна розглядати як кінцевий тип розгортання чат-бота на платформі месенджера Telegram. Іншими словами, існують два принципово різних способи розміщення ботів і вони відрізняються способом доставки і обробки повідомлень вашим Telegram-ботом.

Вибір конкретного методу має велике значення у момент, коли вам необхідно вирішити, де буде розміщуватись ваш чат-бот. Наприклад, деякі хостинг-провайдери надають можливість розгортання лише одним із двох типів. Далі розглянемо більш детально специфіку кожного з методів та наведемо приклади їх використання.

Telegram Webhooks – це технологія, що дозволяє відстежувати події чату в режимі реального часу і відправляти відповідні дані на вказану адресу. Таким чином, даний метод використовується при створенні бота, якщо є необхідність того, щоб система автоматично реагувала на повідомлення та дії користувача. Отримуючи дані про них, сервер проводить їх відправку безпосередньо самому програмному рішенню(боту) до алгоритму, в який закладена їх обробка.

Webhooks дозволяють різним програмам і платформам обмінюватися даними у вигляді повідомлень. Цей зв'язок відбувається в режимі реального часу аз допомогою URL-адреси Webhook. Інтеграція Telegram Webhook може бути дуже корисною для отримання даних від інших програм сторонніх розробників. Telegram — це хмарна міжплатформна служба обміну миттєвими повідомленнями. Ця служба обміну повідомленнями пропонує відеодзвінки з наскрізним шифруванням, VoIP (протокол передачі голосу через Інтернет), обмін файлами та багато інших функцій.

Вебхук (також відомий як зворотний веб-дзвінок або HTTP push API) — це інструмент, який надає дані в реальному часі іншим програмам. На відміну від традиційних API (інтерфейсу прикладного програмування), вам не потрібно опитувати дані, щоб отримати дані в реальному часі. Веб-хуки стали значно ефективнішими як для постачальника, так і для споживача. Ви також можете використовувати Webhooks, щоб налаштувати подію на одному веб-сайті, яка ініціює дію на іншому. Це зменшує необхідність постійного спостереження за об'єктом. Єдиним недоліком Webhooks є складність їх розміщення.

Налаштування вебхуку означає, що ви надаєте Telegram URL-адресу, доступну з загальнодоступного Інтернету. Щоразу, коли вашому боту надходить нове повідомлення, Telegram візьме на себе ініціативу та надішле запит із об'єктом оновлення на ваш сервер.

Коли надходить запит на вебхук, ваш бот може викликати один метод у відповіді. Як перевага, це позбавляє вашого бота від створення одного запиту HTTP на оновлення. Однак у використанні цього є ряд недоліків:

- Ви не зможете обробляти потенційні помилки відповідного виклику API. Це включає помилки обмеження швидкості, тому ви не можете бути гарантовано, що ваш запит матиме якийсь ефект.
- Що ще важливіше, ви також не матимете доступу до об'єкта відповіді. Наприклад, виклик «sendMessage» не дасть вам доступу до надісланого повідомлення.
- Крім того, скасувати запит неможливо. Команду «AbortSignal» буде проігноровано.

Принцип роботи Telegram-бота на основі метода Webhooks виглядає наступним чином, зображеним на рисунку 1.

Коли Telegram надсилає оновлення з одного чату вашому боту, він чекатиме, поки ви завершите запит, перш ніж доставити наступне оновлення, яке належить цьому чату. Іншими

словами, Telegram послідовно надсилатиме оновлення з одного чату, а оновлення з різних чатів надсилатимуться одночасно.

Telegram намагається переконатися, що ваш бот отримує всі оновлення. Це означає, що якщо доставка оновлення для чату не вдається, наступні оновлення будуть поставлені в чергу, доки перше оновлення не буде успішним.

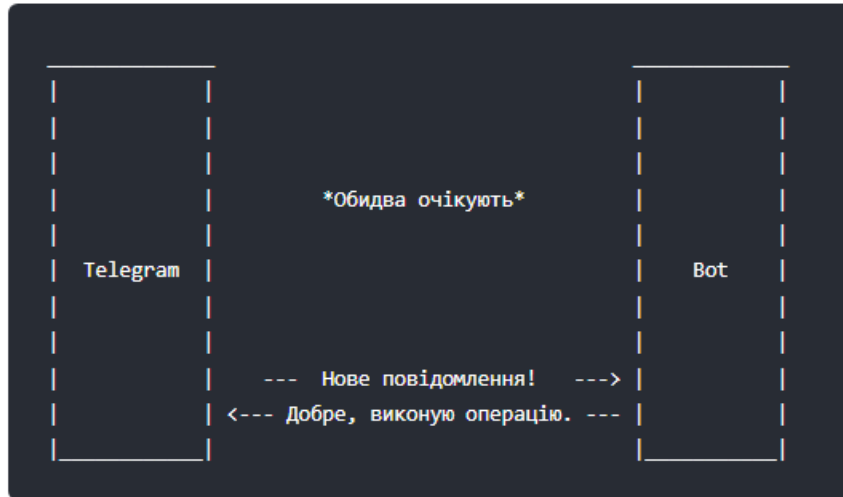


Рис. 1. Принцип роботи Telegram-бота на основі методу Webhooks

Telegram має тайм-аут для кожного оновлення, яке він надсилає на вашу кінцеву точку вебхуку. Якщо ви не завершите запит на вебхук досить швидко, Telegram повторно надішле оновлення, припускаючи, що воно не було доставлено. У результаті ваш бот може неочікувано обробити те саме оновлення кілька разів. Це означає, що він виконуватиме всю обробку оновлень, включаючи надсилання будь-яких відповідних повідомлень, кілька разів.

Таким чином, можемо стикнутися із ситуацією, зображеною на рисунку 2.

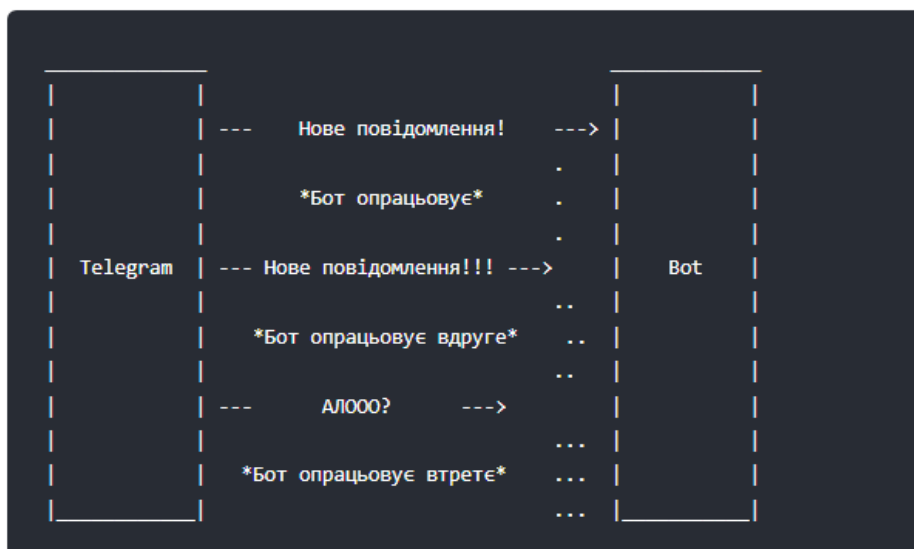


Рис. 2. Проблема із не завершенням запиту при використанні Webhooks

Коли Telegram надішле оновлення вашому боту вдруге, малоймовірно, що ви впораетесь з ним швидше, ніж у перший раз. У результаті, імовірно, знову мине тайм-аут, і Telegram знову надішле оновлення. Таким чином, ваш бот побачить оновлення не просто два рази, а кілька

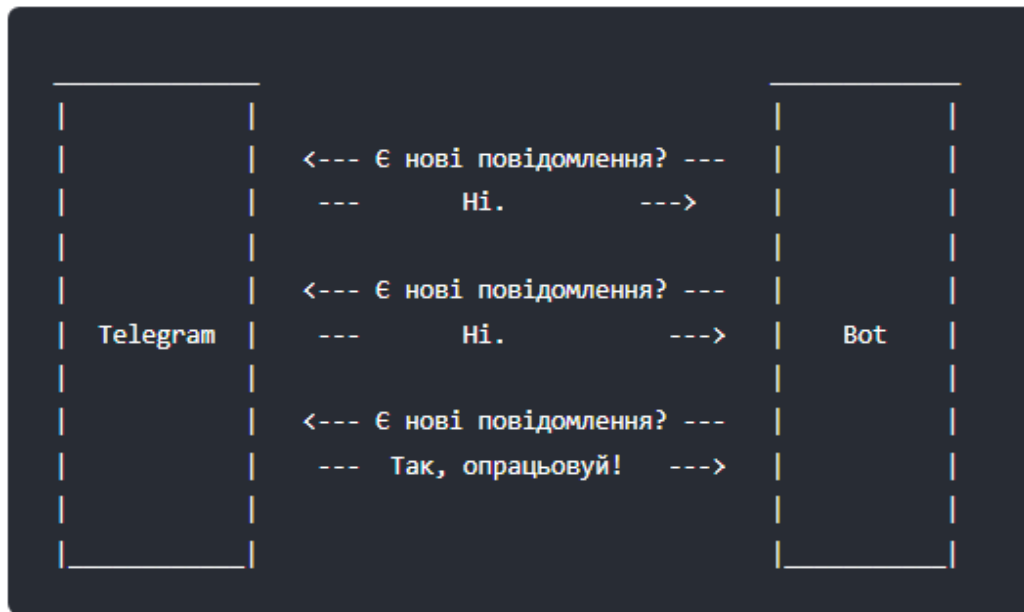


Рис. 4. Принцип роботи методу Polling

Одразу можна побачити, що це має деякі недоліки. Ваш бот отримує нові повідомлення лише тоді, коли запитує, тобто приблизно кожні кілька секунд. Щоб ваш бот відповідав швидше, ви можете просто надіслати більше запитів і не чекати так довго між ними. Наприклад, можна зробити так, щоб оновлення запитувались кожної мілісекунди.

Замість того, щоб спамити сервери Telegram, можна використовувати метод довгого опитування Long Polling замість звичайного (короткого). Довге опитування означає, що сервер завчасно надсилає запит до Telegram із запитом на нові оновлення. Якщо повідомлень немає, Telegram залишатиме з'єднання відкритим до надходження нових повідомлень, а потім відповідатиме на запит цими новими повідомленнями. Таким чином, отримаємо принцип обробки даних, зображений на рисунку 5.

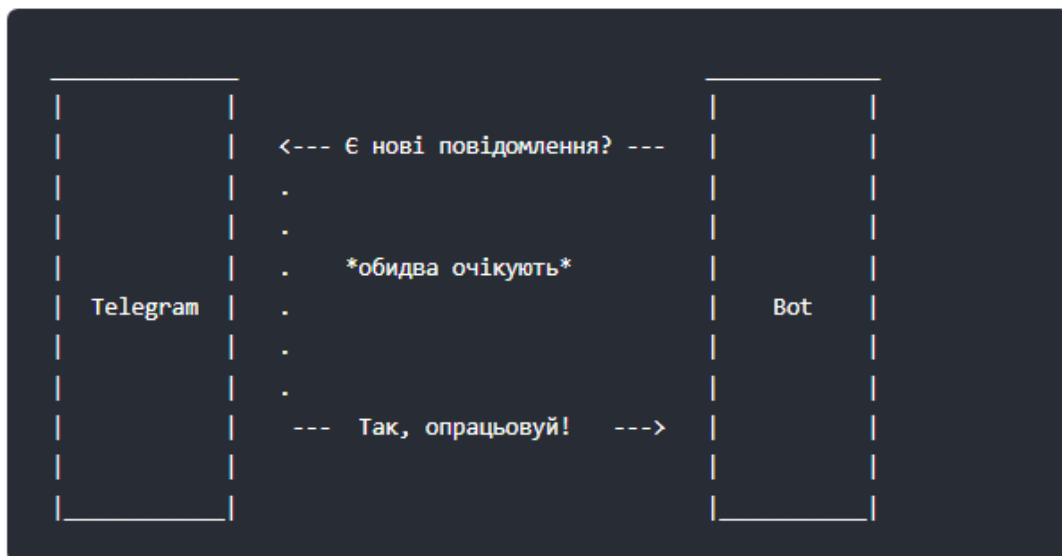


Рис. 5. Принцип роботи методу Long Polling

Основна перевага довгого опитування над веб-хуками полягає в тому, що воно простіше. Вам не потрібен домен або загальнодоступна URL-адреса. Вам не потрібно мати справу з

налаштуванням SSL-сертифікатів, якщо ви використовуєте бота на VPS. Таким чином, немає необхідності в додаткових налаштуваннях. Під час навантаження ви повністю контролюєте кількість повідомлень, які можете обробити.

Випадки, коли варто використовувати метод Long Polling:

- Під час розробки на вашій локальній машині.
- На більшості серверів.
- На розміщених «серверних» екземплярах, тобто машинах, на яких ваш бот активно працює 24/7.

Основна перевага вебхуків перед довгим опитуванням полягає в тому, що вони швидші. Ви заощаджуєте масу зайвих запитів. Вам не потрібно постійно тримати мережеве з'єднання відкритим. Ви можете використовувати служби, які автоматично зменшують масштаб вашої інфраструктури до нуля, коли запити не надходять. Якщо ви хочете, ви навіть можете зробити виклик API, відповідаючи на запит Telegram, хоча це має ряд недоліків.

Випадки, коли краще використовувати метод Webhooks:

- На серверах із сертифікатами SSL [6].
- На розміщених екземплярах «інтерфейсу», які масштабуються відповідно до навантаження.
- На безсерверних платформах, таких як хмарні функції або програмовані крайові мережі.

Таким чином, Long Polling є більш зручним для використання методом та має значно менше недоліків. Основна перевага Webhooks у свою чергу, це його швидкодія.

Висновки та перспективи подальшого дослідження. У даній роботі було проведено аналіз методів обробки даних Telegram-ботом при його розміщенні. В результаті отримано порівняльну характеристику Webhooks та Long Polling. Розглянуто їх особливості та випадки застосування. Підводячи підсумки по даній роботі, можна дійти висновку, що у більшості випадків найбільш доречним є використання методу Long Polling, оскільки він має значно менше недоліків, порівняно з Webhooks. Більшість із них пов'язані безпосередньо з особливостями розміщення, серед яких необхідність активного сертифікату SSL та dome ного імені.

Перспективи подальших досліджень ґрунтуються на вдосконаленні алгоритмів обробки даних, що використовуються під час створення чат-ботів та програмних засобів в цілому.

Список бібліографічного опису

1. Behl M Webhooks – Events for RESTful APIs/ M. Behl // API-University Press. – 2017.– P. 23-43, ISBN-13: 978-1979717069, isbn-10: 1979717060
2. Server-Push Documents (HTML & XHTML: The Definitive Guide) Archived 2008-04-17 at the Wayback Machine. O'Reilly book explaining server-push.
3. "What is push notifications & How do push notifications work?" // електрон. текст. Дані URL: <https://push-maze.com> (дата звернення: 01.10.2022).
4. Maulaaya, F. R., Arifin, M. Z., & Hariono, T. (2019). RANCANG BANGUN “TELEGRAM BOT API” UNTUK LAYANAN SISTEM INFORMASI AKADEMIK DI UNWAHA MENGGUNAKAN METODE LONG POLLING. SAINTEKBU, 11(1), 68–77.
5. Kurniawan A., Lau W. - Practical Azure Functions/ Agus Kurniawan, Wely Lau // Springer – 2017 – P 23-49, ISBN: 978-1-4842-5066-2
6. Шайдаров, С.В. Чат-бот для telegram. Віртуальний співрозмовник та розумний помічник [Текст]: робота на здобуття кваліфікаційного ступеня бакалавра; спец.: 122 - комп'ютерні науки (інформатика) / С.В. Шайдаров; наук. керівник М.С. Бабій. - Суми: СумДУ, 2020. - 32 с.
7. S. Kostiuchko, M. Polishchuk, O. Zabolotnyi, A. Tkachuk, B. Twarog The Auxiliary Parametric Sensitivity Method as a Means of Improving Project Management Analysis and Synthesis of Executive Elements. Lecture Notes of the Institute for Computer Sciences. Social-Informatics and Telecommunications Engineering. Springer International Publishing, 2021. Pp. 174–184.
8. Костючко С., Багнюк Н., Кузьмич О., Поліщук М., Кирилюк Л. Біометрична ідентифікація засобами Python та Raspberry Pi. Науковий журнал «Комп'ютерно-інтегровані технології: освіта, наука, виробництво». Луцьк: Видавництво ЛНТУ. Вип. 42. 2021. С. 142-146.

9.Savaryn, P., Strekha, V., Brych, M., Kabak, V., Polishchuk, M. The Original Method of Controlling a Computer Using Distance Sensors. Proceedings - 16th International Conference on Advanced Trends in Radioelectronics, Telecommunications and Computer Engineering, TCSET 2022, 2022, pp. 683–688.

References

1. Behl M Webhooks – Events for RESTful APIs/ M. Behl // API-University Press. – 2017.– P. 23-43, ISBN-13: 978-1979717069, isbn-10: 1979717060
2. Server-Push Documents (HTML & XHTML: The Definitive Guide) Archived 2008-04-17 at the Wayback Machine. O'Reilly book explaining server-push.
3. "What is push notifications & How do push notifications work?" // el. text. URL data: <https://push-maze.com> (access date: 10/01/2022).
4. Maulayya, F. R., Arifin, M. Z., & Hariono, T. (2019). RANCANG BANGUN “TELEGRAM BOT API” UNTUK LAYANAN SISTEM INFORMASI AKADEMIK DI UNWAHA MENGGUNAKAN METODE LONG POLLING. SAINTEKBU, 11(1), 68–77.
5. Kurniawan A., Lau W. - Practical Azure Functions/ Agus Kurniawan, Wely Lau // Springer – 2017 – P 23-49, ISBN: 978-1-4842-5066-2
6. Shaidarov, E. Chat bot for telegram. Virtual interlocutor and smart assistant.[Text]: work to obtain a bachelor's degree; spec.: 122 - computer science (informatics) / E.V. Shaidarov; of science head M.S. Womanizer. - Sumy: Sumy State University, 2020. - 32 p.
7. S. Kostyuchko, M. Polishchuk, O. Zabolotnyi, A. Tkachuk, B. Twarog The Auxiliary Parametric Sensitivity Method as a Means of Improving Project Management Analysis and Synthesis of Executive Elements. Lecture Notes of the Institute for Computer Sciences. Social-Informatics and Telecommunications Engineering. Springer International Publishing, 2021. Pp. 174–184.
8. Костючко С., Багнюк Н., Кузьмич О., Поліщук М., Кирилюк Л. Біометрична ідентифікація засобами Python та Raspberry Pi. Науковий журнал «Комп'ютерно-інтегровані технології: освіта, наука, виробництво». Луцьк: Видавництво ЛНТУ. Вип. 42. 2021. С. 142-146.
- 9.Savaryn, P., Strekha, V., Brych, M., Kabak, V., Polishchuk, M. The Original Method of Controlling a Computer Using Distance Sensors. Proceedings - 16th International Conference on Advanced Trends in Radioelectronics, Telecommunications and Computer Engineering, TCSET 2022, 2022, pp. 683–688.

DOI: <https://doi.org/10.36910/6775-2524-0560-2022-49-14>

УДК 621.391.8

Радзівілов Григорій Данилович, к.т.н., доцент

<https://orcid.org/0000-0002-6047-1897>

Військовий інститут телекомунікацій та інформатизації ім. Героїв Крут, м. Київ, Україна

АНАЛІЗ ДОСТОВІРНОСТІ ПРИЙОМУ ПОВІДОМЛЕНЬ В ІОТ ТЕХНОЛОГІЯХ З ОДНОСТОРОННІМИ ПРОТОКОЛАМИ ВЗАЄМОДІЇ

Радзівілов Г.Д. Аналіз достовірності прийому повідомлень в ІоТ технологіях з односторонніми протоколами взаємодії. Проведено аналіз достовірності прийому інформації в ІоТ технологіях з односторонніми протоколами взаємодії. Встановлено закономірності впливу завадової обстановки на значення ймовірності правильного прийому повідомлень з урахуванням можливих хибних спрацювань та застосованих методів підвищення достовірності прийому. Розглянуто можливість виникнення хибних спрацювань. Визначено напрямки досліджень для забезпечення використання ІоТ технологій з односторонніми протоколами взаємодії при проведенні сучасних бойових дій в умовах складної завадової обстановки та впливу структурних завад.

Ключові слова: ІоТ технології з односторонніми протоколами взаємодії, достовірність прийому повідомлень, ймовірність правильного прийому повідомлень, ймовірність хибного спрацювання.

Radzivilov H. The reliability analysis of receiving messages in IoT technologies with one-way interaction protocols. An analysis of information receiving reliability in IoT technologies with one-way interaction protocols was carried out. The influence regularities of the interference environment on the value of messages reception correct probability have been established. The possibility of false detecting and the applied methods of increasing the message reception reliability are taken into account. Research directions to ensure the possibility of using IoT technologies with one-way interaction protocols during the conduct of modern combat operations have been determined. It was made for difficult interference environment and the influence of structural interference.

Keywords: IoT technologies with one-way interaction protocols, reliability of message reception, probability of correct message reception, probability of false detecting.

Постановка наукового завдання. При проведенні сучасних бойових дій все частіше знаходять застосування технології, які отримали назву "Інтернет речей" (Internet of Things). Сенсори і керовані механізми, що використовуються на полі бою (Internet of Battlefield Things – ІоВТ) стають невід'ємною складовою сучасних військових операцій [1, 2]. Для покращення електромагнітної сумісності різних засобів, зменшення енергоспоживання, масо габаритних показників, затрат на розгортання та експлуатацію обладнання окреме місце у вказаних системах знаходить одностороння радіопередача [3, 4]. Необхідна умова використання односторонніх режимів роботи – забезпечення заданих та взаємопов'язаних показників достовірності доведення інформації, а саме – ймовірності правильного прийому повідомлення (P_n) та ймовірності прийому хибного повідомлення P_x (хибного спрацювання), що зумовлене впливом структурних завад коли корисний сигнал в каналі – відсутній [5]. У зв'язку з цим, важливим завданням є здійснення аналізу достовірності прийому повідомлень за критерієм максимуму значення ймовірності правильного прийому при заданих вимогах до ймовірності хибного спрацювання в існуючих ІоТ технологіях з односторонніми протоколами взаємодії.

Аналіз досліджень. За результатами аналізу досліджень за обраним напрямком можна зробити висновок, що більшість публікацій присвячені розгляду особливостей функціонування ІоТ технологій з односторонніми протоколами взаємодії. В них вказані методи, що застосовуються для підвищення достовірності прийому інформації, серед яких – застосування корегуючих кодів з виправленням помилок, повторів передачі, як на одній частоті, так і на наборові частот [3, 4, 6, 7]. Однак недостатньо дослідженим залишається напрямком, що пов'язаний з оцінкою достовірності прийому повідомлень за критерієм максимуму значення P_n в доволі складній завадовій обстановці, з урахуванням вказаних вище методів підвищення достовірності та впливу структурних завад.

Метою статті є здійснення аналізу достовірності прийому повідомлень в ІоТ технологіях з односторонніми протоколами взаємодії для визначення перспектив їх використання при проведенні сучасних бойових дій в умовах доволі складної завадової обстановки та впливу структурних завад.

Виклад основного матеріалу. В якості прикладів систем з односторонньою радіопередачею розглянуто технології Internet of Things (ІоТ) з односторонніми протоколами взаємодії (технологія Sigfox та Weightless-N).

Технологія Sigfox [4] підтримує як односторонню, так і двосторонню радіопередачу. Для підвищення достовірності прийому кадрів використовуються повтори передачі на різних частотах. Передача першої копії кадру висхідною лінією здійснюється випадково вибраним частотним каналом, а після цього передається дві наступні копії через різні інтервали часу, використовуючи інші частотні канали. Завадостійкі коди з виправленням помилок не застосовуються [7].

Визначений в технології Sigfox формат кадру для висхідної лінії показано на рисунку 1 [4].

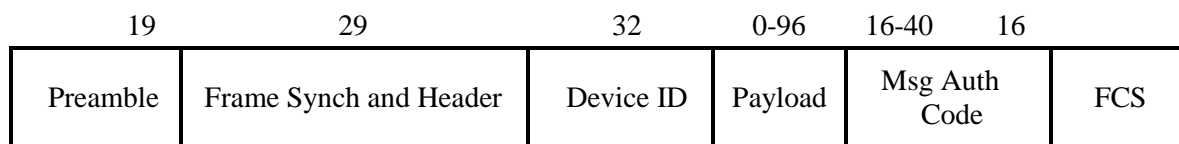


Рисунок 1. Формати кадрів Sigfox у висхідній лінії

Всі розміри поля кадру вказані в бітах. Третє поле після заголовка кадру є ідентифікатор пристрою (ID пристрою). Передостаннє поле кадру – це код автентифікації повідомлення (Msg Auth Code), останнє поле – контрольна послідовність, що призначена для виявлення помилок в кадрі (FCS).

Технологія Weightless-N [3, 8] повністю базується на односторонній радіопередачі висхідною лінією. Всі пристрої відправляють повідомлення на центральну базову станцію без синхронізації та підтвердження. Виділений діапазон частот розподілений на шість широких смуг (таблиця 1). Кожна смуга призначена для окремої базової станції. Кінцеві пристрої працюють в вузькій смузі частот 200 Гц (мікроканал).

Таблиця 1. Смуги частот технології Weightless-N

Band №	Lower Band (MHz)	Upper Band (MHz)	Bandwidth (MHz)	Number of Channels
1	863	864.998	1.998	9990
2	865	868	3	15000
3	868	868.6	0.6	3000
4	868.7	869.2	0.5	2499
5	869.4	869.64	0.24	1200
6	869.7	870	0.3	1500

Окрім того, кожна широкосмугова мережа ділиться на три під смуги (макроканали), кожен з яких містить декілька мікроканалів. Наприклад, кожен макроканал в діапазоні 0.6 МГц містить 1000 каналів.

Щоб підвищити ймовірність правильного прийому повідомлення, кожен термінал відправляє три ідентичні копії кожного повідомлення різними макро та мікро каналами, використовуючи режим стрибкоподібного переналаштування частоти. Загальна кількість копій може бути збільшена до 8-ми.

В технології Weightless-N кожне повідомлення складається з семи блоків (рисунку 2) [8]: преамбули (Preamble), ідентифікаційного номера вузла (Identification number - ID), блоку "Data length", де вказується розрядність блоку Payload (корисне навантаження), мітки часу (Time stamp), код автентифікації повідомлення (Message Authentication Code – MAC) та FCS (Frame-Check Sequence) – кодова послідовність для виявлення помилок.

Preamble	ID	Data length	Time stamp	Payload	MAC	FCS
3 bytes	6 bytes	5 bits	19 bits	0 – 20 bytes	24 bits	16 bits

Рисунок 2. Повідомлення в технології Weightless-N

Базова станція перевіряє FCS і в тому випадку коли виникає помилка повідомлення ігнорується. В іншому випадку базова станція перевіряє відмітку часу (Time stamp), яка є лічильником хвилин у внутрішньому таймері терміналу. Передбачено, що повідомлення з однаковою відміткою часу є копіями одного і того ж повідомлення. В системі використовується згортковий код, що дозволяє виправляти помилки [6].

Оцінка достовірності прийому інформації в IoT технологіях з односторонніми протоколами взаємодії здійснюється шляхом розрахунку значення ймовірності правильного прийому повідомлення та ймовірності хибного спрацювання в продовж усього часу роботи системи. Для здійснення розрахунків ймовірнісних характеристик достовірності застосовані наступні аналітичні співвідношення. Ймовірність правильного прийому повідомлення з першої спроби описується виразом [5]:

$$P_{n1} = \sum_{i=0}^N C_N^i \cdot p_{\text{бит}}^i (1 - p_{\text{бит}})^{N-i}, \quad (1)$$

де N – загальна довжина повідомлення, $p_{\text{бит}}$ – ймовірність помилки на біт.

Для визначення достовірності прийому повідомлень при використанні багатократної передачі ймовірність правильного прийому повідомлення хоча б один раз за R спроб записується наступним чином [5]:

$$P_{nR} = \sum_{i=1}^R C_R^i \cdot P_{n1}^i (1 - P_{n1})^{R-i}. \quad (2)$$

Для розрахунку ймовірності правильного прийому повідомлення хоча б на одній придатній частоті хоча б один раз за умови, що ймовірність придатності кожної з частот – однакова визначається наступним співвідношенням [5]:

$$P_n = \sum_{i=1}^F C_F^i \cdot P_{nR}^i (1 - P_{nR})^{F-i}, \quad (3)$$

де F – загальна кількість частот для передачі повідомлення.

Виходячи з умов функціонування систем радіозв'язку різного призначення, варто враховувати, що в процесі очікування повідомлення може виникнути ситуація, коли канал відсутній, а через вплив завади, яка складається з послідовності статистично незалежних рівноймовірних символів приймається якесь хибне повідомлення. У цьому випадку $p_{\text{бит}} = 0,5$ [5].

Тоді ймовірність хибного прийому повідомлення з першої спроби при впливові структурних завод різного походження задається наступним співвідношенням [5]:

$$P_{x1} = \sum_{i=0}^N C_N^i \cdot (p_{\text{бит}})^i (1 - p_{\text{бит}})^{N-i} = 2^{-N} \sum_{n=0}^N C_N^i \quad (4)$$

Відомо, що коли, наприклад, час роботи системи радіозв'язку в режимі радіомовчання дорівнює ΔT , то атакуючій стороні досить нескладно підібрати таке значення розрядності регістра зсуву n (розрядність регістра для формування послідовності максимальної довжини), за допомогою якого можна буде сформувати імітуючу послідовність загальної довжини $L = 2^n - 1$, що не повторюватиметься за структурою впродовж часу ΔT на довжині n при будь-якому місці його розташування [5]. Ймовірність прийому хибного повідомлення хоча б один раз за час ΔT , при швидкості передачі V розраховується за наступним аналітичним виразом [5]:

$$P_{xL} = \left(\sum_{i=1}^{L-N+1} C_{L-N+1}^i \cdot P_{x1}^i (1 - P_{x1})^{L-N+1-i} \right), \quad (5)$$

де $L = \Delta T \cdot V$.

Ймовірність того, що хибне спрацювання відбудеться хоча б на одній з частот описується формулою [5]:

$$P_x = \sum_{i=1}^F C_F^i \cdot P_{xL}^i (1 - P_{xL})^{F-i}. \quad (6)$$

Оцінка достовірності прийому повідомлень здійснювалась за наступними вихідними даними.

Вихідні дані.

1. Час роботи системи $\Delta T = 7$ діб.

2. Вимоги по ймовірності правильного прийому повідомлень. Вимоги до ймовірності правильного прийому повідомлення в IoT системах розраховуються відповідно до вимог QoS по максимально допустимому значенні ймовірності помилки на символ, яка при передачі повідомлень C2 (Command & control messages containing critical operational) має бути не більша ніж 10^{-3} [9]. Максимальна довжина повідомлення в Weightless-N системах становить 280 двійкових символів (рисунок 2) [8], а в системі SigFox – 216 (рисунок 1) [4]. Для підвищення достовірності прийому повідомлень в обох випадках здійснюється трикратна передача на різних частотах. Здійснивши елементарні розрахунки можна визначити, що мінімально допустима ймовірність правильного прийому повідомлення в таких системах становить – $P_{\text{п}} \geq 0.99$.

3. Вид модуляції та завадостійкого кодування. В технологіях SigFox (у висхідній лінії) та Weightless-N використовується відносно фазова модуляція (Differential Binary Phase-Shift Keying). Застосовуються завадостійкі коди з виявленням помилок [4, 8]. В Weightless-N є можливість використання коду з виправленням помилок [6]. Вказані параметри враховано при розрахунках значення $p_{\text{вн}}$.

4. Швидкість передачі в радіоканалі та максимально допустимий час на передачу повідомлень. Швидкість передачі в радіоканалі (в країнах Європи) становить 100 біт/с [4, 8].

5. Вимоги по ймовірності хибного прийому повідомлень.

Вимоги до ймовірності хибного прийому повідомлень розраховуються виходячи з того, що за час роботи системи ΔT не повинно виникати хибних спрацювань [10]. Іншими словами – середній час між двома хибними спрацюваннями T_c має не перевищувати значення ΔT $T_c \leq \Delta T$. Зв'язок між ймовірністю хибного спрацювання і T_c є наступним [10]. При відсутності сигналу ймовірність помилкового прийому символу $p_c = 0.5$, тоді при швидкості передачі V біт/с заповнення регістра зсуву в приймачеві буде відбуватись $0.5 \cdot V$ раз за секунду. Після кожного наступного заповнення хибне спрацювання можливе з ймовірністю – P_x . Тоді за одну секунду може бути здійснено β хибних заповнень регістру, де $\beta = 0.5 \cdot V \cdot P_x$. Розглядаючи ці події, як рідкісні, можна скористатись розподілом Пуассона з параметром β . Тоді величина, зворотна β , буде визначати середній час між двома хибними спрацюваннями $T_c = 2/V \cdot P_x$. Вимоги до ймовірності хибного спрацювання можна записати наступним чином [10]:

$$P_x \leq \frac{2}{V \cdot \Delta T}, \quad (7)$$

В таблиці 2 наведено вимоги до ймовірності хибного спрацювання при різному часі роботи системи та швидкості передачі 100 біт/с.

Таблиця 2. Гранично допустимі значення ймовірності хибного спрацювання

Час ΔT	Вимоги до ймовірності хибного спрацювання (P_x), при швидкості передачі 100 біт/с.
1 година	$5.56 \cdot 10^{-6}$
1 доба	$2.3 \cdot 10^{-7}$
1 тиждень	$3.3 \cdot 10^{-8}$
1 місяць	$7.7 \cdot 10^{-9}$

7. Максимально допустима кількість частот для передачі повідомлень в технології Weightless-N, $F^* = 8$ [8], в технології SigFox $F^* = 3$ [4].

Для проведення досліджень було використано аналітичні вирази (1 – 6). За допомогою яких здійснено розрахунки достовірності прийому повідомлень для наведених вище вихідних даних. В результаті розрахунків встановлено, що значення ймовірності хибного спрацювання при будь-якому часі очікування повідомлень для систем, що аналізуються є значно меншою ніж та яка наведена в

таблиці 2. На рисунку 3 представлено залежність приросту значення ймовірності правильного прийому повідомлення від завадової обстановки (значення $P_{\text{бит}}$) при трикратній передачі.

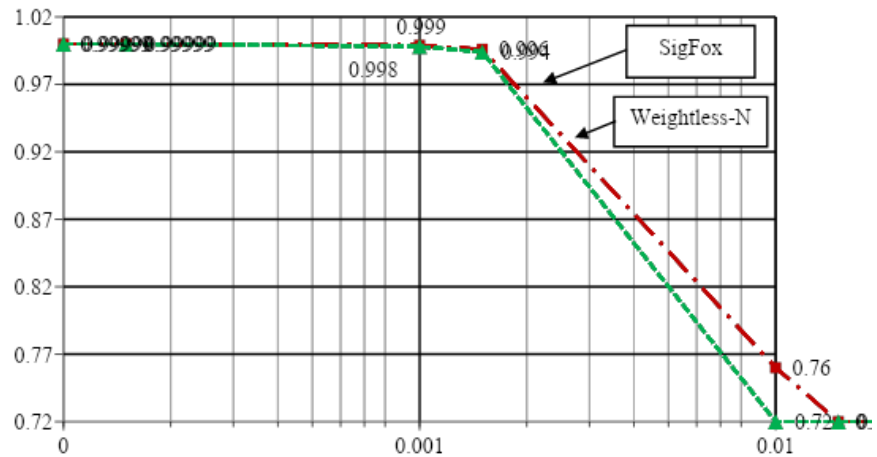


Рисунок 3. Залежність приросту значення P_n від значення $p_{\text{бит}}$ при трикратній передачі

На рисунку 4 наведено залежність приросту значення ймовірності правильного прийому повідомлення від значення $p_{\text{бит}}$ при використанні технології Weightless-N в якій передбачена можливість восьмикратної передачі.

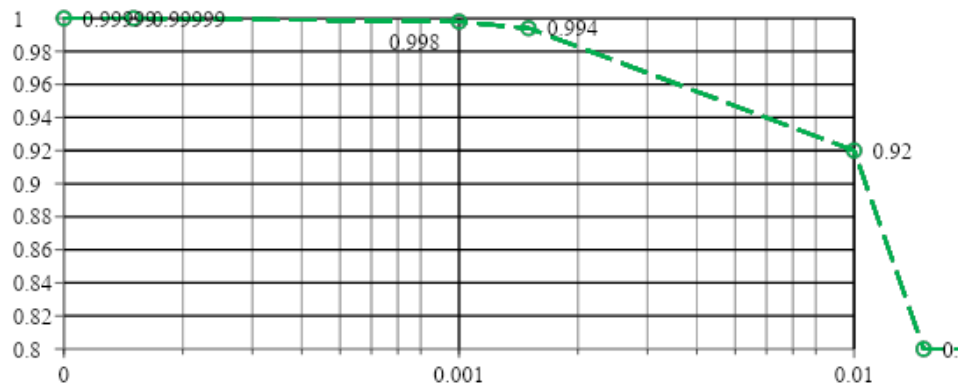


Рисунок 4. Залежність приросту значення P_n від значення $p_{\text{бит}}$ при восьмикратній передачі повідомлень в технології Weightless-N

Аналіз отриманих результатів свідчить про наступне:

– при трикратній передачі та відношенні сигнал-шум в каналі такому, що $P_{\text{бит}}$ знаходиться в межах від $1,5 \cdot 10^{-3}$ до 10^{-2} ймовірність правильного прийому повідомлень при застосуванні технології SigFox знаходиться в межах від 0,99 до 0,76, технології Weightless-N – від 0,99 до 0,72, ;

– при восьмикратній передачі та відношенні сигнал-шум в каналі такому, що $P_{\text{бит}}$ знаходиться в межах від $1,5 \cdot 10^{-3}$ до 10^{-2} ймовірність правильного прийому повідомлень при застосуванні технології Weightless-N знаходиться в межах від 0,99 до 0,92. При подальшому погіршенні завадової обстановки ймовірність правильного прийому повідомлень стрімко зменшується і при $P_{\text{бит}} = 0,015$ становить 0,8.

Висновки та перспективи подальшого дослідження.

В результаті досліджень – встановлено, що в розглянутих IoT технологіях, коли ймовірність помилкового прийому символу повідомлення не перевищує 10^{-2} , ймовірність правильного прийому повідомлень є не меншою ніж 0,99. При подальшому погіршенні завадової обстановки не

забезпечується прийом повідомлень з необхідною достовірністю. Ймовірність хибного спрацювання для розглянутих в роботі систем є значно меншою ніж максимально допустима навіть при тривалому часі роботи системи.

Таким чином існуюча структура повідомлень, застосовані методи прийому та обробки сигналів не дозволяють забезпечити задану достовірність прийому інформації в умовах складної заводської обстановки. Тому застосування технологій інтернету речей з односторонніми протоколами взаємодії при проведенні сучасних бойових дій, що супроводжуються великою насиченістю радіоефіру та застосуванням засобів радіоелектронної боротьби потребує розв'язання завдань із підвищення достовірності прийому інформації. У зв'язку з цим, напрямком подальших досліджень є розробка структури повідомлень, визначення способів прийому та обробки сигналів для підвищення достовірності прийому інформації в IoT технологіях з односторонніми протоколами взаємодії.

Список бібліографічного опису

1. Russell S., Abdelzaher T. The Internet of Battlefield Things: The Next Generation of Command, Control, Communications and Intelligence (C3I) Decision-Making. *MILCOM 2018 – 2018 IEEE Military Communications Conference (MILCOM)*, 29 – 31 Oct. 2018. Los Angeles, CA, USA. Los Angeles. 2018. P. 737 – 742. URL: <https://ieeexplore.ieee.org/document/8599853>.
2. Analyzing the Applicability of Internet of Things to the Battlefield Environment / N. Suri, M. Tortonesi, J. Michaelis, P. Budulas, G. Benincasa, S. Russell, C. Stefanelli, R. Winkler. *ICMCIS 2016: Conference Paper At Brussels, Belgium*, May 2016. DOI: 10.1109/ICMCIS.2016.7496574. URL: https://www.researchgate.net/publication/303839381_Analyzing_the_Applicability_of_Internet_of_Things_to_the_Battlefield_Environment.
3. MAC Layer Protocols for Internet of Things. Luiz Oliveira, Joel J. P. C. Rodrigues, Sergei A. Kozlov, Ricardo A. L. Rabêlo, Victor Hugo C. de Albuquerque. *Journal Future Internet*. 2019. Vol. 11(1). P. 16. URL: https://www.researchgate.net/publication/330381472_MAC_Layer_Protocols_for_Internet_of_Things_A_Survey/fulltext/5c3d680992851c22a375d03d/MAC-Layer-Protocols-for-Internet-of-Things-A-Survey.pdf.
4. A Sigfox Energy Consumption Model. Carles Gomez, Juan Carlos Veras, Rafael Vidal, Lluís Casals. *Journal Sensors*. 2019. Vol. 19. P. 681. URL: https://www.researchgate.net/publication/330947889_A_Sigfox_Energy_Consumption_Model/fulltext/5c5ced9d45851582c3d5a09e/A-Sigfox-Energy-Consumption-Model.pdf.
5. Ерохин В. Ф. Методика расчета длины кодограммы для асимптотически надежной радиолинии управления / Ерохин В. Ф., Залужный О. В. // *Вісник Національного технічного університету України „КПІ”, серія – Радіотехніка. Радіоапаратобудування*. – 2013. – № 54. – С. 44 – 53.
6. Bembe, M., Abu-Mahfouz, A., Masonta, M. A survey on low-power wide area networks for IoT applications. *Telecommun Syst* 71, 249–274 (2019). <https://doi.org/10.1007/s11235-019-00557-9>.
7. B. Buurman, J. Kamruzzaman, G. Karmakar and S. Islam. Low-Power Wide-Area Networks: Design Goals, Architecture, Suitability to Use Cases and Research Challenges. in *IEEE Access*, vol. 8, pp. 17179-17220, 2020, DOI: 10.1109/ACCESS.2020.2968057.
8. Abbas R., Al-Sherbaz A., Bennecer A., Picton P. A new channel selection algorithm for the Weightless-N Frequency Hopping with lower collision probability. *8th International Network of the Future (NoF) Conference Proceedings*. London: IEEE (In Press). 2017. URL: <http://nectar.northampton.ac.uk/id/eprint/9777>.
9. Investigating communication architecture for tactical radio networks design / B. Suman, S.C. Sharma, M. Pant, S. Kumar. *International Journal of Research in Engineering & Applied Sciences*. 2012. Vol. 2, Issue 2. P. 106 – 118. URL: <http://euroasiapub.org/wp-content/uploads/2016/10/11.pdf>.
10. Защищенные радиосистемы цифровой передачи информации / П.Н. Сердюков и др. Москва: АСТ, 2006. 403 с.

References

1. Russell S., Abdelzaher T. The Internet of Battlefield Things: The Next Generation of Command, Control, Communications and Intelligence (C3I) Decision-Making. *MILCOM 2018 – 2018 IEEE Military Communications Conference (MILCOM)*, 29 – 31 Oct. 2018. Los Angeles, CA, USA. Los Angeles. 2018. P. 737 – 742. URL: <https://ieeexplore.ieee.org/document/8599853>.
2. Analyzing the Applicability of Internet of Things to the Battlefield Environment / N. Suri, M. Tortonesi, J. Michaelis, P. Budulas, G. Benincasa, S. Russell, C. Stefanelli, R. Winkler. *ICMCIS 2016: Conference Paper At Brussels, Belgium*, May 2016. DOI: 10.1109/ICMCIS.2016.7496574. URL: https://www.researchgate.net/publication/303839381_Analyzing_the_Applicability_of_Internet_of_Things_to_the_Battlefield_Environment.
3. MAC Layer Protocols for Internet of Things. Luiz Oliveira, Joel J. P. C. Rodrigues, Sergei A. Kozlov, Ricardo A. L. Rabêlo, Victor Hugo C. de Albuquerque. *Journal Future Internet*. 2019. Vol. 11(1). P. 16. URL: https://www.researchgate.net/publication/330381472_MAC_Layer_Protocols_for_Internet_of_Things_A_Survey/fulltext/5c3d680992851c22a375d03d/MAC-Layer-Protocols-for-Internet-of-Things-A-Survey.pdf.
4. A Sigfox Energy Consumption Model. Carles Gomez, Juan Carlos Veras, Rafael Vidal, Lluís Casals. *Journal Sensors*. 2019. Vol. 19. P. 681. URL: https://www.researchgate.net/publication/330947889_A_Sigfox_Energy_Consumption_Model/fulltext/5c5ced9d45851582c3d5a09e/A-Sigfox-Energy-Consumption-Model.pdf.

5. Yerokhin V. F., Zaluzhnyi O. V. Calculating technique of the codegram length for asymptotically reliable radio lines of control / Yerokhin V. F., Zaluzhnyi O. V. // *Visnyk NTUU KPI Seriia - Radiotekhnika Radioaparotobuduvannia*. – 2013. - № 54. – P. 44 – 53.
6. Bembe, M., Abu-Mahfouz, A., Masonta, M. A survey on low-power wide area networks for IoT applications. *Telecommun Syst* 71, 249–274 (2019). <https://doi.org/10.1007/s11235-019-00557-9>.
7. B. Buurman, J. Kamruzzaman, G. Karmakar and S. Islam. Low-Power Wide-Area Networks: Design Goals, Architecture, Suitability to Use Cases and Research Challenges. in *IEEE Access*, vol. 8, pp. 17179-17220, 2020, DOI: 10.1109/ACCESS.2020.2968057.
8. Abbas R., Al-Sherbaz A., Bennecer A., Picton P. A new channel selection algorithm for the Weightless-N Frequency Hopping with lower collision probability. *8th International Network of the Future (NoF) Conference Proceedings*. London: IEEE (In Press). 2017. URL: <http://nectar.northampton.ac.uk/id/eprint/9777>.
9. Investigating communication architecture for tactical radio networks design / B. Suman, S.C. Sharma, M. Pant, S. Kumar. *International Journal of Research in Engineering & Applied Sciences*. 2012. Vol. 2, Issue 2. P. 106 – 118. URL: <http://euroasiapub.org/wp-content/uploads/2016/10/11.pdf>.
10. Protected radio systems for digital transmission of information / P.N. Serdyukov et al. Moscow: AST, 2006. 403 p.

CONTENTS

AUTOMATION AND MANAGEMENT	
Kolpakov P.S., Marchenko O.I. An optimization technique for transferring of identical JSON documents over computer networks.	5
Tohoiev O.R. A method for deanonymizing iOS users through the Airdrop protocol.	12
INFORMATICS AND COMPUTER SCIENCE	
Bahniuk N.V., Bortnyk K.Y., Varchenko L.L. Monitoring system with Windows operating system.	18
Gavrilchenko O.V., Korendii V.M., Shenbor V.S The influence of the parameters of the vibrating conveyor on speed of transportation of parts.	24
Yermolenko D., Marchenko O. Algorithms of the technique for comparing programs written in Lisp-like languages based on abstract semantic trees.	29
Kozubtsov I., Silko O., Kozubtsova L., Lukashenka V. Indicators and criteria for assessing the level of academic integrity of applicants for higher education.	38
Levitskaya T.O., Herasymov D.S. Forecasting time series of financial indicators using a multilayer neural network.	48
Nikolaievskiy O. Business information technologies as the main aspect of the competitive advantage of modern business.	54
Poplavska G., Sachuk Yu. Methodical aspects of teaching students to develop an application for testing students' knowledge using Lazarus.	63
Svyrydiuk K.A., Khrystynets M.O., Chernyashchuk N.L., Khrystynets N.A. Determining the impact of ergonomics and search engine optimization on site ranking factors.	69
Feshchenko I.O., Dadyverin V.V., Potapova K.R., Nalyvaichuk M.V. Adaptation of supersampling technology to increase the performance of real-time ray tracing image rendering.	74

TELECOMMUNICATIONS AND RADIO ENGINEERING

Burban O., Polishchuk M., Bortnyk K.Ya. System of voice control lighting based on Wemos D1 Mini.	80
Polishchuk M., Savaryn P., Furkalo S. Webhooks and Long Polling methods for hosting a Telegram bot.	86
Radzivilov H. The reliability analysis of receiving messages in IoT technologies with one-way interaction protocols.	93

ВИМОГИ ДО СТРУКТУРИ ТА ОФОРМЛЕННЯ МАТЕРІАЛУ СТАТЕЙ

➤ **Наукова стаття обов'язково повинна мати наступні необхідні елементи:**

- 1) **постановка проблеми** у загальному вигляді та її зв'язок із важливими науковими чи практичними завданнями;
 - 2) **аналіз останніх досліджень і публікацій**, в яких започатковано розв'язання даної проблеми і на які спирається автор,
 - 3) **виділення невіршених раніше частин загальної проблеми**, котрим присвячується означена стаття;
 - 4) **формулювання мети дослідження** (постановка завдання);
 - 5) **виклад основного матеріалу дослідження** з повним обґрунтуванням отриманих наукових результатів; **висновки** з даного дослідження, у тому числі з науковою новизною і
 - 6) **перспективи подальших досліджень** у даному напрямку.
- Статтю можна подавати українською, російською або англійською мовами. Вона повинна бути набрана у текстовому редакторі MS WORD 03/07/10 і надрукована на лазерному або струменевому принтері на білих листах формату А4 (297×210 мм). **Нумерацію сторінок** не виконувати. **Обсяг статті** 5-10 сторінок (не менше).
 - **Параметри сторінки**. Верхнє, нижнє та праве поле –1,5 см, лівє – 2 см. Від краю до верхнього колонтитула – 1,25 см, нижнього – 1,25 см.
 - **Шапка статті**. УДК, ORCID (якщо є), автори (ім'я та прізвище повністю), місце роботи кожного автора. Назва організації та назва статті набираються з нового рядка шрифтом Time New Roman Суг розміром 11 пт з одинарним міжрядковим інтервалом та вирівнюються по лівому краю. Назва статті розміщується через один рядок нижче назви організації (розмір шрифту 11 пт з напівжирним виділенням та вирівнюванням по центру).
 - **Анотації** (українською, російською та англійською мовами) повинні містити прізвища та ініціали авторів, назву статті та короткий її зміст і розміщуються через один рядок нижче назви статті та набираються з абзацного відступу 1 см шрифтом Time New Roman Суг розміром 9 пт з одинарним міжрядковим інтервалом і вирівнюються по ширині. Нижче анотацій обов'язково вказуються **ключові слова**.
 - **Основний текст** розміщується на через один рядок нижче анотацій, набирається з абзацного відступу 1 см шрифтом Time New Roman розміром 11 пт з одинарним міжрядковим інтервалом та вирівнюється по ширині.
 - **Формули** набираються у редакторі формул MS WORD (використовувати шрифти: Symbol, Time New Roman Суг; розміри шрифтів: звичайний 12 пт, крупний індекс 7 пт, дрібний індекс 5 пт, крупний символ 18 пт, дрібний символ 12 пт). Формула вирівнюється по центру і не повинна займати більше 5/6 ширини рядка.
 - **Ілюстрації**, що присутні у статті, необхідно розташовувати у тексті по центру, вирівнюючи підписи по центру (Рис. 1. Назва). Другий екземпляр ілюстрації необхідно подати на окремому листі. Ілюстрації повинні бути чіткими та контрастними.
 - **Таблиці** потрібно розташовувати у тексті по центру, причому їх ширина повинна бути на 1 см менша ширини рядка. Над таблицею ставиться її порядковий номер і назва (Таблиця 1. Назва) та вирівнюється по центру.
 - **Посилання** на ту чи іншу роботу повинні позначатися в тексті у квадратних дужках за порядковим номером у списку літератури в кінці статті; посилання на джерела статистичних даних обов'язкові; посилання на публікації дослідників обов'язкові; посилання на підручники, навчальні посібники, газети і ненаукові журнали – небажані; посилання на власні публікації допускаються тільки у випадку крайньої необхідності; роботи авторів, на прізвища яких є посилання в тексті, мають бути в списку літератури до цієї статті.
 - **Список бібліографічного опису та References**. Список літератури («References») потрібно приводити повністю окремим блоком, повторюючи список літератури, який подається українською / російською мовою, незалежно від того, є в ньому іноземні джерела чи ні. Тобто, після статті подається 2 списки: «Список бібліографічного опису» (звичайний список літератури) і «References» (список для міжнародних БД). Необхідно в опис джерела вносити всіх авторів, не скорочуючи їх до трьох, як це рекомендовано діючими у нас державними стандартами. References - повинен бути укладений англійською мовою або транслітерований. Оформлювати згідно з одним із найбільш уживаних у світі стандартів: APA – American Psychological Association; CBE – Council of Biology Editors, Citation-Sequence; Chicago (Author-Date System); Harvard; Harvard – British Standard; MLA (Modern Language Association) – Single Spaced Reference List; NLM – National Library of Medicine; Uniform Requirements for Manuscripts Submitted to Biomedical Journals. У жодному з перелічених стандартів не використовуються розділові знаки: «//», «-». Назва джерела та вихідні дані відокремлюються від авторів і заголовка статті типом шрифту, найчастіше, курсивом (italics), крапкою або комою. Існує багато безкоштовних програм для створення бібліографічних описів у романській абетці, що дають можливість автоматично створювати посилання за одним із світових стандартів наприклад: <http://www.easybib.com/>, <http://www.bibme.org/>, <http://www.sourceaid.com/>, <https://vak.in.ua/>.
 - **Рецензування статей**. Просимо вказати двох рецензентів (ПІБ, електронні адреси та звання) для подальшого рецензування Вашої статті. Адміністратор реєструє поданих рецензентів на сайті журналу, тоді на їхню електронну адресу надсилається форма для рецензування. Редакція залишає за собою право направляти статті на додаткову рецензію та відхиляти їх в разі відсутності рецензій.
 - Стаття обов'язково подається на електронну адресу: cit@lntu.edu.ua.
 - Рукописи, що не відповідають вище вказаним вимогам, не розглядаються і до друку не приймаються.
 - **Усі рукописи проходять перевірку на плагіат!**

ЗРАЗОК ОФОРМЛЕННЯ СТАТТІ
(поля верхнє, нижнє -1.5 см, ліве та праве 2см. дзеркальні поля)

ЗРАЗОК

УДК 621.391

Мороз Борис Іванович, д.т.н., професор,

<https://orcid.org/0000-0002-5625-0864>

Антіпов Олександр Андрійович, аспірант,

Журавльов Володимир Сергійович, аспірант.

<https://orcid.org/0000-0002-7366-9552>

Національний технічний університет «Дніпровська політехніка», м. Дніпро, Україна.

АВТОМАТИЗОВАНА СИСТЕМА ДОСТАВКИ МЕДИКАМЕНТІВ ЗА ДОПОМОГОЮ БЕЗПІЛОТНИХ ЛІТАЛЬНИХ АПАРАТІВ (МУЛЬТИКОПТЕРІВ) ЗА ЗАПИТОМ СПОЖИВАЧА

Мороз Б. І., Антіпов О.А., Журавльов В. С. Автоматизована система доставки медик... **Times New Roman 9**
безпілотних літальних апаратів (мультикоптерів) за запитом споживача. Представлено концепт системи доставки медикаментів за допомогою безпілотних літальних апаратів. Запропоновано архітектуру системи автоматичної диспетчеризації замовлень від споживача, зберігання замовлень, та планування доставки дронами. Також було розглянуто юридичні обмеження роботи запропонованої системи.

Ключові слова: мультикоптер, дрон, доставка, клієнт-серверна архітектура, RSA, APM, HTTPS, Mission Planner.

Мороз Б. И., Антипов А.А., Журавлев В. С. Автоматизированная система доставки медикаментов с помощью беспилотных летательных аппаратов (мультикоптеров) по запросу потребителя. Представлен концепт системы доставки медикаментов с помощью беспилотных летательных аппаратов. Предложена архитектура системы автоматической диспетчеризации заказов от потребителя, хранения заказов, и планирование доставки дронами. Также были рассмотрены юридические ограничения работы предложенной системы.

Ключевые слова: мультикоптер, дрон, доставка, клиент-серверная архитектура, RSA, APM, HTTPS, Mission Planner.

Moroz B., Antipov A., Zhuravlev V. Automated system for the delivery of medical supplies using unmanned aerial vehicles (multicopter) at the request of the consumer. The concept of medical supplies delivery system using unmanned aerial vehicles is presented. The architecture of the system of automatic dispatching orders from the consumer, storage of orders, and scheduling delivery by drones are proposed. The legal limitations of the proposed system were also considered.

Keywords: multicopter, drone, delivery, client-server architecture, RSA, APM, HTTPS, Mission Planner.

Постановка наукової проблеми.

.....

Аналіз досліджень.

.....

Виклад основного матеріалу й обґрунтування отриманих результатів дослідження.

.....

Висновки та перспективи подальшого дослідження.

.....

Список бібліографічного опису

1. Сін Лю, Ціньян Сяо, Віджай Гопалакришнан, Маттео Варвелло (2017) Досліджен... **Times New Roman 9**
панорамного відеопотоку, С. 50-55. АСМ.
2. Б. Хань, Ф., Цянь, Л. Джі та В. Гопалакришнан. (2017) MP-DASH: Адаптивна відео-трансляція через перевагу, орієнтовану на багатфункціональність. У матеріалах 12-ї Міжнародної конференції з нових мережевих експериментів та технологій, С. 129-143. АСМ.

References

1. Xing Liu, Qingyang Xiao, Vijay Gopalakrishnan, Matteo Varvello (2017) Research 360° Innovations for Panoramic Video Streaming, P. 50-55. ACM.
2. Han, B., Qian, F., Ji, L. & Gopalakrishnan, V. (2017) MP-DASH: Adaptive Video Streaming Over Preference-Aware Multipath. In Proceedings of the 12th International on Conference on emerging Networking Experiments and Technologies, P. 129-143. ACM.

Довідки з питань публікації та прийому матеріалів у науковий журнал «Комп'ютерно-інтегровані технології: освіта, наука, виробництво» можна отримати у відповідального секретаря – Свириднюк Катерини Анатоліївни за тел. (0332) 74-61-15, або (063)-940-69-42.

Адреса: 43018, м. Луцьк, вул. Львівська, 75, ауд. 141

Автор статті отримує 1 примірник збірника у форматі *pdf.

Вартість однієї сторінки становить 40 – грн. (для працівників Луцького НТУ), 50 грн – для інших ЗВО.

Окремо, кожній статті, буде присвоєний DOI (digital object identifier) - ідентифікатор цифрового об'єкту, що веде за собою додаткову оплату 60 грн.(ціна з 2021 року).

**Реквізити для зарахування коштів
(за публікацію наукових праць, участь у наукових заходах)**

ЛУЦЬКИЙ НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ

код в ЄДРПОУ 05477296

UA16 820172 0 3132 5 1 002 3 02 017820

в ДКСУ у м. Київ МФО 820172

Україна, Волинська область,

43018, м. Луцьк, вул. Львівська, 75

Міністерство освіти і науки України
Луцький національний технічний університет

Колектив авторів

КОМП'ЮТЕРНО-ІНТЕГРОВАНІ ТЕХНОЛОГІЇ: ОСВІТА, НАУКА, ВИРОБНИЦТВО

Науковий журнал

Підп. до друку 14.12.2022. Формат А4. Папір офс.
Гарн. Таймс. Ум. друк. арк. 15.25 Обл. – вид. арк. 15.75
Тираж 20 прим. Зам. № 14/22

Комп'ютерний набір та верстка:

Н.А. Христинець

Друк Відділ іміджу та промоції ЛНТУ
Свідоцтво Держкомтелерадіо України ДК №4123 від 28.07.2011 р.
43018, м. Луцьк, вул. Львівська, 75, тел.: (0332) 74-61-02
e-mail: rvv_intu@ukr.net