

**НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ  
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ ІМЕНІ ІГОРЯ  
СІКОРСЬКОГО»**

Навчально-науковий інститут атомної та теплової енергетики

Кафедра цифрових технологій в енергетиці

"На правах рукопису"  
УДК \_\_\_\_\_

«До захисту допущено»

Завідувач кафедри

Наталія АУШЕВА

“ \_\_\_\_\_ ” \_\_\_\_\_ 2026 р.

**Магістерська дисертація**

на здобуття ступеня другого (магістерського) рівня вищої освіти  
за освітньою програмою “Комп’ютерні науки”  
зі спеціальності 122 “Комп’ютерні науки”

на тему **Реалізація керованого прийняття рішень у гейміфікованому навчальному середовищі**

Виконав (-ла): студент (-ка) 2 курсу, групи ТР-41мн

ШПАДКІВСЬКИЙ Олександр В’ячеславович

(прізвище, ім’я, по батькові)

\_\_\_\_\_ (підпис)

Науковий керівник доцент, к.т.н. ТИХОХОД В.О.

(посада, науковий ступінь, вчене звання, прізвище та ініціали)

\_\_\_\_\_ (підпис)

Рецензент доц. каф. ПЗЕ, к.т.н., доц. ГАГАРІН О.О.

(посада, науковий ступінь, вчене звання, прізвище та ініціали)

\_\_\_\_\_ (підпис)

Н.контроль провідний інженер Людмила КУЗЬМІНА

\_\_\_\_\_ (підпис)

Засвідчую, що у цій магістерській дисертації  
немає запозичень з праць інших авторів без  
відповідних посилань.

Студент (-ка) \_\_\_\_\_  
(підпис)

**НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ  
“КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ  
імені ІГОРЯ СІКОРСЬКОГО”**

НАВЧАЛЬНО-НАУКОВИЙ ІНСТИТУТ АТОМНОЇ ТА ТЕПЛОВОЇ ЕНЕРГЕТИКИ

Кафедра ЦИФРОВИХ ТЕХНОЛОГІЙ В ЕНЕРГЕТИЦІ

Рівень вищої освіти – другий (магістерський)

спеціальність 122 “Комп’ютерні науки”

Освітньо-наукова програма “Комп’ютерні науки”

ЗАТВЕРДЖУЮ

Завідувач кафедри ЦТЕ

Наталія АУШЕВА

«\_\_» \_\_\_\_\_ 2026 р.

**ЗАВДАННЯ  
на магістерську дисертацію студенту**

Шпадківському Олександрові В’ячеславовичу

(прізвище, ім’я, по батькові)

1. Тема роботи Реалізація керованого прийняття рішень у  
гейміфікованому навчальному середовищі

керівник роботи Тихоход Володимир Олександрович, к.т.н., доцент

(прізвище, ім’я, по батькові, науковий ступінь, вчене звання)

затверджені наказом по університету від « 30\_березня 2026 р. № 1326-с

2. Термін подання студентом дисертації 11.05.2026

3. Об’єкт дослідження : процес формувального аналізу знань студентів на основі  
текстових відповідей у цифровому навчальному середовищі

4. Вихідні дані: Теоретичні засади концепції «освіта як сервіс»; наукові публікації  
та нормативні документи щодо організації формувального оцінювання у закладах  
вищої освіти; існуючі теоретичні підходи до інтелектуального аналізу текстових  
даних та проектування інформаційних систем підтримки прийняття рішень.

5. Перелік завдань: Проаналізувати особливості цифрових освітніх середовищ і  
вимоги до систем формувального аналізу, дослідити загальні методи обробки  
тексту та обґрунтувати їх використання, розробити модель представлення і  
порівняння знань на основі метрик відповідності, розробити алгоритм керованого

прийняття рішень, а також спроектувати та реалізувати програмну систему з подальшою експериментальною перевіркою ефективності запропонованого підходу.

6. Орієнтований перелік ілюстративного матеріалу : діаграми, що демонструють архітектуру системи та технологічний стек, роботу алгоритмів лінгвістичної обробки тексту, взаємодію компонентів та послідовність передачі даних; приклади інтерфейсів користувача та сценарії роботи з програмним продуктом.

7. Орієнтований перелік публікацій : тези доповідей на Міжнародній науково-практичній конференції, та стаття у фаховому науковому виданні України (категорія «Б»).

8. Дата видачі завдання «15» вересня 2025 р.

### КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів підготовки магістерської дисертації	Термін виконання	Примітка
1	Аналіз методів дослідження за літературними джерелами	17-20.02.26	
2	Вибір засобів реалізації методів/моделей	21-25.02.26	
3	Програмна реалізація	25.02-07.03.26	
4	Проведення обчислювальних експериментів	08-12.03.26	
5	Аналіз результатів обчислювальних експериментів	13-15.03.26	
6	Захист створеного програмного забезпечення	25.03.26	
7	Оформлення магістерської дисертації	26.03.26-25.04.26	

Студент

\_\_\_\_\_ (підпис)

**Олександр ШПАДКІВСЬКИЙ**  
(ім'я, ПРІЗВИЩЕ)

Керівник роботи

\_\_\_\_\_ (підпис)

**Володимир ТИХОХОД**  
(ім'я, ПРІЗВИЩЕ)

# РЕФЕРАТ

**Структура та обсяг магістерської дисертації:** Робота складається зі вступу, 4 розділів, висновків, списку використаних джерел та 2 додатків. Загальний обсяг роботи становить 113 сторінок текстового документа, який містить 17 рисунків та 5 таблиць.

**Актуальність теми:** зумовлена переходом вищої освіти до сервісної парадигми та необхідністю впровадження ефективного формувального оцінювання. Існуючі системи аудиторного опитування не забезпечують глибокого когнітивного аналізу текстових відповідей, що призводить до інформаційного перевантаження викладача. Це створює гостру потребу в алгоритмічних інструментах інтерпретації для підтримки керованого прийняття рішень у цифровому гейміфікованому середовищі у режимі, близькому до реального часу.

**Мета роботи:** розроблення моделі керованого прийняття рішень на основі формалізованого аналізу текстових відповідей студентів та її програмної реалізації у вигляді системи, здатної автоматично інтерпретувати результати аналізу та визначати подальші дії навчального процесу.

## **Завдання дослідження:**

- Проаналізувати особливості цифрових освітніх середовищ та визначити вимоги до систем формувального аналізу знань у гейміфікованому контексті.
- Дослідити підходи до обробки текстових відповідей та методи визначення значущості термінів і обґрунтувати їх використання для аналізу коротких відповідей студентів.
- Розробити модель представлення знань студентів у вигляді множини термінів та сформувану еталонну модель знань для подальшого порівняння.
- Розробити метод порівняння множин термінів на основі обчислення показників повноти, точності та узгодженості.
- Розробити модель інтерпретації отриманих показників на основі формалізованих правил.
- Розробити алгоритм керованого прийняття рішень, що визначає подальші дії

навчального процесу на основі результатів аналізу.

- Реалізувати програмну систему, що інтегрує розроблені моделі та алгоритми в єдиний процес обробки даних та підтримки прийняття рішень.
- Провести експериментальну перевірку ефективності запропонованого підходу та проаналізувати отримані результати.

**Об'єкт дослідження:** процес формування аналізу знань студентів на основі текстових відповідей у цифровому навчальному середовищі.

**Предмет дослідження:** методи та моделі формалізованого представлення знань студентів у вигляді множини термінів, їх аналізу та інтерпретації на основі текстових відповідей.

**Методи дослідження:** методи обробки природної мови для токенизації та лематизації текстів; теорія множин для обчислення подібності; методологія Design Science Research; описові статистичні методи.

**Наукова новизна** отриманих результатів полягає у розробленні моделі керованого прийняття рішень, що реалізує алгоритмічний перехід від результатів аналізу текстових відповідей студентів до визначення подальших дій навчального процесу шляхом відображення кількісних показників подібності множин термінів у якісні рівні структури знань на основі формалізованих правил.

**Практичне значення:** створено працездатну програмну систему, яка забезпечує автоматизовану обробку масивів коротких відповідей у реальному часі та формує обґрунтовані рекомендації для адаптації сценарію заняття. Для валідації розробленої системи проведено пілотний практичний експеримент за участю 30 студентів, який показав зростання результативності експериментальної групи на 13,8%.

**Апробація результатів роботи:** основні результати дослідження доповідались на Міжнародній науково-практичній конференції та опубліковані у фаховому виданні.

**Публікації:** 2 наукові праці (1 стаття у фаховому виданні, 1 теза доповідей).

**Ключові слова:** ФОРМУВАЛЬНЕ ОЦІНЮВАННЯ, ХМАРА ТЕРМІНІВ, ОБРОБКА ПРИРОДНОЇ МОВИ, АЛГОРИТМ ВМ25, ТАКСОНОМІЯ SOLO, КЕРОВАНЕ ПРИЙНЯТТЯ РІШЕНЬ, ГЕЙМІФІКАЦІЯ.

# ABSTRACT

**Structure and volume of the master's thesis:** The work consists of an introduction, 4 chapters, a conclusion, a list of references, and 2 appendices. The total volume of the work is 113 pages of a text document, containing 17 figures and 5 tables.

**Relevance of the topic:** s due to the transition of higher education to a service paradigm and the need for effective formative assessment based on open-ended text responses. Existing approaches do not provide a formalized transition from the results of unstructured data analysis to informed decision-making. This creates a critical need for algorithmic interpretation tools to support guided decision-making in a digital gamified environment in real time.

**Purpose of the work:** to develop a guided decision-making model based on the formalized analysis of students' text responses and its software implementation as a system capable of automatically interpreting analysis results and determining subsequent actions in the educational process.

## **Research tasks:**

- analyze the features of digital educational environments and determine the requirements for formative knowledge analysis systems in a gamified context;
- investigate approaches to processing text responses and methods for determining the significance of terms, and justify their use for analyzing short student responses;
- develop a model for representing students' knowledge as a set of terms and form a reference knowledge model for subsequent comparison;
- develop a method for comparing sets of terms based on calculating completeness, precision, and consistency indicators;
- develop a model for interpreting the obtained indicators based on formalized rules;
- develop a guided decision-making algorithm that determines further actions in the educational process based on the analysis results;
- implement a software system that integrates the developed models and algorithms into a single process of data processing and decision support;

- conduct an experimental verification of the proposed approach's effectiveness and analyze the obtained results.

**Object of research:** the process of formative knowledge analysis of students based on text responses in a digital learning environment.

**Subject of research:** methods and models for the formalized representation of students' knowledge as a set of terms, their analysis, and interpretation based on text responses.

**Research methods:** natural language processing methods for text tokenization and lemmatization; set theory for calculating similarity; Design Science Research (DSR) methodology; descriptive statistical methods.

**The scientific novelty of the obtained results** lies in the development of a controlled decision-making model that implements an algorithmic transition from the analysis of students' textual responses to the determination of further actions in the educational process by mapping quantitative indicators of term set similarity into qualitative levels of knowledge structure based on formalized rules.

**Practical significance:** a functional software system has been created that provides automated real-time processing of short response arrays and generates reasoned recommendations for adapting the lesson scenario. For the validation of the developed system, a pilot practical experiment was conducted involving 30 students, which showed a 13.8% increase in the performance of the experimental group.

**Approbation of the results:** the main results of the research were presented at an International scientific and practical conference and published in a professional journal.

**Publications:** 2 scientific papers (1 article in a professional journal, 1 conference abstract).

**Keywords:** FORMATIVE ASSESSMENT, WORD CLOUD, NATURAL LANGUAGE PROCESSING, BM25 ALGORITHM, SOLO TAXONOMY, GUIDED DECISION-MAKING, GAMIFICATION.

## ЗМІСТ

ВСТУП .....	11
1 ПОСТАНОВКА ЗАДАЧІ ТА ОГЛЯД МЕТОДІВ І ЗАСОБІВ СТВОРЕННЯ СИСТЕМИ КЕРОВАНОГО ПРИЙНЯТТЯ РІШЕНЬ .....	16
1.1 Постановка задачі розробки системи керованого прийняття рішень..	16
1.2 Роль гейміфікації та когнітивні моделі оцінювання (таксономія SOLO, рамка ICAP).....	18
1.3 Огляд та порівняльний аналіз сучасних систем аудиторного опитування .....	20
1.4 Обмеження методів лінгвістичного аналізу та візуалізації в сучасних цифрових системах.....	22
Висновки до розділу 1 .....	24
2 МАТЕМАТИЧНЕ ТА АЛГОРИТМІЧНЕ ЗАБЕЗПЕЧЕННЯ СИСТЕМИ АНАЛІЗУ ЗНАНЬ .....	26
2.1 Формалізація процесу формування множини термінів .....	26
2.2 Методологія формування еталонної множини термінів.....	27
2.3 Математичне обґрунтування методу зважування термінів у коротких текстах .....	28
2.4 Математичні метрики порівняння множин та логіка інтерпретації ....	31
2.5 Застосування алгоритму BM25 для нормалізації термінів та формування їх ваги .....	32
2.6 Модель інтерпретації аналітичного сигналу та правила прийняття рішень .....	34
2.7 Узагальнена архітектура аналітичного конвеєра системи.....	37

Висновки до розділу 2 .....	38
3 ПРОЄКТУВАННЯ ТА ПРОГРАМНА РЕАЛІЗАЦІЯ СИСТЕМИ.....	40
3.1 Обґрунтування вибору технологічного стека .....	40
3.1.1 Вибір засобів розробки клієнтської частини (Frontend) .....	41
3.1.2 Вибір засобів розробки серверної частини (Backend).....	41
3.1.3 Вибір підсистеми зберігання даних (Google Sheets як EaaS-база) 42	
3.1.4 Вибір системи аутентифікації та розмежування ролей.....	42
3.2 Теоретичні аспекти клієнтської оптимізації: Віртуальний DOM та алгоритми узгодження .....	43
3.3 Серверна архітектура: Node.js, Event Loop та асинхронність .....	46
3.4 Забезпечення безпеки та криптографічна стійкість токенів.....	47
3.5 Архітектура системи та діаграма класів .....	48
3.5.1 Компонентна структура клієнтської частини .....	48
3.5.2 Архітектура серверної частини та сервісний шар .....	49
3.5.3 Логіка потоків даних.....	50
3.6 Реалізація клієнтської та серверної складових інформаційної системи .....	52
3.6.1 Реалізація клієнтської частини та інтерфейсу ролей .....	52
3.6.2 Реалізація серверної логіки та захист маршрутів .....	53
3.6.3 Специфіка взаємодії з Google Sheets та конфігурація розгортання .....	53
Висновки до розділу 3 .....	54
4 АПРОБАЦІЯ СИСТЕМИ ТА АНАЛІЗ ЕФЕКТИВНОСТІ РІШЕНЬ.....	56
4.1. Опис користувацьких інтерфейсів та сценаріїв взаємодії (Керівництво користувача).....	56
4.2. Методологія проведення пілотного експерименту .....	62
4.3 Аналіз результатів та оцінка ефективності системи .....	65

Висновок до розділу 4 .....	68
ВИСНОВКИ.....	70
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	73
ДОДАТОК А.....	79
ДОДАТОК Б .....	103

# ПЕРЕЛІК СКОРОЧЕНЬ, УМОВНИХ ПОЗНАЧЕНЬ І ТЕРМІНІВ

## Скорочення:

СКБД — система керування базами даних

ШІ — штучний інтелект

API (Application Programming Interface) — програмний інтерфейс застосунку

ARS (Audience Response Systems) — системи аудиторного опитування

BM25 (Best Matching 25) — алгоритм ймовірнісного зважування термінів

CSRF (Cross-Site Request Forgery) — підробка міжсайтового запиту

DDDM (Data-Driven Decision Making) — прийняття рішень на основі даних

DOM (Document Object Model) — об'єктна модель документа

DSR (Design Science Research) — дизайн-орієнтоване дослідження

EaaS (Education as a Service) — освіта як сервіс

FPS (Frames Per Second) — кількість кадрів на секунду

HMR (Hot Module Replacement) — гаряче оновлення модулів

HTTP (HyperText Transfer Protocol) — протокол передачі гіпертексту

I/O (Input/Output) — введення-виведення

ICAP (Interactive, Constructive, Active, Passive) — модель когнітивного залучення

JSON (JavaScript Object Notation) — текстовий формат обміну даними

JWT (JSON Web Token) — стандарт веб-маркерів JSON

LLM (Large Language Model) — велика мовна модель

MVP (Minimum Viable Product) — мінімально життєздатний продукт

NLP (Natural Language Processing) — обробка природної мови

Q&A (Questions and Answers) — сесія питань та відповідей

RBAC (Role-Based Access Control) — управління доступом на основі ролей

REST (Representational State Transfer) — архітектурний стиль взаємодії компонентів розподіленого застосунку

SOLO (Structure of Observed Learning Outcome) — таксономія структури спостережуваних результатів навчання

SPA (Single Page Application) — односторінковий застосунок

SVG (Scalable Vector Graphics) — масштабована векторна графіка

TF-IDF (Term Frequency — Inverse Document Frequency) — частота слова — зворотна частота документа

TTL (Time to Live) — час життя токена

VDOM (Virtual Document Object Model) — віртуальна об'єктна модель документа

XSS (Cross-Site Scripting) — міжсайтовий скриптинг

**Умовні позначення:**

$F1 - score$  (F-міра) — узагальнена метрика, середнє гармонійне між точністю та повнотою

*Precision* (Точність) — метрика семантичної чистоти аналітичного сигналу

*Recall* (Повнота) — метрика когнітивного охоплення цільового об'єкта знань

$S$  — масив неструктурованих коротких текстових відповідей користувачів

$T'$  — множина нормалізованих термінів, отримана з відповідей користувачів

$T^*$  — еталонна множина термінів (детермінована цільова модель знань)

## ВСТУП

**Актуальність теми.** Сучасні цифрові освітні середовища характеризуються переходом від статичних моделей організації навчального процесу до динамічних, інтерактивних форматів, зокрема гейміфікованих [1–3]. У таких середовищах навчальний сценарій формується та змінюється безпосередньо під час взаємодії зі студентами, що вимагає оперативного аналізу їхніх відповідей і своєчасного коригування подальшого ходу заняття [1, 3].

Одним із джерел інформації про стан засвоєння навчального матеріалу є відповіді студентів. Водночас використання виключно закритих форм контролю знань (тестових завдань із фіксованими варіантами відповідей) не дозволяє повною мірою оцінити глибину розуміння навчального матеріалу, оскільки обмежує можливість виявлення структури знань здобувачів та зв'язків між окремими поняттями [4, 5]. У зв'язку з цим у сучасних освітніх практиках все більшого поширення набувають відкриті форми відповідей, які передбачають формулювання студентами власних текстових пояснень, визначень або асоціацій. Текстові відповіді дозволяють відобразити структуру знань студентів, що неможливо досягти за допомогою закритих форм контролю.

Однак використання відкритих текстових відповідей породжує нову проблему: отримані дані є неструктурованими [6, 7]. Ручна обробка таких масивів призводить до втрати динаміки навчального сценарію та не відповідає вимогам гейміфікованого середовища. На передній план виходить проблема відсутності формалізованого механізму прийняття рішень у реальному часі. Це створює необхідність у застосуванні автоматизованих методів лінгвістичної обробки неструктурованого тексту для нормалізації термінів, їх зважування, порівняння з еталоном, обчислення метрик та фінальної інтерпретації результатів для формування керованого рішення і адаптації навчального сценарію.

Огляд існуючих підходів до навчальної аналітики свідчить про активний розвиток формалізованих, параметризованих та латентних моделей, зокрема

моделей відстеження знань [9, 10], теорії відповіді на завдання [11, 12], методів тематичного моделювання [13]. Попри їхню високу прогностичну ефективність, застосування таких складних математичних апаратів у межах окремого заняття в режимі квазіреального часу є методично та технічно обмеженим. Натомість більш практичним інструментом обробки текстових даних та їх візуалізації виступають хмари термінів.

Сучасні дослідження розглядають застосування хмар термінів переважно як дидактичний або інструментальний прийом для швидкого узагальнення коротких письмових відповідей та ініціювання обговорень [6, 7]. Водночас існуючі підходи дозволяють лише агрегувати відповіді студентів та отримувати узагальнене уявлення про їх зміст. Дослідники фіксують суттєві обмеження таких інструментів, зокрема втрату контексту, чутливість до якості тексту та ризику спрощення інтерпретації [8]. Існуючі засоби не забезпечують формалізованого переходу від результатів аналізу до прийняття обґрунтованих рішень щодо подальшого розвитку навчального сценарію.

Таким чином, виникає потреба у створенні підходу, який поєднує засоби обробки текстових відповідей, їх формалізованого представлення та обчислювального аналізу з механізмами інтерпретації результатів, орієнтованими на підтримку керованого прийняття рішень. Такий підхід дозволяє перетворити результати аналізу відповідей студентів на формалізовані критерії вибору подальших дій у гейміфікованому навчальному середовищі, забезпечуючи адаптивну зміну навчального сценарію в режимі реального часу.

**Мета дослідження** полягає у розробленні моделі керованого прийняття рішень на основі формалізованого аналізу текстових відповідей студентів та її програмної реалізації у вигляді системи, здатної автоматично інтерпретувати результати аналізу та визначати подальші дії навчального процесу.

**Завдання дослідження.** Для досягнення поставленої мети необхідно вирішити такі завдання:

- Проаналізувати особливості цифрових освітніх середовищ та визначити вимоги до систем формувального аналізу знань у гейміфікованому контексті.

- Дослідити підходи до обробки текстових відповідей та методи визначення значущості термінів і обґрунтувати їх використання для аналізу коротких відповідей студентів.
- Розробити модель представлення знань студентів у вигляді множини термінів та сформулювати еталонну модель знань для подальшого порівняння.
- Розробити метод порівняння множин термінів на основі обчислення показників повноти, точності та узгодженості.
- Розробити модель інтерпретації отриманих показників на основі формалізованих правил.
- Розробити алгоритм керованого прийняття рішень, що визначає подальші дії навчального процесу на основі результатів аналізу.
- Реалізувати програмну систему, що інтегрує розроблені моделі та алгоритми в єдиний процес обробки даних та підтримки прийняття рішень.
- Провести експериментальну перевірку ефективності запропонованого підходу та проаналізувати отримані результати.

**Об'єкт дослідження** — процес формувального аналізу знань студентів на основі текстових відповідей у цифровому навчальному середовищі.

**Предмет дослідження** — методи та моделі формалізованого представлення знань студентів у вигляді множини термінів, їх аналізу та інтерпретації на основі текстових відповідей.

**Методи дослідження.** Для вирішення поставлених завдань використано: методи обробки природної мови (англ. Natural Language Processing, NLP) для токенизації та лематизації текстів; теорію множин для обчислення подібності текстових масивів; методологію дизайн-орієнтованого дослідження (англ. Design Science Research, DSR) для проєктування системи; а також описові статистичні методи (обчислення середніх значень та порівняння груп) для обробки результатів експерименту.

**Наукова новизна** отриманих результатів полягає у розробленні моделі керованого прийняття рішень, що реалізує алгоритмічний перехід від результатів аналізу текстових відповідей студентів до визначення подальших дій навчального

процесу шляхом відображення кількісних показників подібності множин термінів у якісні рівні структури знань на основі формалізованих правил.

**Практичне значення роботи** полягає у створенні програмної системи, що реалізує алгоритм керованого прийняття рішень та забезпечує автоматизовану обробку коротких текстових відповідей із подальшим формуванням рекомендацій для адаптації сценарію заняття. Експериментальна перевірка, проведена за участю 30 студентів, показала зростання результативності експериментальної групи на 13,8 %.

**Апробація результатів.** Основні результати дослідження доповідалися на Міжнародній науково-практичній конференції та опубліковані у фаховому виданні.

Магістерська дисертація містить 112 сторінок, 2 додатки, 14 зображень, 5 таблиць і 60 джерел у переліку посилань. У вступі розкрито актуальність теми, визначено об'єкт, предмет, мету та завдання дослідження, а також обґрунтовано структуру та обсяг роботи.

Перший розділ присвячений постановці задачі розробки системи керованого прийняття рішень у межах парадигми «освіта як сервіс». У розділі проаналізовано роль гейміфікації та когнітивних моделей оцінювання, таких як таксономія структури спостережуваних результатів навчання (англ. Structure of Observed Learning Outcome, далі — таксономія SOLO) та модель когнітивного залучення (англ. Interactive, Constructive, Active, Passive, далі — рамка ICAP). Здійснено порівняльний аналіз сучасних систем аудиторного опитування та визначено обмеження існуючих методів лінгвістичного аналізу в цифрових освітніх системах.

У другому розділі розглянуто математичне та алгоритмічне забезпечення системи аналізу знань. Описано конвеєр обробки природної мови для аналізу текстових відповідей та еволюцію моделей зважування термінів від алгоритму «частота слова — зворотна частота документа» (англ. Term Frequency — Inverse Document Frequency, TF-IDF) до Okapi BM25. Детально обґрунтовано вибір асиметричних метрик подібності (Precision, Recall, F1-score) та розроблено логіку інтерпретації результатів аналізу через призму рівнів розуміння таксономії SOLO.

Третій розділ містить опис проєктування та програмної реалізації

інформаційної системи. Проведено обґрунтування технологічного стека, зокрема використання бібліотеки React, середовища Node.js та сервісів Firebase. Описано інноваційну архітектуру зберігання даних на базі Google Sheets та механізми забезпечення безпеки. Представлено компонентну структуру системи та логіку потоків даних між клієнтською та серверною частинами.

Четвертий розділ присвячений апробації спроєктованої системи та аналізу її ефективності. Описано сценарії взаємодії користувачів із системою та методологію проведення практичного експерименту на базі лекції про еволюцію зір. Наведено результати незалежного оцінювання, які підтвердили зростання успішності студентів. Проаналізовано вплив системи на кероване прийняття рішень викладачем та когнітивну активність аудиторії.

Висновки містять узагальнені результати дослідження, підтверджують виконання поставлених завдань та окреслюють перспективи подальшого розвитку системи. Список використаних джерел включає перелік фахової літератури, документації та наукових праць, що стали теоретичним підґрунтям роботи.

Додатки містять публікацію у фаховому науковому журналі та публікацію тез конференції.

# **1 ПОСТАНОВКА ЗАДАЧІ ТА ОГЛЯД МЕТОДІВ І ЗАСОБІВ СТВОРЕННЯ СИСТЕМИ КЕРОВАНОГО ПРИЙНЯТТЯ РІШЕНЬ**

Розробка спеціалізованого інформаційно-аналітичного інструментарію вимагає чіткого розуміння проблеми, цілей дослідження та визначення технологічних підходів до їх досягнення. У першому розділі наводиться детальна постановка задачі розробки програмного продукту керованого прийняття рішень для гейміфікованого навчального середовища. Підрозділ 1.1 визначає конкретні вимоги до системи, її функціональне призначення та архітектурні обмеження. У підрозділі 1.2 проводиться аналіз сучасних методів формувального оцінювання, когнітивних моделей (зокрема таксономії SOLO) та візуальної аналітики в контексті сервісно-орієнтованої освіти. Підрозділ 1.3 присвячений огляду програмних засобів, методів обробки природної мови та баз даних, що дозволить визначити оптимальний інструментарій для реалізації поставленого завдання.

## **1.1 Постановка задачі розробки системи керованого прийняття рішень**

В умовах цифровізації освітнього простору та зростаючої популярності змішаних і дистанційних форм навчання виникає гостра потреба в інструментах, які здатні забезпечити безперервний, об'єктивний та миттєвий зворотний зв'язок між викладачем та аудиторією. Однією з головних проблем, з якою стикаються викладачі при проведенні формувального оцінювання у великих академічних групах, є проблема «інформаційного перевантаження». Спроба проаналізувати масив неструктурованих текстових відповідей від десятків студентів у режимі, реального часу вимагає значних часових витрат і часто унеможливорює надання якісного зворотного зв'язку безпосередньо під час заняття.

Загальна мета розробки полягає у створенні програмної системи для агрегації відкритих студентських відповідей, генерації візуальних хмар слів та їх автоматизованого оцінювання без застосування надлишкових обчислювальних потужностей. Програма має функціонувати як ефективний інструмент підтримки прийняття рішень, що допомагає викладачу оперативно оцінити рівень розуміння теми та скоригувати подальший хід лекції.

Виходячи з цього, конкретна постановка інженерної задачі декомпозується на декілька логічних етапів. По-перше, система повинна мати механізм отримання коротких текстових відповідей від аудиторії та формування єдиного масиву неструктурованих даних. Разом із цим передбачається можливість задання викладачем «еталонної хмари» — набору ключових понять, які вичерпно описують тему.

По-друге, необхідна реалізація модуля лінгвістичної обробки та нормалізації тексту. Цей етап включає видалення знаків пунктуації, фільтрацію стоп-слів та лематизацію, щоб семантично ідентичні поняття зводилися до єдиної базової форми і коректно відображалися у підсумковій статистиці.

По-третє, система повинна агрегувати дані та будувати візуальну хмару термів, де розмір кожного слова залежить від частоти його використання (або обчисленої ваги), що супроводжується гейміфікованими елементами для стимулювання активності студентів.

Наступною вимогою є математичне порівняння згенерованої студентами множини термінів з еталонною множиною за допомогою метрик інформаційного пошуку: покриття (Recall), чистоти (Precision) та гармонічного середнього (F1-score).

Остаточним завданням є розробка логіки інтерпретації отриманих метрик за рівнями таксономії SOLO [14]. На основі обчислених даних система повинна автоматично класифікувати результат групи та згенерувати рекомендацію для викладача, наприклад, зупинитися на повторенні базових концепцій, провести дискусію для закріплення або перейти до більш складних практичних завдань.

У роботі розглядається задача аналізу текстових відповідей, у межах якої

формується множина термінів  $T'$ , отримана з відповідей, та еталонна множина термінів  $T^*$ , що відображає предметну область. Задача полягає у визначенні ступеня відповідності між множинами  $T'$  та  $T^*$  і подальшій інтерпретації результатів такого порівняння у вигляді дискретних рівнів за таксономією SOLO.

## **1.2 Роль гейміфікації та когнітивні моделі оцінювання (таксономія SOLO, рамка ICAP)**

Інтеграція механізмів формувального оцінювання у сервісно-орієнтоване середовище вимагає високого рівня залученості здобувачів освіти. Просте перенесення традиційних тестів у цифровий формат часто не вирішує проблеми мотивації, особливо під час роботи з великими аудиторіями. У цьому контексті потужним інструментом підтримки активної взаємодії виступає гейміфікація — використання ігрових елементів, візуальних нагород та змагальної динаміки у неігрових, навчальних контекстах [15]. Гейміфікація процесу формувального оцінювання трансформує рутинне опитування на захоплюючу спільну взаємодію, що значно знижує когнітивний стрес та страх зробити помилку, оскільки тестування сприймається як частина командної роботи [15].

Водночас сучасні дослідження підкреслюють, що поверхневе застосування зовнішніх ігрових атрибутів (балів, бейджів, таблиць лідерів) без їх інтеграції у когнітивну структуру навчання не забезпечує сталого освітнього ефекту. Такий підхід породжує системні ризики: ефект надмірного виправдання, коли зовнішні стимули витісняють внутрішню мотивацію, а також маніпулятивну адаптацію поведінки, за якої дії оптимізуються виключно під кількісні показники [16]. Тому більш перспективним є підхід конструктивної гейміфікації. При цьому саму хмару термінів не слід розглядати як ігрову механіку чи візуалізацію. Гейміфікацією є безпосередньо процес її формування, що включає колективну участь аудиторії, миттєвий внесок кожного окремого студента та наочний вплив індивідуального результату на загальну картину, що зміщує фокус здобувачів на процеси аналізу,

синтезу та рефлексії навчального матеріалу.

Проте сама по собі гейміфікована візуалізація не дає кількісної або якісної оцінки академічного рівня студентів. Для того, щоб результати інтерактивного оцінювання стали основою для прийняття керуючих рішень, їх необхідно інтерпретувати через призму визнаних когнітивних моделей. Найбільш релевантною для аналізу структури відкритих відповідей є таксономія SOLO, розроблена Дж. Біггсом та К. Коллісом [14]. На відміну від популярної таксономії Блума, яка орієнтована на класифікацію внутрішніх типів мислення, SOLO фокусується виключно на вимірюванні структурної складності спостережуваних результатів навчання [14].

Таксономія SOLO визначає чітку ієрархію розуміння матеріалу: від доструктурного (використання нерелевантної інформації) та одноструктурного (фрагментарний фокус) до багатоструктурного (накопичення непов'язаних фактів) і реляційного рівня, де окремі факти об'єднуються у цілісну структуру (рисунок 1.1) [14]. У даній роботі таксономія SOLO не використовується як пряма модель оцінювання, а застосовується як інтерпретаційний рівень, на який відображаються числові метрики подібності текстових множин.

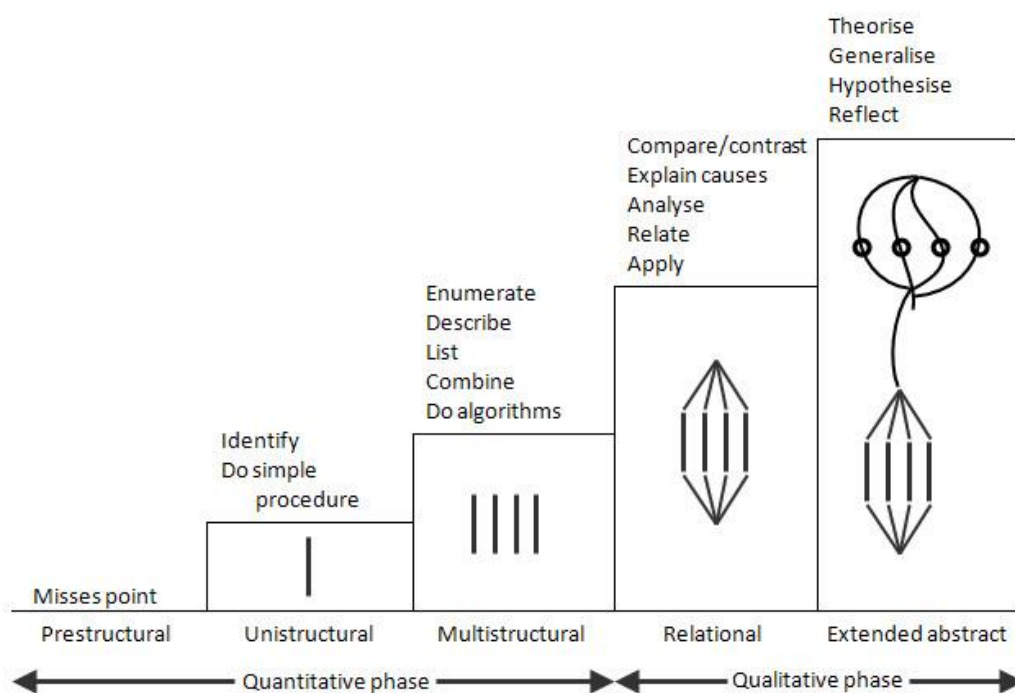


Рисунок 1.1 – Схема таксономії SOLO [<https://obl.ku.dk/theme/how-can-you-use-different-e-learning-tools-to-evaluate-student-learning/>]

Окрім оцінки структури знань, важливо розуміти характер залучення студентів до роботи, для чого доцільно використовувати рамку ICAR [17]. Ця концепція класифікує навчальне залучення не за формальними показниками (наприклад, часом, проведеним у системі), а за типом когнітивної діяльності: пасивне, активне, конструктивне та інтерактивне. У межах генерації хмар термінів продукування власних понять інтерпретується як конструктивний режим, а участь в обговоренні згенерованої хмари та взаємодія з відповідями одногрупників — як інтерактивний. Згідно з теоретичними положеннями ICAR, саме ці вищі рівні залучення асоціюються з найкращими результатами навчання [17].

Таким чином, поєднання таксономії SOLO для оцінювання структурних аспектів знань та рамки ICAR для аналізу характеру когнітивного залучення створює надійне методологічне підґрунтя. Це дозволяє перетворити агрегований зворотний зв'язок (у вигляді гейміфікованої хмари термінів) на чіткі індикатори, необхідні системі для автоматизованої генерації сценарних рішень для викладача.

### **1.3 Огляд та порівняльний аналіз сучасних систем аудиторного опитування**

Протягом останніх років глобальний освітній ландшафт зазнав безпрецедентних трансформацій, зумовлених як екстремним переходом до дистанційного навчання, так і інтеграцією передових цифрових інструментів у традиційні навчальні практики. У цьому контексті системи аудиторного опитування (англ. Audience Response Systems — ARS), такі як Mentimeter, Slido та Kahoot!, стали невід'ємним компонентом сучасного освітнього процесу в закладах вищої освіти [18]. Вони функціонують як комплексні медіасистеми, що забезпечують комунікативну модель взаємодії між викладачем та аудиторією, дозволяючи отримувати миттєвий зворотний зв'язок та створювати інтерактивне навчальне середовище [18].

Попередні академічні дослідження неодноразово підтверджували загальний

позитивний вплив ARS на мотивацію студентів та рівень їхньої участі в навчальному процесі [19]. Зокрема, гейміфіковані платформи, такі як Kahoot!, лідирують за рівнем впровадження, оскільки вони сприяють розвитку соціальних навичок та забезпечують дружню конкуренцію під час формувального оцінювання [20]. Інструменти Mentimeter та Slido позиціонуються як універсальні платформи для вищої освіти, орієнтуючись на сесії запитань та відповідей (Q&A) та візуалізацію текстових даних за допомогою генерації «хмар слів» [21–23].

Застосування таких систем дозволяє ефективно вирішити проблему психологічного бар'єра. Технологічно доповнена взаємодія з аудиторією довела свою здатність створювати безпечне психологічне середовище через механізми анонімності [19]. Студенти, які зазвичай утримуються від активності через страх зробити помилку, отримують голос у цифровому просторі, що значно розширює охоплення аудиторії під час обговорення складних тем [24].

Однак незважаючи на широке впровадження цих технологій, архітектура та аналітичний інструментарій більшості комерційних ARS залишаються фундаментально обмеженими з точки зору глибинної когнітивної оцінки та семантичного аналізу [25]. Для розуміння цих обмежень необхідно розглянути базовий функціонал провідних платформ (таблиця 1.1).

Таблиця 1.1 — Порівняльний функціональний ландшафт існуючих ARS-систем

Платформа	Основний фокус	Ключові формати взаємодії	Можливості аналітики та обробки тексту (NLP)
Mentimeter	Інтерактивні лекції та опитування	Хмари слів, шкали, відкриті питання	Відсутність глибинного NLP-аналізу та складних інструментів лематизації.
Slido	Q&A сесії та вебінари	Краудсорсинг питань з апвоутингом	Можливість генерації «AI-підсумків» виключно у преміум-сегменті

Продовження таблиці 1.1

Kahoot!	Гейміфікація та вікторини	Тести на швидкість, змагання	Базова статистика успішності без семантичного аналізу тексту відповідей.
Microsoft Forms	Асинхронний збір даних	Анкети, довгі текстові відповіді	Повна відсутність вбудованої динамічної візуалізації або хмар слів у реальному часі.
Vevox	Гібридні зустрічі та опитування	Опитування, Q&A, вікторини	Інтеграція з ШІ для генерації тестів та оцінки їхньої складності.

Дані таблиці демонструють технологічний парадокс: хоча платформи різняться за бізнес-моделями, їхній інструментарій для обробки та аналізу відкритих текстових відповідей є універсально обмеженим [21–23, 26–28]. Переважна більшість систем спирається на просту лінійну агрегацію відповідей. Хоча ці платформи забезпечують миттєву трансляцію даних на екран, вони перекладають весь когнітивний тягар інтерпретації складних візуальних образів на плечі викладача, що в умовах великих лекційних потоків робить автоматизоване формувальне оцінювання практично неможливим [25].

Отже, жодна з розглянутих систем не реалізує автоматизовану інтерпретацію текстових відповідей із використанням формальної моделі знань та не підтримує прийняття керуючих рішень на основі обчислювальних метрик.

#### **1.4 Обмеження методів лінгвістичного аналізу та візуалізації в сучасних цифрових системах**

Незважаючи на доведену ефективність систем аудиторного опитування у підвищенні рівня студентської залученості, глибокий аналіз їхнього внутрішнього

алгоритмічного забезпечення виявляє суттєві епістемологічні та технологічні обмеження. Основна проблема полягає в тому, що більшість комерційних платформ розроблялася як інструменти для корпоративних презентацій, а не як спеціалізовані діагностичні системи.

Метод візуалізації текстових масивів у вигляді хмар термінів (англ. word clouds) набув поширення як інструмент швидкого формувального оцінювання [19]. Основний принцип їх побудови полягає у відображенні слів, де візуальна вага (розмір шрифту) є прямо пропорційною частоті появи слова у тексті, що дозволяє миттєво отримати загальне уявлення про домінуючі концепти [20]. Проте з інженерної точки зору використання простої метрики частотності (англ. Term Frequency, TF) є фундаментальним недоліком. З погляду інформаційного пошуку такий підхід призводить до явища семантичної втрати: загальноживані поняття отримують найбільший розмір, витісняючи специфічні, вузькопрофільні терміни, які є справжніми маркерами глибокого розуміння теми [29–31]. Сучасні дослідження доводять, що проста частотна хмара слів без застосування штрафних коефіцієнтів формує у викладача ілюзію колективного розуміння, маскуючи реальні когнітивні прогалини аудиторії [32].

Ця проблема додатково посилюється примітивністю вбудованих конвеєрів лінгвістичної обробки. Переважна більшість доступних систем не підтримує глибоку лематизацію для специфічної академічної термінології. Замість зведення слів до базової форми вони використовують алгоритми стемінгу або просту токенизацію за пробілами. Внаслідок цього однокореневі слова чи синоніми відображаються як окремі візуальні сутності, що штучно занижує вагу правильних відповідей, розсіюючи їх між кількома графічними елементами [33].

Наступним критичним обмеженням є нездатність існуючих систем обробляти «короткий текст». Відповіді студентів під час блиц-опитувань зазвичай складаються з 1–3 слів. Класичні алгоритми ранжування виявляються неефективними в таких умовах, оскільки вони були розроблені для аналізу великих масивів. Робота з короткими текстами вимагає застосування більш адаптивних математичних моделей, здатних враховувати нелінійне насичення частоти, що

наразі не реалізовано в масових платформах [33].

Найбільш вагомим недоліком сучасних систем аудиторного опитування є їхній відрив від когнітивних моделей. Система надає викладачу лише набір агрегованих даних, залишаючи процес їхньої інтерпретації ручним. Автоматизовані системи не здатні обчислити метрики повноти засвоєння матеріалу і класифікувати результати групи за рівнями таксономії SOLO [14] або рамки когнітивного залучення ICAP [30]. Відсутність такого інтелектуального шару обґрунтовує гостру науково-практичну необхідність розробки нового програмного продукту, який би поєднував у собі вдосконалені алгоритми нормалізації тексту та автоматизовану логіку генерації рекомендацій.

Крім того, існуючі системи не використовують еталонну модель знань, що унеможливорює формальне оцінювання повноти засвоєння навчального матеріалу.

## **Висновки до розділу 1**

Комплексний аналіз предметної області та дослідження теоретико-методологічних засад розробки систем керованого прийняття рішень для гейміфікованих освітніх середовищ показали, що сучасна цифровізація вищої освіти вимагає переходу до сервісної парадигми, ключовим елементом якої є безперервне формувальне оцінювання. Обґрунтовано, що використання конструктивної гейміфікації сприяє зниженню психологічних бар'єрів та підвищенню студентської залученості, а застосування когнітивних моделей, зокрема таксономії SOLO та рамки ICAP, створює необхідне методологічне підґрунтя для об'єктивного вимірювання структурної складності засвоєних знань.

Детальний порівняльний аналіз провідних систем аудиторного опитування (Mentimeter, Slido, Kahoot! та ін.) виявив технологічний парадокс: незважаючи на популярність цих інструментів, їхній аналітичний апарат є універсально обмеженим. Дослідження свідчить, що використання примітивної лінгвістичної обробки та наївної метрики частотності при генерації хмар слів призводить до

семантичної втрати та формування хибної ілюзії розуміння матеріалу. Крім того, аналіз показав, що існуючі комерційні платформи не адаптовані для обробки коротких академічних текстів і не мають вбудованих механізмів інтелектуальної інтерпретації результатів.

Виявлені недоліки та методичні обмеження сучасних цифрових інструментів візуалізації обґрунтовують науково-практичну актуальність магістерського дослідження. Результати проведеного аналізу дозволили сформулювати постановку задачі, яка полягає у розробці спеціалізованого програмного продукту, здатного інтегрувати вдосконалені алгоритми обробки природної мови, математично стійкі моделі зважування термінів та автоматизовану логіку перетворення кількісних метрик інформаційного пошуку на якісні рекомендації для викладача.

## 2 МАТЕМАТИЧНЕ ТА АЛГОРИТМІЧНЕ ЗАБЕЗПЕЧЕННЯ СИСТЕМИ АНАЛІЗУ ЗНАНЬ

Розробка ефективної системи керованого прийняття рішень вимагає надійного алгоритмічного підґрунтя, здатного обробляти неструктуровані текстові дані та перетворювати їх на кількісні показники. У другому розділі розглядається математична модель системи та наводиться детальний опис алгоритмів, використаних для вирішення поставленої задачі. Підрозділ 2.1 присвячений архітектурі конвеєра обробки природної мови, який відповідає за лінгвістичне очищення вхідних даних. У підрозділі 2.2 формалізується процес утворення цільового аналітичного сигналу та еталонної множини знань. Підрозділ 2.3 описує математичний апарат алгоритму BM25 для ймовірнісного зважування термінів. У підрозділі 2.4 розглядаються метрики кількісного оцінювання семантичної відповідності, а підрозділ 2.5 фокусується на алгоритмі лінійної нормалізації ваг для візуального представлення. Підрозділи 2.6 та 2.7 присвячені розробленій моделі інтерпретації аналітичного сигналу за рівнями таксономії SOLO та узагальненій архітектурі аналітичного конвеєра загалом.

### 2.1 Формалізація процесу формування множини термінів

Вхідними даними для процесу аналізу у цифровому середовищі є масив коротких текстових відповідей отриманих від користувачів у реальному часі  $S = \{s_1, s_2, \dots, s_n\}$ , де  $s_n$  — відповідь  $n$ -го користувача. Оскільки ці дані є неструктурованими, для їхнього перетворення у формалізоване представлення, придатне для подальших математичних операцій, необхідне застосування методів обробки природної мови [31]. Для побудови адекватної термінологічної моделі вхідний сигнал має пройти через конвеєр лінгвістичної нормалізації.

Першим етапом обробки є токенізація, яка передбачає розбиття вхідних текстових рядків на окремі атомарні одиниці (токени). Це дозволяє виокремити слова як базові об'єкти аналізу та підготувати їх до подальшої фільтрації [33]. Далі здійснюється видалення службових слів та знаків пунктуації, що дозволяє очистити масив від інформаційного шуму, який не несе специфічного семантичного навантаження у контексті предметної області. Це критично важливо для систем, що працюють у реальному часі, оскільки дозволяє суттєво зменшити розмірність оброблюваного простору ознак [31].

Ключовим етапом нормалізації є лематизація — приведення кожного виокремленого токена до його базової форми [33]. На відміну від простіших методів, таких як стемінг, лематизація враховує морфологічний контекст, що важливо для коректної обробки академічної термінології та усунення синтаксичної варіативності. Такий підхід забезпечує ідентифікацію семантично еквівалентних концептів, які можуть бути виражені різними граматичними формами.

В результаті проходження зазначеного конвеєра лінгвістичної обробки формується множина термінів  $T'$ , яка є об'єднанням усіх нормалізованих токенів, отриманих із масиву відповідей  $S$ . Дана множина виступає у ролі формалізованого аналітичного сигналу, що відображає поточний стан системи та використовується як базис для подальшого математичного порівняння та обчислення метрик. Це представлення дозволяє системі оперувати очищеними даними, мінімізуючи вплив індивідуальних особливостей написання тексту на фінальний результат аналізу.

## 2.2 Методологія формування еталонної множини термінів

Для забезпечення об'єктивності аналізу та можливості розрахунку кількісних показників відхилення системи від цільового стану необхідно визначити базисний вектор порівняння. У запропонованій моделі таку роль відіграє еталонна множина термінів  $T^*$ , яка репрезентує детермінований цільовий стан знань предметної області. Дана множина виступає як «Ground Truth» — набір істинних значень,

відносно яких здійснюється оцінка повноти та точності отриманого аналітичного сигналу  $T'$  [31].

Формування  $T^*$  здійснюється експертним шляхом до початку циклу взаємодії. Даний метод передбачає виокремлення ключових дескрипторів, які найбільш повно описують структуру знань у межах конкретного сценарію. Обов'язковою умовою є попередня лематизація всіх елементів  $T^*$ , що дозволяє забезпечити семантичну відповідність на етапі зіставлення з нормалізованим масивом токенів, виділених із відповідей користувачів [33].

Ефективність аналітичного контуру безпосередньо залежить від повноти та несуперечливості еталонної множини. Критерій повноти  $T^*$  визначає верхню межу значущості отриманого результату. У випадку недостатньої репрезентативності еталона виникає ризик хибної інтерпретації характеристик знань, що унеможливорює коректне відображення стану системи на відповідні рівні таксономії. Експертний підхід до формування моделі дозволяє мінімізувати вплив семантичного шуму та гарантувати релевантність бази порівняння. Таким чином,  $T^*$  виконує функцію фіксованого оператора: будь-яке рішення щодо вибору подальшого сценарію взаємодії ґрунтується на математично обчисленій відстані між поточним аналітичним сигналом  $T'$  та статичним еталоном  $T^*$  [31].

### **2.3 Математичне обґрунтування методу зважування термінів у коротких текстах**

Обчислення характеристик сформованої множини  $T'$  відносно еталона  $T^*$  потребує застосування математичного апарату зважування термінів для визначення їхньої інформаційної значущості. У сучасній практиці інтелектуального аналізу даних підходи до оцінки ваги термінів зазвичай класифікують на частотні, частотно-інверсні та ймовірнісні моделі [31, 33]. Вибір конкретного методу критично впливає на здатність системи коректно інтерпретувати вхідний сигнал, особливо за умов обмеженого обсягу даних. У межах розробки системи підтримки

прийняття рішень вибір моделі зважування зумовлений необхідністю об'єктивної обробки коротких масивів тексту, що генеруються у процесі цифрової взаємодії.

У сучасній парадигмі систем інформаційного пошуку та інтелектуального аналізу даних обробка та класифікація коротких текстів виокремилися у фундаментальну дослідницьку галузь. Головною проблемою аналізу коротких текстів є їхня феноменальна розрідженість простору ознак [35]. У коротких відповідях користувачів більшість термінів зустрічається лише один раз, що руйнує традиційні частотні припущення, на яких базуються класичні векторні моделі

Для перетворення неструктурованого тексту на числові ознаки застосовується метод зважування термінів. Історичним стандартом у цій сфері є метод TF-IDF. Базова математична формалізація ваги TF-IDF для терміна  $t$  у документі  $d$  визначається як добуток локальної частоти та логарифму зворотної частоти документа [36]:

$$TFIDF_{d,t} = tf_{d,t} \cdot \log\left(\frac{N}{df_t}\right),$$

де  $tf_{d,t}$  — частота терміна у документі,

$N$  — загальна кількість документів (відповідей) у вибірці,

$df_t$  — кількість документів, які містять даний термін.

Проте класичний підхід TF-IDF володіє серйозними теоретичними обмеженнями при роботі з короткими текстами, оскільки він позбавлений вбудованого механізму адекватного насичення частоти терміна та спирається на лінійні методи нормалізації довжини [37].

Для усунення цих недоліків у межах запропонованого аналітичного контуру застосовується алгоритм BM25, який є сучасною ймовірнісною еволюцією класичної моделі. Архітектура BM25 значно вдосконалює попередні моделі шляхом інтеграції двох ключових евристичних інновацій: нелінійного асимптотичного насичення частоти терміна та складної динамічної нормалізації за довжиною документа. Згідно з оригінальною специфікацією, вага конкретного терміна  $q_i$  по відношенню до масиву відповідей  $D$  обчислюється за такою формулою [37]:

$$\text{score}(D, q_i) = \text{IDF}(q_i) \cdot \frac{f(q_i, D) \cdot (k_1 + 1)}{f(q_i, D) + k_1 \cdot \left(1 - b + b \cdot \frac{|D|}{\text{avgdl}}\right)},$$

де  $f(q_i, D)$  — частота зустрічання терміна  $q_i$  у документі  $D$ ,

$|D|$  — довжина поточного документа (кількість токенів у текстовому вході користувача),

$\text{avgdl}$  — середня довжина документів у загальному масиві відповідей,

$k_1$  — коефіцієнт насичення частоти терміна, який обмежує вплив багаторазового повторення одного й того ж слова (за емпіричними рекомендаціями авторів алгоритму зазвичай обирається в діапазоні від 1.2 до 2.0),

$b$  — коефіцієнт нормалізації за довжиною, що визначає ступінь впливу довжини відповіді на фінальну вагу (стандартне значення  $b = 0.75$ ).

Ключовим компонентом формули є множник зворотної частоти документа  $\text{IDF}(q_i)$ , який діє як штрафний коефіцієнт для занадто популярних слів і обчислюється наступним чином [37]:

$$\text{IDF}(q_i) = \ln \left( \frac{N - n(q_i) + 0.5}{n(q_i) + 0.5} + 1 \right),$$

де  $N$  — загальна кількість відповідей (документів) у вибірці,

$n(q_i)$  — кількість відповідей, у яких зустрічається даний термін.

Використання параметрів  $k_1$  та  $b$  дозволяє системі гнучко адаптуватися до специфіки коротких відповідей. На відміну від TF-IDF, де кожне наступне входження слова дає лінійний приріст ваги, у моделі BM25 вплив частоти досягає жорсткої верхньої межі. Це критично важливо для аналітичних систем, оскільки захищає результати від штучних маніпуляцій (англ. keyword stuffing). Таким чином, BM25 забезпечує більш стійке та об'єктивне зважування термінів для їх подальшої алгоритмічної нормалізації та інтерпретації.

Важливо зазначити, що алгоритм BM25 використовується виключно для формування вагових коефіцієнтів термінів та їх подальшої візуалізації у вигляді хмари слів. Результат його обчислень не використовується у метриках семантичної відповідності.

## 2.4 Математичні метрики порівняння множин та логіка інтерпретації

Для отримання об'єктивного аналітичного сигналу та подальшого прийняття рішень щодо адаптації сценарію взаємодії необхідно провести кількісне зіставлення сформованої множини термінів  $T'$  з еталоном  $T^*$ . Основним інструментарієм для такого оцінювання у системах інтелектуального аналізу тексту є метрики точності (англ. Precision), повноти (англ. Recall) та їхнього гармонійного поєднання — F-міри (англ. F1-score) [31].

Важливою особливістю розробленого підходу є те, що метрики семантичної відповідності обчислюються тільки по факту наявності терміна у множині (бінарний принцип: 0 або 1). Вагові коефіцієнти BM25 на цьому етапі спеціально не використовуються. Це зроблено для чіткого відокремлення оцінки структури знань від простої частоти використання окремих слів, що забезпечує більшу стійкість алгоритму прийняття рішень.

Математичне визначення точності дозволяє оцінити чистоту отриманого аналітичного сигналу. Цей показник відображає частку релевантних термінів серед усіх знайдених у відповідях користувачів та обчислюється як [31]:

$$Precision = \frac{|T' \cap T^*|}{|T'|}$$

З точки зору аналізу, висока точність свідчить про те, що користувачі чітко фокусуються на цільовій термінології предметної області, не вносячи до системи семантичного шуму чи помилкових дескрипторів [31].

Натомість повнота характеризує рівень охоплення цільового об'єкта знань. Вона визначає частку знайдених термінів відносно загальної кількості дескрипторів у еталонній множині [31]:

$$Recall = \frac{|T' \cap T^*|}{|T^*|}$$

Низькі значення повноти сигналізують про фрагментарне відображення предметної області у вхідному масиві даних  $S$ , що вказує на дефіцит ключових концептів у процесі взаємодії [31].

Оскільки максимізація лише одного з показників може призвести до хибних висновків про стан системи, використовується узагальнена оцінка у формі F-міри (F1-score). Вона базується на розрахунку середнього гармонійного між точністю та повнотою, що дозволяє виявити дисбаланс у семантичному сигналі [33]:

$$F1 = 2 \cdot \frac{Precision \cdot Recall}{Precision + Recall}$$

Узагальнена логіка інтерпретації результатів аналізу наведена у таблиці 2.1.

Таблиця 2.1 – Характеристики та інтерпретація метрик семантичної відповідності

Метрика	Об'єкт вимірювання	Ознаки низьких значень	Ознаки високих значень
Precision	Семантична чистота	Наявність сторонніх асоціацій, термінологічний шум	Використання виключно валідних дескрипторів
Recall	Когнітивне охоплення	Пропуски фундаментальних понять предметної області	Повне відтворення структури еталонної моделі
F1-score	Системна рівновага	Суттєвий перекося між релевантністю та повнотою сигналу	Цілісне та виважене представлення знань

Таке поєднання є критично важливим для контуру підтримки прийняття рішень, оскільки дозволяє системі ідентифікувати сценарії, де висока точність досягається за рахунок критично низького охоплення теми або навпаки.

## 2.5 Застосування алгоритму BM25 для нормалізації термінів та формування їх ваги

Після обчислення значень інформаційної значущості кожного токена за алгоритмом BM25, описаним у підрозділі 2.3, виникає математична задача

відображення отриманих величин у нормований діапазон. Специфіка функції ранжування VM25 полягає у тому, що її результати є абсолютними числами, які не мають жорсткої верхньої межі та суттєво залежать від потужності множини  $S$  та дисперсії довжини текстових входів [31].

Використання безпосередніх значень ваг для подальшої інтерпретації ускладнює процес формування стабільного аналітичного сигналу, оскільки варіативність числових показників може призводити до спотворення семантичних пропорцій у межах різних сесій взаємодії. Відповідно до принципів аналізу даних у складних системах перетворення кількісних атрибутів у репрезентативні форми вимагає їхнього попереднього масштабування до обмеженого інтервалу, що дозволяє уніфікувати подальші операції порівняння та візуального представлення [38].

Для розв'язання цієї задачі у межах розробленої моделі застосовується етап лінійної нормалізації (англ. Min-Max scaling) масиву обчислених ваг до заданого діапазону. Це класичний метод попередньої обробки даних, який гарантує перенесення значень у новий заданий інтервал без зміни характеру їхнього внутрішнього розподілу [39]. Нехай  $V_{min}$  та  $V_{max}$  — це встановлені у параметрах моделі граничні межі цільового діапазону значень. Тоді фінальна нормалізована вага  $\omega'_i$  для конкретного терміна обчислюється за формулою [39]:

$$\omega'_i = V_{min} + \frac{\omega_i - \omega_{min}}{\omega_{max} - \omega_{min}} \cdot (V_{max} - V_{min}),$$

де  $\omega'_i$  — поточна вага терміна, обчислена за VM25,

$\omega_{min}$  та  $\omega_{max}$  — відповідно найменша та найбільша ваги серед усіх унікальних термінів у поточній множині  $T'$ .

Така нормалізація забезпечує системну стабільність аналітичного конвеєра: найбільш семантично значущий концепт (із найвищим показником VM25) завжди набуває значення  $V_{max}$ , а найменш значущий дескриптор, що пройшов фільтрацію, — значення  $V_{min}$ . Решта термінів пропорційно розподіляється у межах цього коридору, що відповідає кращим практикам алгоритмічного формування термінологічних структур [40].

Двоетапна обробка — спочатку ймовірнісне зважування BM25 для об'єктивної математичної оцінки важливості концептів, а потім лінійна нормалізація простору ознак — забезпечує передбачувану генерацію результатів. Це дозволяє системі формувати інтуїтивно зрозумілий аналітичний зріз незалежно від загального обсягу вхідних даних чи довжини текстових відповідей користувачів у масиві  $S$ .

Отримані нормалізовані ваги використовуються тільки для візуалізації хмари термінів. Вони не впливають на розрахунок метрик і безпосередньо на алгоритм керованого прийняття рішень.

## **2.6 Модель інтерпретації аналітичного сигналу та правила прийняття рішень**

Ключовим етапом функціонування системи підтримки прийняття рішень є перетворення кількісних метрик у дискретні стани, що дозволяють ініціювати керуючий вплив. У запропонованій моделі цей процес формалізується як функція відображення значень семантичної відповідності на множину цільових рівнів. Модель інтерпретації описується кортежем:  $M = \langle T', T^*, L, f \rangle$ , де  $T'$  — множина нормалізованих термінів користувачів,  $T^*$  — еталонна множина знань,  $L$  — множина цільових станів системи, яка базується на рівнях таксономії SOLO [14], а  $f$  — логічна функція виводу. Таким чином, алгоритм визначає відображення множини текстових відповідей у множину керуючих впливів.

Робота моделі базується на двоетапному алгоритмі обробки даних. На першому етапі здійснюється зіставлення множин  $T'$  та  $T^*$  для обчислення числових характеристик точності, повноти та F-міри, як це визначено у стандартах інтелектуального аналізу текстів [31]. На другому етапі отримані характеристики інтерпретуються через систему продукційних правил, яка ідентифікує фінальний стан системи. Такий підхід дозволяє перевести абстрактні статистичні показники у конкретні характеристики семантичної структури відповіді.

Система правил прийняття рішень ґрунтується на наступних логічних умовах:

- Преструктурний рівень: ідентифікується за умови вкрай низького охоплення еталона, коли показник повноти є критично малим ( $Recall < 0,15$ ). Це свідчить про відсутність релевантних дескрипторів у вхідному сигналі [33].
- Уніструктурний рівень: стан, при якому спостерігається висока точність ( $Precision > 0,6$ ) при низькому показнику повноти ( $Recall < 0,4$ ). Це означає виокремлення системою лише одиничних коректних концептів без формування цілісного семантичного контексту.
- Мультиструктурний рівень: ситуація, коли показник повноти є достатнім ( $Recall \geq 0,5$ ), проте точність суттєво знижена через наявність інформаційного шуму ( $Precision < 0,5$ ). Це вказує на наявність дескрипторів без встановлення ієрархічних зв'язків між ними [31].
- Реляційний рівень: досягається при збалансованих високих значеннях узагальненої метрики ( $F1 > 0,7$ ), що підтверджує формування стійкої семантичної структури, яка відповідає еталону  $T^*$ .
- Рівень розширеного абстрагування: стан, за якого система фіксує не лише повне охоплення еталона ( $Recall > 0,8$ ), а й генерацію розширеного концептуального поля, що виражається у перевищенні потужності множини  $|T'|$  над потужністю еталона  $|T^*|$  у 1,3 раза і більше (зазначений коефіцієнт обрано емпірично як поріг, що відокремлює розширення семантичного поля від надлишкового шуму).

Варто наголосити, що наведені порогові значення для класифікації станів не є статичними константами. Вони визначаються емпіричним шляхом на основі аналізу репрезентативних вибірок даних та оптимізуються під час пілотної експлуатації системи для мінімізації похибки класифікації. Такий підхід є загальноприйнятою практикою для систем зі зворотним зв'язком, оскільки дозволяє адаптувати чутливість аналітичного контуру до складності конкретної предметної області [33].

Схематично алгоритм інтерпретації метрик у рівні SOLO наведено на рисунку 2.1.

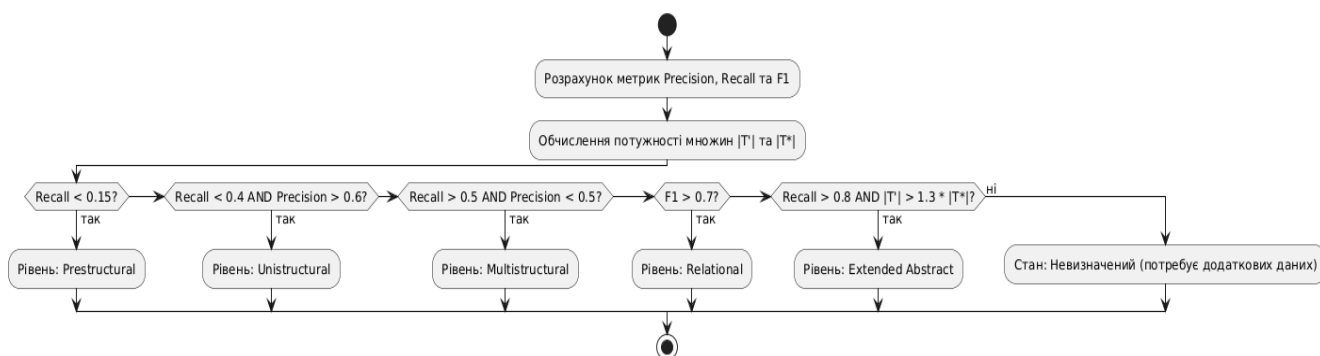


Рисунок 2.1 — Схема алгоритму інтерпретації метрик у рівні SOLO

Зведена система правил, що описує залежність між числовими інтервалами метрик та рівнями інтерпретації, наведена у таблиці 2.2.

Таблиця 2.2 — Правила відображення метрик семантичної відповідності у рівні SOLO

Рівень SOLO	Критерій (логічна умова)	Характеристика стану системи
<b>Prestructural</b>	$Recall < 0,15$	Відсутність релевантного семантичного сигналу у вхідних даних
<b>Unistructural</b>	$Recall < 0,4 \wedge Precision > 0,6$	Фрагментарна фіксація одиничних концептів без системних зв'язків
<b>Multistructural</b>	$Recall \geq 0,5 \wedge Precision < 0,5$	Наявність незв'язної надлишкової термінології та семантичного шуму.
<b>Relational</b>	$F1 > 0,7$	Висока семантична цілісність та релевантність структури знань
<b>Extended Abstract</b>	$Recall > 0,8$	Генерація розширеного концептуального поля та встановлення нових семантичних зв'язків поза межами еталона

Застосування наведеної системи правил дозволяє детерміновано перетворювати кількісні результати роботи лінгвістичного конвеєра на якісні

індикатори. Це створює надійне алгоритмічне підґрунтя для автоматизації процесу генерації рекомендацій без необхідності ручного втручання викладача в математичний апарат оцінювання.

## 2.7 Узагальнений алгоритм керованого прийняття рішень на основі аналізу текстових відповідей

Підсумовуючи викладене у попередніх підрозділах, процес функціонування запропонованої системи підтримки прийняття рішень формалізується у вигляді алгоритму обробки даних та прийняття рішень. На початковому етапі неструктуровані текстові відповіді користувачів (масив  $S$ ) проходять процедури лінгвістичної обробки — токенізацію, фільтрацію та лематизацію, утворюючи цільовий аналітичний сигнал у вигляді множини термінів  $T'$ . Специфіка обробки такого сигналу зумовлена високою розрідженістю даних, що потребує застосування спеціалізованих методів аналізу коротких текстів [41].

Далі цей сигнал розгалужується на два паралельні алгоритмічні потоки. Перший потік — для візуалізації, де для кожного терміна обчислюється його інформаційна значущість за ймовірнісною моделлю BM25 із подальшою лінійною нормалізацією для формування збалансованого цільового представлення. Другий потік — для прийняття рішень, де множина  $T'$  порівнюється з детермінованим еталоном  $T^*$  для обчислення базових метрик семантичної відповідності: точності, повноти та F-міри [42, 43], виключно на основі бінарної присутності концептів. На завершальному етапі отримані кількісні значення метрик надходять до моделі інтерпретації, яка на основі системи емпіричних правил та статистичних критеріїв оцінювання результатів [44] ідентифікує поточний стан знань за таксономією SOLO, що слугує об'єктивною підставою для генерації керуючого впливу та адаптації подальшого сценарію цифрової взаємодії.

Узагальнена схема аналітичного конвеєра обробки даних та прийняття рішень наведена на рисунку 2.2.

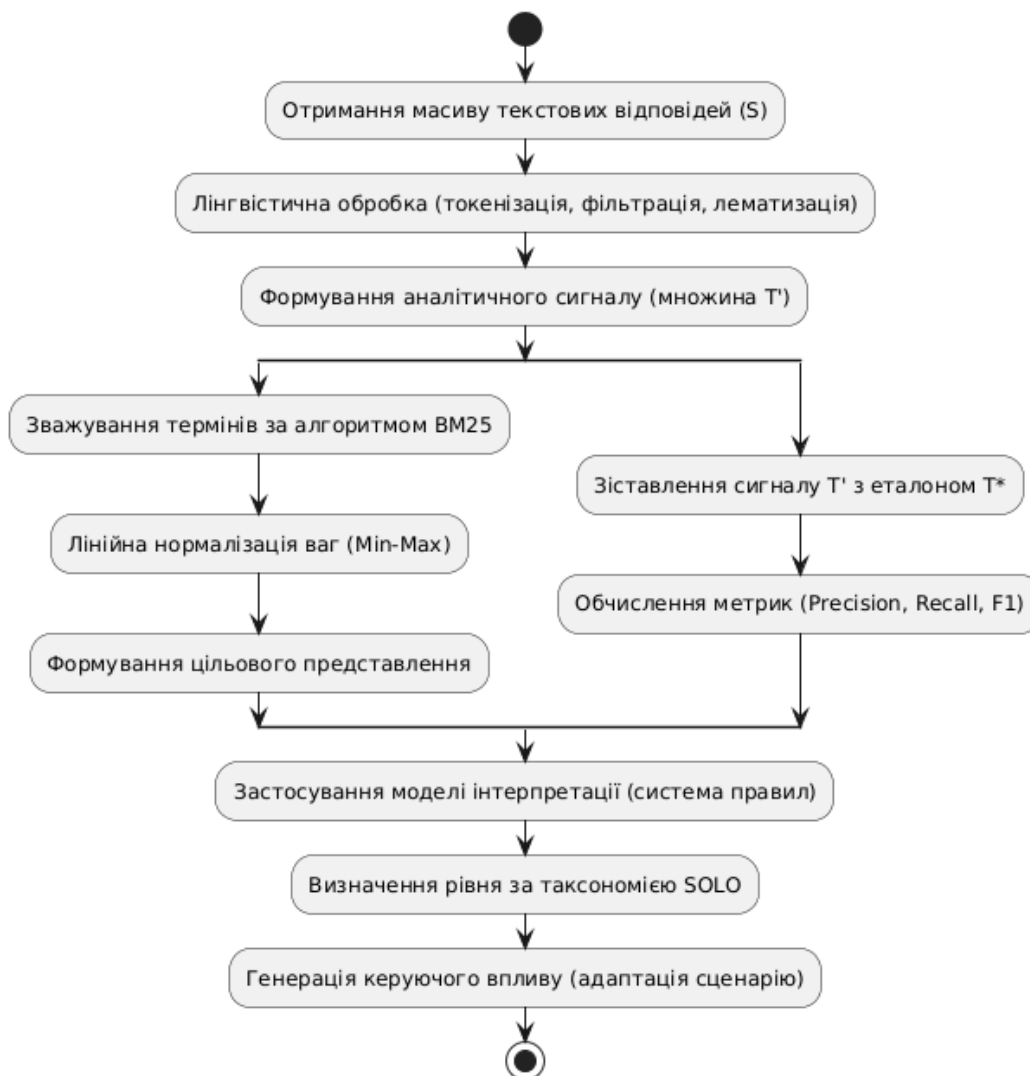


Рисунок 2.2 – Узагальнена схема конвеєра обробки даних та прийняття рішень

Запропонована архітектура аналітичного конвеєра забезпечує строгу послідовність виконання обчислювальних операцій та мінімізує ризик втрати даних на проміжних етапах.

## Висновки до розділу 2

Математичне та алгоритмічне забезпечення інформаційної системи підтримки прийняття рішень, спроектоване у другому розділі, спирається на попередній лінгвістичний конвеєр. Дослідження обґрунтовує необхідність його

застосування для нормалізації відповідей користувачів та подолання проблеми високої синтаксичної варіативності й розрідженості простору ознак коротких текстів.

Створена модель представлення знань формалізує єдиний нормалізований аналітичний сигнал, який живить алгоритм з двома незалежними потоками обробки даних. Перший потік, спрямований на візуальне представлення результатів, використовує ймовірнісну модель зважування Окарі VM25 у поєднанні з лінійною нормалізацією Min-Max. Аналіз показує, що такий підхід враховує механізми нелінійного насичення частоти, запобігає маніпуляціям та забезпечує об'єктивне формування цільових розмірів для побудови концептуальних хмар слів.

Другий потік, орієнтований на генерацію керуючого впливу, реалізує метод порівняння множин термінів (користувацьких відповідей та еталонної моделі) на основі асиметричних метрик точності, повноти та узагальненої F-міри. Обчислення цих показників спирається виключно на бінарне зіставлення множин без урахування статистичних ваг, що дозволяє відокремити структурну цілісність відповіді від частотних характеристик термінів.

Спроектowana модель інтерпретації застосовує систему емпірично визначених формалізованих правил для автоматичної класифікації поточного стану знань аудиторії за рівнями таксономії SOLO. На основі розрахованих показників алгоритм керованого прийняття рішень визначає відповідність між множиною текстових відповідей та множиною подальших дій навчального процесу.

У підсумку, запропонований комплексний алгоритм забезпечує повний цикл роботи з інформацією: від сирого тексту до формування обґрунтованих рекомендацій щодо адаптації сценарію цифрової взаємодії. Це забезпечує формалізований перехід від аналізу текстових даних до обґрунтованих керуючих впливів.

## 3 ПРОЄКТУВАННЯ ТА ПРОГРАМНА РЕАЛІЗАЦІЯ СИСТЕМИ

Процес розробки інтерактивної освітньої системи, орієнтованої на аналіз великих масивів текстових даних у реальному часі, потребує комплексного підходу до проєктування архітектурних рішень. Третій розділ присвячений детальному обґрунтуванню технологічного стека та опису інженерної реалізації програмної системи підтримки прийняття рішень.

У підрозділі 3.1 наведено аргументацію вибору ключових інструментів розробки клієнтської та серверної частин, а також підсистем зберігання даних та автентифікації. Подальші підрозділи (3.2–3.4) фокусуються на глибокому технічному аналізі внутрішніх механізмів обраних технологій, що забезпечують працездатність системи: роботі віртуальної об'єктної моделі документа у React для динамічної візуалізації, асинхронній моделі Node.js для обробки потокових запитів та криптографічному захисту даних на основі стандартів веб-маркерів JSON (англ. JSON Web Token, JWT).

Підрозділ 3.5 присвячений безпосередньому проєктуванню архітектури програмного продукту, де описується компонентна структура, сервісний шар та детальна логіка потоків даних у межах аналітичного конвеєра. Завершується розділ підрозділом 3.6, в якому розглядаються особливості програмної реалізації клієнтської та серверної складових, специфіка взаємодії з хмарними API та конфігурація системи для розгортання у реальному навчальному середовищі.

### 3.1 Обґрунтування вибору технологічного стека

Розробка сучасних веб-орієнтованих освітніх систем у парадигмі EaaS висуває жорсткі вимоги до інструментарію розробника. Система повинна забезпечувати високу швидкість обробки користувацьких запитів, гнучкість

користувацького інтерфейсу для відтворення складної SVG-графіки (хмар термінів), а також безшовну інтеграцію з NLP-бібліотеками. З огляду на ці вимоги, базовою екосистемою проєкту було обрано екосистему JavaScript з обов'язковим використанням суворої статичної типізації за допомогою TypeScript. Такий підхід дозволив реалізувати концепцію ізоморфного (або універсального) коду, де клієнтська та серверна частини розмовляють «однією мовою», а спільні інтерфейси даних запобігають виникненню помилок під час мережевої взаємодії.

### **3.1.1 Вибір засобів розробки клієнтської частини (Frontend)**

Для реалізації інтерфейсу було обрано бібліотеку React (версії 18+) у поєднанні зі збирачем Vite [45]. Вибір React обґрунтований його компонентною архітектурою та наявністю віртуальної об'єктної моделі документа (англ. Virtual DOM, VDOM). Оскільки оновлення хмари термінів у реальному часі є ресурсомісткою операцією перемальовування сотень графічних вузлів, алгоритм узгодження React дозволяє мінімізувати прямі звернення до реального DOM (англ. Document Object Model, DOM), оновлюючи лише ті елементи, характеристики яких змінилися.

Використання Vite значно оптимізувало процес розробки порівняно з класичними бандлерами, забезпечивши миттєве гаряче оновлення модулів (англ. HMR) та швидку компіляцію TypeScript-коду [45]. Для розрахунку координат та візуалізації хмари термінів застосовано бібліотеку *react-d3-cloud*, що базується на математичному рушії *d3-cloud* [46]. Цей рушій використовує алгоритми пакування полігонів для запобігання накладанню слів, динамічно обчислюючи розмір шрифту на основі вагових коефіцієнтів алгоритму BM25.

### **3.1.2 Вибір засобів розробки серверної частини (Backend)**

Серверне ядро реалізовано на базі середовища Node.js з використанням фреймворку Express [47]. Асинхронна архітектура Node.js є оптимальною для задач з високим рівнем операцій вводу/виводу, зокрема для одночасного

приймання великої кількості відповідей від студентів. Важливою перевагою є нативне середовище для виконання NLP-бібліотеки winkNLP без необхідності розгортання окремих мікросервісів на інших мовах програмування.

Взаємодія побудована за стилем REST API на базі протоколу HTTP. У межах дослідження прийнято рішення відмовитися від постійних з'єднань (англ. WebSockets) на користь моделі «запит за вимогою» (англ. fetch on demand). Це спрощує розгортання системи на хмарних платформах, зменшує споживання оперативної пам'яті сервером та гарантує стабільність в умовах нестабільного з'єднання в аудиторіях.

### **3.1.3 Вибір підсистеми зберігання даних (Google Sheets як EaaS-база)**

Архітектурним рішенням стало використання Google Sheets як підсистеми зберігання даних. Взаємодія з таблицями реалізована через офіційний пакет Google APIs з використанням серверної аутентифікації [48].

Вибір Google Sheets обґрунтовано специфікою освітньої галузі: для викладача таблиця є звичним форматом роботи. Кожна сесія формує запис із міткою часу, ідентифікатором та JSON-об'єктом частотності. Такий підхід надає викладачу готовий аналітичний артефакт для експорту без написання SQL-запитів. Проте варто зауважити, що використання Google Sheets як сховища є доцільним саме для прототипу та систем середнього масштабу. Дане рішення має архітектурні обмеження і не призначене для масштабованих високонавантажених систем, де вимагається підтримка тисяч одночасних транзакцій на запис.

### **3.1.4 Вибір системи аутентифікації та розмежування ролей**

Оскільки система має два чітко розділені інтерфейси («кімната студента» для генерації даних та «панель викладача» для їх аналізу та прийняття рішень), виникла потреба в надійній системі захисту маршрутів. Для реалізації моделі управління доступом на основі ролей (англ. Role-Based Access Control, RBAC) використано Firebase Admin SDK [49]. Всі запити викладача до конфіденційних ендпойнтів: отримання агрегованої статистики, очищення бази, робота з

еталонною хмарою — підписуються спеціальним ID-токеном, який бекенд верифікує за допомогою Firebase Admin, гарантуючи захист навчальних даних від несанкціонованого втручання.

Узагальнена архітектурна схема технологічного стеку наведена на рисунку 3.1.

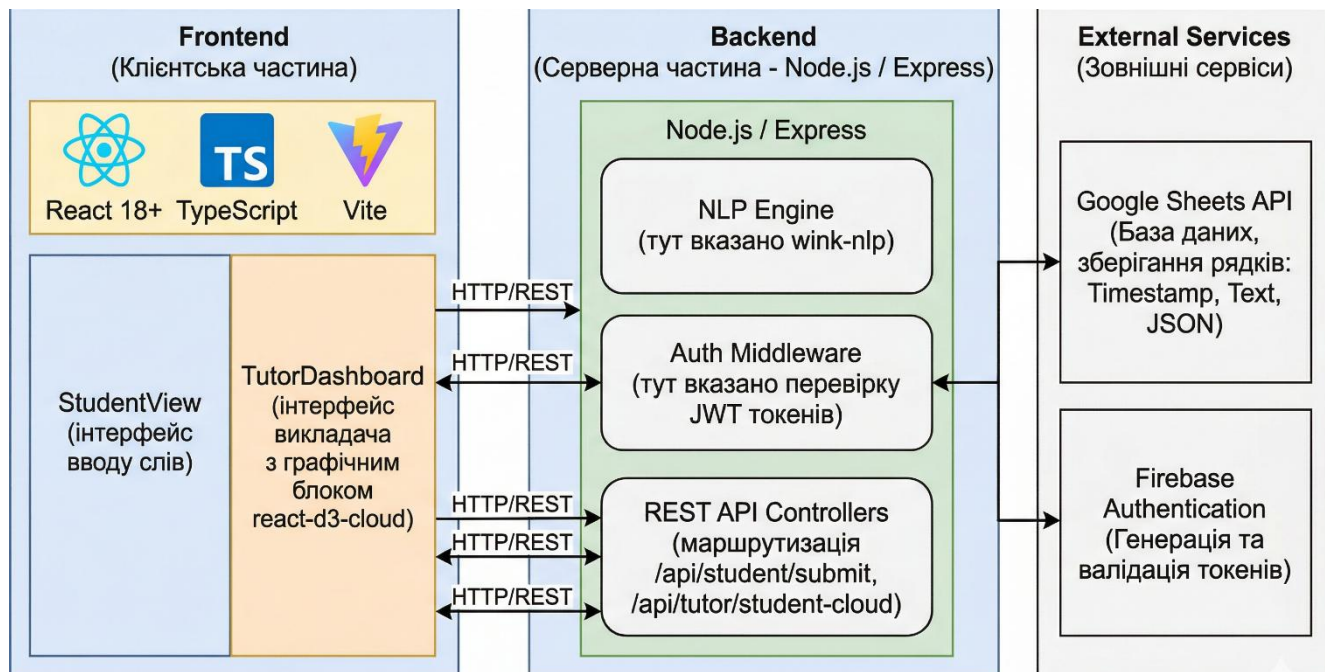


Рисунок 3.1 – Архітектурна схема технологічного стеку та взаємодії компонентів системи

Таким чином, інтеграція наведених технологічних рішень у єдиний програмний комплекс дозволяє повністю задовольнити вимоги парадигми «освіта як сервіс». Обрана архітектура гарантує не лише високу швидкість клієнтських інтерфейсів, а й забезпечує необхідний рівень безпеки та гнучкості для подальшого масштабування системи.

### 3.2 Теоретичні аспекти клієнтської оптимізації: Віртуальний DOM та алгоритми узгодження

Фундаментальною інженерною проблемою розробки динамічних освітніх інтерфейсів є ресурсомісткість прямих маніпуляцій з об'єктною моделлю документа. У межах розробленої системи, зокрема під час сесії спільної побудови

хмари слів, частота надходження нових термінів від аудиторії може бути високою. Оскільки кожне слово у хмарі є окремим SVG-елементом із власними координатами та параметрами трансформації, безперервне пряме оновлення реального DOM призвело б до значних витрат ресурсів центрального процесора та втрати чутливості інтерфейсу.

Для подолання цих обмежень у системі використано підхід, що базується на концепції Virtual DOM. Віртуальний DOM виступає як полегшене представлення інтерфейсу в оперативній пам'яті у вигляді JavaScript-об'єктів, що синхронізуються з реальним графічним деревом [50]. Це дозволяє застосувати декларативний підхід: програмна логіка описує лише кінцевий стан компонента WordCloud на основі масиву частотностей VM25, а бібліотека автоматично обчислює найбільш продуктивний шлях оновлення екрана.

Процес синхронізації віртуального представлення з реальним відображенням — узгодження (англ. Reconciliation) — дозволяє уникнути повного перемальовування інтерфейсу при кожній новій відповіді студента [51]. Для інтерактивної хмари термінів, що може містити сотні унікальних вузлів, використання стандартних алгоритмів порівняння дерев було б неефективним. Тому в системі реалізовано евристичний алгоритм порівняння з лінійною складністю  $O(n)$ , де ключову роль відіграє стабільність типів та використання унікальних ключів для кожного слова у хмарі [52]. Це означає, що якщо слово вже присутнє на екрані і змінилася лише його вага, система виконає точкову мутацію атрибута font-size, не видаляючи вузол.

Детальний порівняльний аналіз стратегій роботи з об'єктною моделлю документа, що обґрунтовує вибір даної архітектури для завдань візуалізації знань, наведено у таблиці 3.1.

Таблиця 3.1 — Порівняльний аналіз парадигм роботи з об'єктною моделлю документа

Характеристика	Традиційний (Реальний) DOM	Віртуальний DOM (React)
<b>Представлення даних</b>	Дерево об'єктів, безпосередньо прив'язане до графічного рушія браузера	Полегшені абстрактні JavaScript-об'єкти в оперативній пам'яті
<b>Швидкість маніпуляцій</b>	Низька. Кожна зміна ініціює глобальний перерахунок макета	Дуже висока. Обчислення відбуваються в пам'яті без впливу на екран.
<b>Стратегія оновлення</b>	Прямі, імперативні зміни через повільні API	Пакетні оновлення шляхом дифінгу та точкових мутацій.
<b>Механізм переривання</b>	Відсутній. Великі цикли невідворотно блокують інтерфейс	Наявний завдяки планувальнику React Fiber — інкрементальний рендеринг.
<b>Обчислювальна складність</b>	Складність алгоритмів порівняння довільних дерев становить $O(n^3)$	Використання оптимізованої евристики

Окрему увагу при проектуванні клієнтської частини приділено архітектурі React Fiber. Головною інновацією цього механізму є реалізація кооперативної багатозадачності, що дозволяє розбивати роботу на дрібні фрагменти (англ. work units). У контексті розробленої системи це має критичне значення: планувальник може призупиняти обчислення нових 2D-координат слів у хмарі, якщо викладач ініціює високопріоритетну подію, наприклад, перемикання між аналітичними вкладками або редагування еталона, і повертатися до рендерингу графіки в наступних кадрах анімації [50, 53]. Це забезпечує плавність роботи застосунку навіть під час інтенсивного математичного перерахунку семантичних ваг термінів.

### 3.3 Серверна архітектура: Node.js, Event Loop та асинхронність

Серверна частина розроблюваної системи базується на середовищі виконання Node.js, архітектурна модель якого оптимізована для задач із високою інтенсивністю мережевих операцій. В основі обраного рішення лежить подійний цикл (англ. Event Loop) та неблокуюча модель введення-виведення (англ. Non-blocking I/O). Для освітнього продукту, де десятки або сотні студентів одночасно надсилають текстові відповіді, здатність сервера обробляти велику кількість паралельних запитів без створення окремих потоків на кожне з'єднання є критичним фактором підтримки прийняття рішень у реальному часі.

На відміну від традиційних багатопотокових серверів, які використовують модель «один потік на один запит» і потребують значних витрат оперативної пам'яті на підтримку стеків потоків [54], Node.js використовує однопотоковий цикл подій. Це дозволяє делегувати тривалі операції, такі як звернення до API Google Sheets або виконання процедур лінгвістичного аналізу, системному ядру або пулу потоків через бібліотеку libuv [55]. Після завершення операції сервер отримує зворотний виклик, що дозволяє основному потоку залишатися вільним для прийняття нових вхідних масивів даних  $S$  від студентської аудиторії.

Архітектурний аналіз підтверджує, що така модель є більш ефективною для парадигми «освіта як сервіс». Забезпечуючи високу пропускну здатність при меншому споживанні пам'яті порівняно з системами на базі Java або PHP, Node.js дозволяє гнучко масштабувати систему горизонтально через механізми кластеризації. Неблокуюча природа середовища дозволяє паралелізувати запити до зовнішніх сховищ даних, не перериваючи логіку обробки природної мови, що є необхідною умовою для миттєвого оновлення аналітичного сигналу  $T'$ .

Важливою особливістю імплементації є безшовна інтеграція бібліотеки winkNLP безпосередньо у середовище JavaScript. Це усуває потребу у складній міжпроцесній взаємодії, яка була б неминучою при використанні Python-серверів, та суттєво знижує затримки під час трансформації сирих відповідей у зважені вектори термінів [56]. Таким чином, обрана серверна архітектура забезпечує

цілісність технологічного стека та високу швидкість обробки даних, необхідну для динамічної візуалізації освітніх результатів.

### 3.4 Забезпечення безпеки та криптографічна стійкість токенів

У парадигмі «освіта як сервіс», де програмний продукт оперує академічними результатами та навчальною аналітикою, питання захисту інформації та розмежування прав доступу є критичними. Для реалізації моделі управління доступом на основі ролей у розроблюваному модулі обрано стандарт JWT. Вибір цього механізму зумовлений його «безстановою» (англ. stateless) природою, що дозволяє REST-серверу верифікувати сесії без надмірних запитів до бази даних.

Криптографічна стійкість системи базується на трикомпонентній структурі токена. Замість загального опису полів, у системі реалізовано алгоритм хешування, де цифровий підпис обчислюється на основі заголовка та корисного навантаження із використанням секретного ключа на стороні хмарної інфраструктури [57]. Це гарантує, що дані, які ідентифікують роль користувача (викладач або студент), не можуть бути модифіковані зловмисником під час передачі.

Інтеграція з Firebase Admin SDK дозволяє побудувати надійний ланцюжок довіри. При проходженні автентифікації викладачем система генерує ID-токен, який передається у заголовок кожного запиту до конфіденційних ендпойнтів, наприклад, для очищення таблиці або зміни еталонної моделі. Серверний модуль виконує криптографічну верифікацію підпису за допомогою публічних сертифікатів [58]. Будь-яка спроба втручання в структуру токена призводить до невідповідності фінального хешу, і сервер негайно блокує доступ до аналітичних функцій [49].

Використання JWT у проєкті дозволило чітко ізолювати інтерфейси. Студентський профіль технічно позбавлений можливості звертатися до методів агрегації знань, оскільки бекенд-валідатор автоматично відхиляє запити без відповідного підпису ролі tutor. Додатково, для запобігання атакам типу

«міжсайтовий скриптинг» (англ. Cross-Site Scripting, XSS) та «міжсайтова підробка запиту» (англ. Cross-Site Request Forgery, CSRF), у системі встановлено обмежений час життя токена (англ. TTL), що мінімізує вікно потенційного використання викраденого ідентифікатора в мережі навчального закладу [59, 60]. Таким чином, криптографічні механізми забезпечують цілісність освітнього контенту та конфіденційність викладацької аналітики без зниження продуктивності системи.

### 3.5 Архітектура системи та діаграма класів

Архітектурна побудова системи базується на принципах модульності та суворого розподілу обов'язків (англ. Separation of Concerns). Такий підхід дозволяє ізолювати логіку лінгвістичного аналізу від механізмів графічної візуалізації та персистентного зберігання даних, що забезпечує високу підтримуваність коду. Програмна система спроектована за клієнт-серверною моделлю, де клієнтська частина відповідає за гейміфіковану взаємодію з користувачем та візуальне подання результатів, а серверна — за оркестрацію даних, виконання NLP-процедур та інтеграцію із хмарною інфраструктурою.

#### 3.5.1 Компонентна структура клієнтської частини

На рівні фронтенд-дodatка архітектура організована як ієрархія React-компонентів, де управління станом та життєвим циклом об'єктів здійснюється за допомогою функціональних хуків. Ключовим вузлом системи є кореневий компонент *App.tsx*, який виконує роль високорівневого диспетчера ролей. Залежно від результатів автентифікації та призначеної ролі користувача (викладач або студент), система динамічно монтує відповідні робочі простори.

Інтерфейс студента інкапсульований у модулі *StudentView.tsx*. Його внутрішня логіка передбачає не лише збір вхідного тексту, а й асинхронну комунікацію з бекендом для отримання проміжної статистики. Компонент візуалізації *WordCloud.tsx* є універсальним модулем, що отримує різні набори

даних залежно від контексту: персональний внесок студента або агрегований результат групи. Панель управління викладача *TutorDashboard.tsx* є найбільш функціонально насиченим вузлом, оскільки вона об'єднує в собі модулі моніторингу результатів групи, редактор еталонної моделі знань *EtalonEditor.tsx* та аналітичний блок відображення рекомендацій, що базуються на розрахованих метриках.

### 3.5.2 Архітектура серверної частини та сервісний шар

Серверна частина побудована за принципом сервісно-орієнтованої архітектури, де кожен модуль відповідає за окремий аспект обробки інформації. Основним вхідним пунктом системи є контролер *server.ts*, який описує маршрути REST API та координує потік запитів. Логіка взаємодії з Google Sheets винесена в ізольований сервісний шар *sheets.ts*. Це дозволяє абстрагувати складність роботи з низькорівневими запитами до Google APIs [48] та надає іншим модулям сервера можливість використовувати високорівневі методи, такі як *appendStudentRow()* для запису відповідей або *getStudentAggregate()* для отримання сумарної статистики знань.

Критично важливим елементом архітектури є NLP-модуль *nlp.ts*. Він інкапсулює в собі конфігурацію бібліотеки *winkNLP*, підключення лінгвістичних моделей (англ. *wink-eng-lite-web-model*) та механізми формування векторних представлень термінів [34]. Така ізоляція дозволяє змінювати параметри лематизації або додавати кастомні словники стоп-слів без модифікації коду маршрутизації сервера чи логіки роботи з базою даних. Це гарантує цілісність аналітичного конвеєра та стабільність обчислення вагових коефіцієнтів для кожного унікального терміна.

Узагальнена діаграма компонентів всієї розробленої системи наведена на рисунку 3.2.

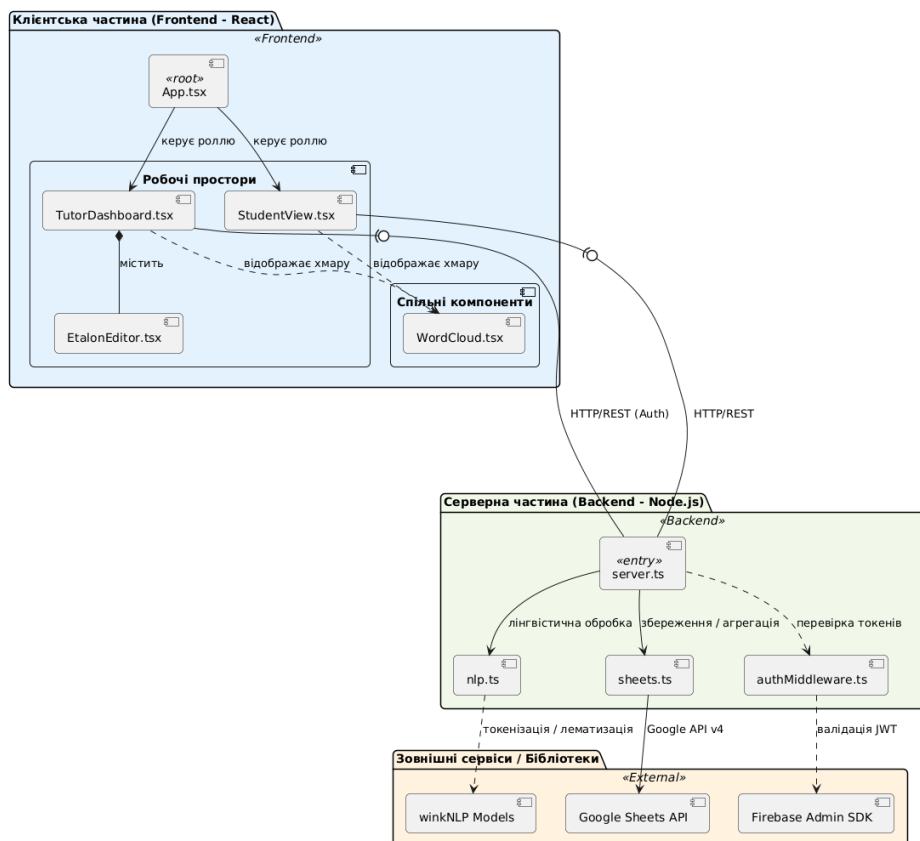


Рисунок 3.2 — Діаграма компонентів системи

Реалізована компонентна структура серверної частини ефективно мінімізує зв'язність між окремими вузлами системи. Суворі ізоляція логіки лінгвістичного аналізу від процесів маршрутизації та взаємодії з базою даних дозволяє масштабувати NLP-модуль або оновлювати словники стоп-слів без ризику порушення стабільності інших сервісів

### 3.5.3 Логіка потоків даних

Архітектура системи з точки зору обробки даних включає такі етапи: отримання тексту, лінгвістична нормалізація, обчислення ваг термінів, побудова множини, обчислення метрик та генерація рішення. Цей багатоступеневий процес функціонування системи в межах навчальної сесії чітко розділяється на два незалежні потоки взаємодії: етап збору даних та етап прийняття рішень.

На етапі збору даних ініціатором виступає студент. Користувач вводить терміни у клієнтському інтерфейсі, після чого фронтенд формує POST-запит до серверного маршруту `/api/response`. Серверна частина приймає вхідний масив і звертається до підсистеми зберігання Google Sheets для запису сирих відповідей

студента. Після успішного збереження рядка база даних повертає підтвердження, а сервер транслює статус успішної операції клієнту.

На етапі прийняття рішень ініціатором процесу є викладач, який відправляє запит на отримання аналітики сесії через GET-запит до маршруту */api/session/results*. Сервер звертається до бази даних для отримання всього масиву накопичених відповідей студентської групи. Отримані «сирі» дані передаються до спеціалізованого аналітичного ядра — SOLO Engine.

У межах SOLO Engine виконується ключовий етап обробки: розрахунок вагових коефіцієнтів термінів за алгоритмом BM25, обчислення метрик семантичної відповідності (зокрема F1-міри) та класифікація поточного стану знань аудиторії за рівнем таксономії SOLO. Після завершення обчислень аналітичне ядро повертає на бекенд визначений когнітивний рівень та згенеровану рекомендацію. Серверна частина формує фінальний аналітичний сигнал у форматі JSON і відправляє його на фронтенд, де відбувається візуалізація семантичної хмари слів та відображення конкретних порад щодо подальшого навчального сценарію для викладача (рисунок 3.3).

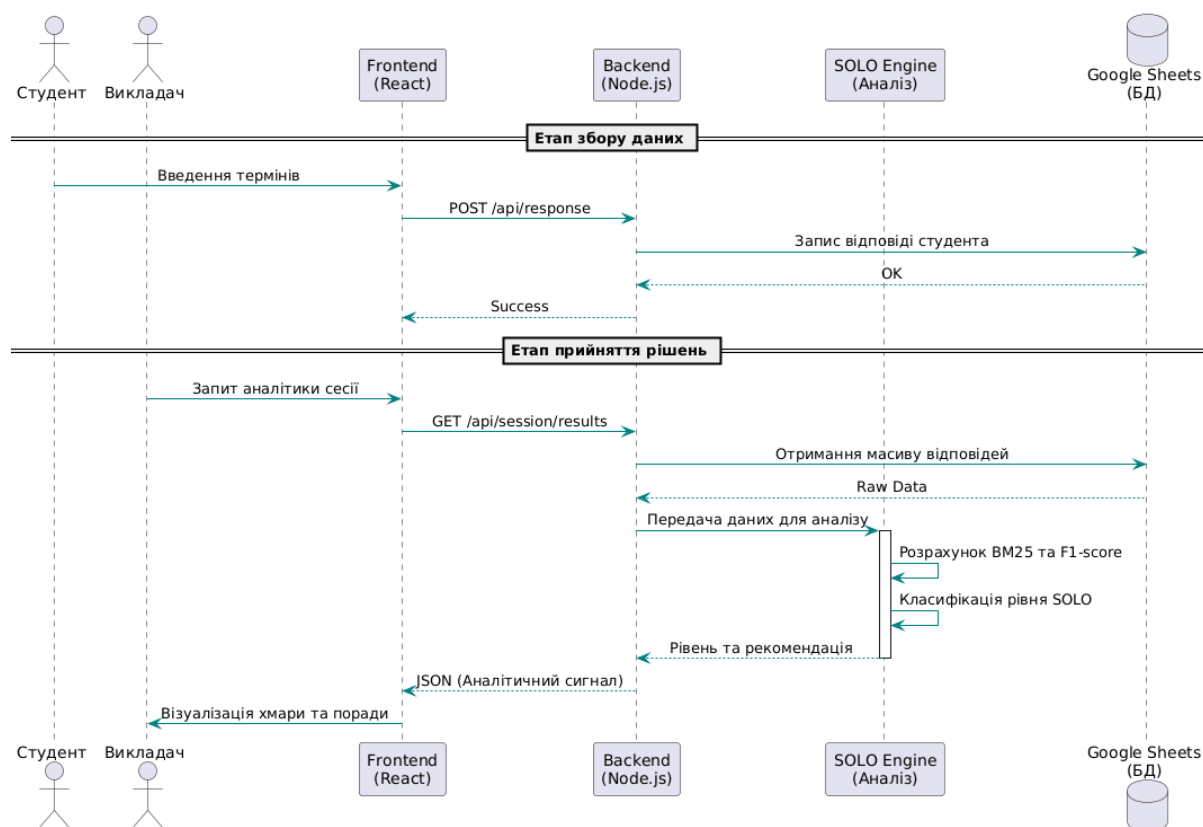


Рисунок 3.3 — Схема послідовності передачі даних

Наведена послідовність взаємодії чітко розмежовує фазу асинхронного збору інформації від студентів та фазу синхронної аналітичної обробки на боці викладача. Це оптимізує навантаження на серверну інфраструктуру та гарантує, що ресурсомісткі обчислення метрик у SOLO Engine ініціюються лише за необхідності, зберігаючи загальну реактивність веб-додатка.

## **3.6 Реалізація клієнтської та серверної складових інформаційної системи**

Програмна реалізація розробленої системи керованого прийняття рішень виконана з дотриманням принципів модульності та чіткого розмежування обов'язків між клієнтським і серверним середовищами. Кожен функціональний блок системи виконує детерміновану роль у загальному конвеєрі обробки даних, що забезпечує стабільність аналітичного сигналу та гнучкість системи до масштабування.

### **3.6.1 Реалізація клієнтської частини та інтерфейсу ролей**

Клієнтський застосунок побудований як односторінковий додаток на базі React. Для забезпечення ізоляції робочих процесів інтерфейс жорстко розділено на дві функціональні зони: середовище студента та панель управління викладача.

Середовище студента, реалізоване у модулі StudentView, містить мінімалістичну форму для введення вільного тексту. Після відправки даних фронтенд отримує підтвердження від сервера та автоматично ініціює запит на отримання актуальної хмари групи, дозволяючи студенту побачити свій внесок у загальний семантичний контекст.

Панель управління викладача виступає головним аналітичним центром. Доступ до неї захищений компонентом-обгорткою, який взаємодіє з сервісом аутентифікації. Основним елементом логіки викладача є модуль візуалізації когнітивних метрик. На відміну від попередніх ітерацій, у фінальній реалізації

клієнтська частина виступає споживачем обчислених даних: вона отримує від сервера вже готові значення точності, повноти та F1-міри, а також діагностований рівень за таксономією SOLO. Використання бібліотеки `react-d3-cloud` [46] дозволяє динамічно перемальовувати хмару термінів на основі вагових коефіцієнтів BM25, забезпечуючи викладача наочним інструментом для швидкої оцінки стану знань аудиторії.

### 3.6.2 Реалізація серверної логіки та захист маршрутів

Серверна частина, розроблена на Node.js з використанням фреймворку Express [47], функціонує як захищений шлюз. Маршрутизація системи структурована таким чином, щоб розділити потоки збору даних та потоки аналітики.

Приватні маршрути, призначені для викладача, наприклад, `/api/session/results` або команди очищення бази, захищені спеціалізованим проміжним ПЗ `authMiddleware.ts`. Цей модуль виконує вилучення токена із заголовка запиту та здійснює його верифікацію через Firebase Admin SDK [49]. Під час перевірки сервер не лише підтверджує автентичність токена, а й контролює наявність ролі `tutor` у навантаженні. Такий підхід гарантує, що студентський інтерфейс позбавлений технічної можливості звертатися до аналітичних функцій SOLO Engine або модифікувати параметри еталонної моделі знань  $T^*$ .

### 3.6.3 Специфіка взаємодії з Google Sheets та конфігурація розгортання

Для реалізації персистентного зберігання використано модуль `sheets.ts`, що базується на Google Sheets API [48]. Дані зберігаються у структурованому вигляді: позначка часу, ідентифікатор сесії, оригінальний текст відповіді та результати попереднього лінгвістичного аналізу. При запиті викладачем аналітики серверна функція виконує агрегацію даних безпосередньо перед передачею їх до аналітичного ядра, що дозволяє уникнути надлишкового навантаження на базу даних.

Для забезпечення безпечного розгортання на хмарних хостингах архітектура

системи повністю виключає зберігання секретних ключів у програмному коді. Всі критичні параметри, включаючи `GS_PRIVATE_KEY` та `FIREBASE_PROJECT_ID`, передаються через змінні оточення. Це дозволяє гнучко змінювати конфігурацію системи, без необхідності перезбирання всього проекту.

## Висновки до розділу 3

Комплексне проектування та програмна реалізація архітектури інформаційної системи на основі сучасного стека вебтехнологій підтверджують ефективність обраного інструментарію (React, Node.js, Google Sheets API) для створення освітніх систем у парадигмі «освіта як сервіс».

Вибір бібліотеки React для побудови клієнтської частини базується на аналізі переваг концепції Virtual DOM та архітектури React Fiber. Практична імплементація цих технологій дозволяє оптимізувати рендеринг складної SVG-графіки та зберегти високу чутливість інтерфейсу викладача під час інтенсивних математичних обчислень координат термінів.

Архітектурна логіка системи спирається на чіткий розподіл обов'язків між клієнтським та серверним середовищами, що мінімізує мережеві затримки. Застосування компонентного підходу дозволяє ізолювати аналітичні модулі від інтерфейсу візуалізації, забезпечуючи можливість незалежної модифікації алгоритмів обробки без переписування графічної частини. Такий підхід гарантує розширюваність системи та її здатність до інтеграції додаткових метрик семантичної відповідності у майбутньому.

Використання середовища Node.js як бекенд-платформи забезпечує високу пропускну здатність системи завдяки асинхронній моделі Event Loop, що є критично важливим під час масового одночасного надсилання відповідей студентами. Це дозволяє безшовно інтегрувати NLP-модуль на базі winkNLP та реалізувати аналітичне ядро SOLO Engine як централізований вузол прийняття рішень.

Впроваджена архітектура зберігання даних на основі Google Sheets виступає оптимальним рішенням для прототипування, оскільки автоматично формує готовий аналітичний артефакт для викладача. Разом з тим, аналіз виявив архітектурні обмеження Google Sheets, що зумовлює доцільність переходу на реляційні СКБД у разі масштабування системи до рівня високонавантаженого сервісу.

Надійна система захисту даних на основі токенів JWT та Firebase Admin SDK у поєднанні із запропонованою логікою потоків даних створює завершене інженерне рішення. Реалізований програмний продукт, що охоплює всі етапи від збору тексту до генерації рекомендацій, повністю готовий до експериментальної апробації у реальному навчальному процесі.

## **4 АПРОБАЦІЯ СИСТЕМИ ТА АНАЛІЗ ЕФЕКТИВНОСТІ РІШЕНЬ**

Розробка складних алгоритмічних рішень для освітнього середовища вимагає обов'язкової практичної перевірки їхньої дієвості. Четвертий розділ присвячений апробації створеного програмного продукту в умовах реального навчального процесу. У підрозділі 4.1 наведено керівництво користувача з детальним описом розроблених інтерфейсів та сценаріїв взаємодії у форматі мінімально життєздатного продукту (англ. Minimum Viable Product, MVP). Підрозділ 4.2 описує методологію проведення практичного експерименту з використанням інструментарію системи. У підрозділі 4.3 здійснюється аналіз отриманих результатів, оцінюється вплив алгоритмів керованого прийняття рішень на загальну результативність групи та обґрунтовується ефективність впровадження системи на основі таксономії SOLO.

### **4.1. Опис користувацьких інтерфейсів та сценаріїв взаємодії (Керівництво користувача)**

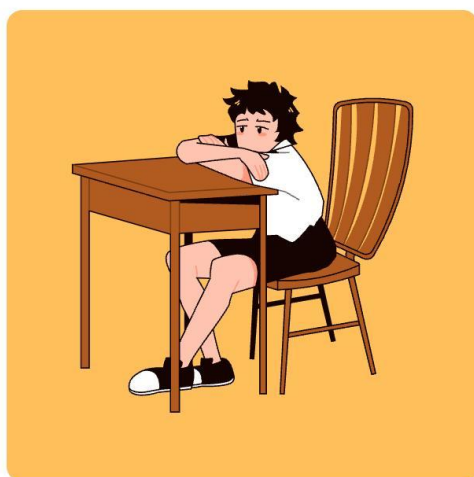
Програмний продукт реалізовано у форматі MVP, головним завданням якого є демонстрація роботи математичного ядра та забезпечення безперебійної комунікації між учасниками навчального процесу. Відповідно до спроектованої рольової моделі, взаємодія із системою розпочинається з екрана маршрутизації («Semantic Cloud Classroom»), де користувачу пропонується обрати свій статус: «I am a student» або «I am a tutor».

Для забезпечення максимальної доступності та уникнення затримок під час лекції вхід для студентів здійснюється без створення облікових записів. Після вибору відповідної ролі студент перенаправляється до свого робочого простору. Натомість доступ до аналітичної панелі викладача регламентований механізмом

безпарольного доступу (Passwordless Authentication) через інфраструктуру Firebase, що після верифікації email-адреси генерує JWT-токен і відкриває доступ до закритої частини системи (рисунок 4.1).

## Semantic Cloud Classroom

Choose your role:



I am a student



I am a tutor

Рисунок 4.1 — Стартовий екран маршрутизації ролей користувачів

Для завершення процедури авторизації та запобігання несанкціонованому доступу до аналітики, система ініціює додатковий крок перевірки безпеки. Як показано на рисунку 4.2, викладач має підтвердити свою корпоративну електронну адресу у модальному вікні браузера. Лише після успішної валідації введеної адреси через сервіс Firebase відбувається фінальна генерація JWT-токена та перенаправлення до закритої панелі управління.

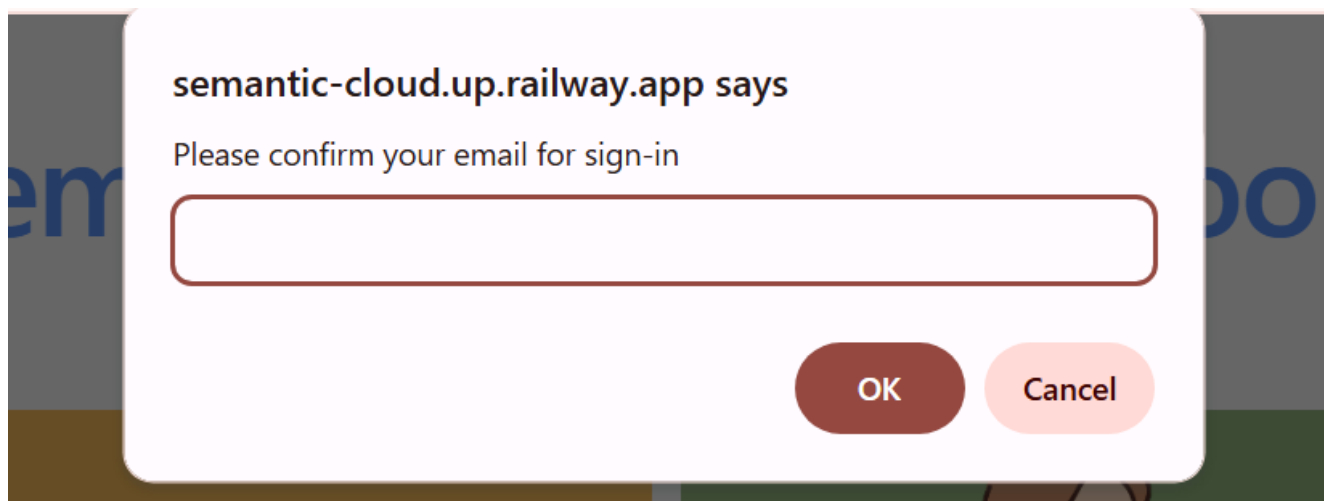


Рисунок 4.2 — Діалогове вікно підтвердження електронної адреси викладача для безпарольної авторизації

Панель управління викладача є центральним вузлом агрегації даних. Навігація здійснюється через бічне меню, що розділяє робочий процес на логічні етапи. Ключовим інструментом підготовчого етапу є модуль «Etalon cloud editor» (укр. Редактор еталонної хмари). Інтерфейс дозволяє викладачу не лише генерувати еталонну множину  $T^*$  з тексту, а й зберігати різні її версії для повторного використання. Унікальною функцією редактора є таблиця налаштування параметрів (Word, Base freq, Coeff), де викладач може вручну коригувати вагові коефіцієнти для специфічних термінів, посилюючи або послаблюючи їхній вплив на фінальну візуалізацію (рисунок 4.3).

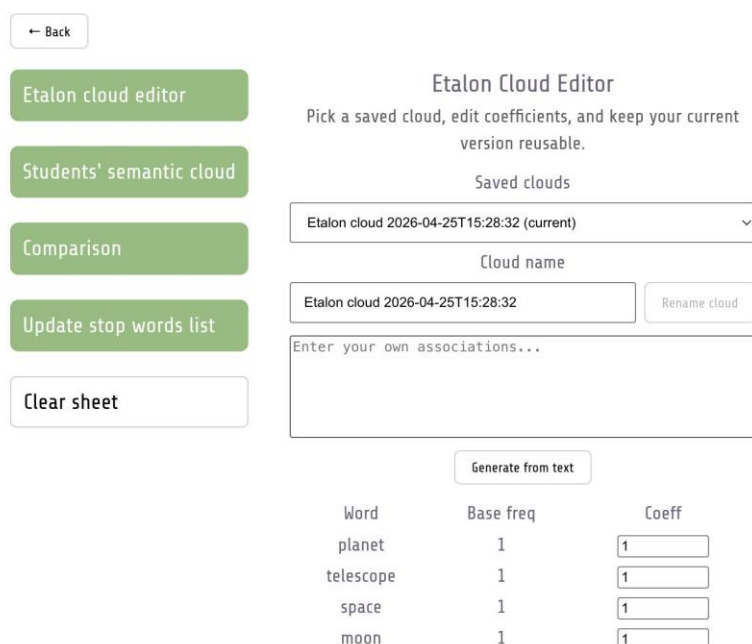


Рисунок 4.3 — Інтерфейс редактора еталонної моделі знань

Для підвищення якості лінгвістичного аналізу в системі реалізовано модуль «Update stop words list» (укр. Управління стоп-словами). Оскільки кожна тема лекції має свою специфіку, викладач може динамічно додавати або видаляти слова-паразити чи нерелевантні терміни безпосередньо під час сесії. Ці зміни автоматично кешуються системою і враховуються NLP-конвеєром при нормалізації відповідей студентів, очищуючи фінальний результат від інформаційного шуму (рисунок 4.4).

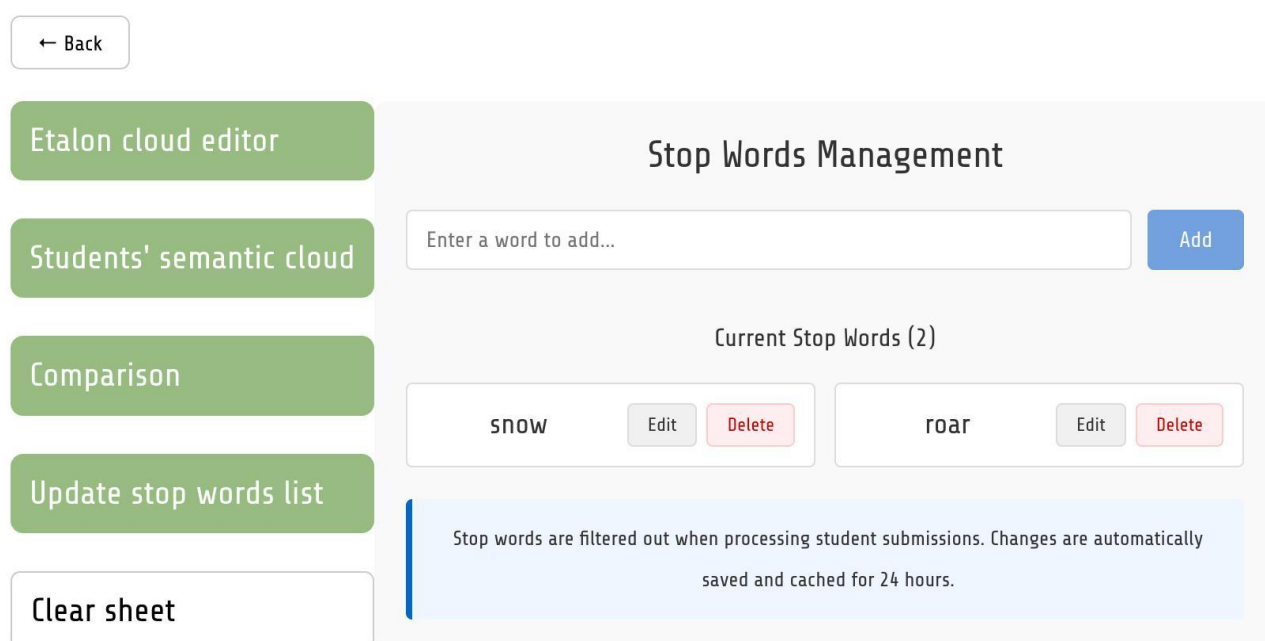


Рисунок 4.4 — Модуль динамічного управління лінгвістичними фільтрами

Робочий простір студента містить поля для введення імені та безпосередньої відповіді на запитання викладача. Система імплементує жорсткий контроль стану сесії (англ. State Management): після відправки даних інтерфейс блокує повторне введення повідомленням «You already submitted for this round...», запобігаючи дублюванню відповідей. На цьому ж екрані студент у режимі реального часу спостерігає за формуванням «Group word cloud» (укр. Колективної хмари слів групи) (рисунок 4.5).

Enter words about the subject:

Star

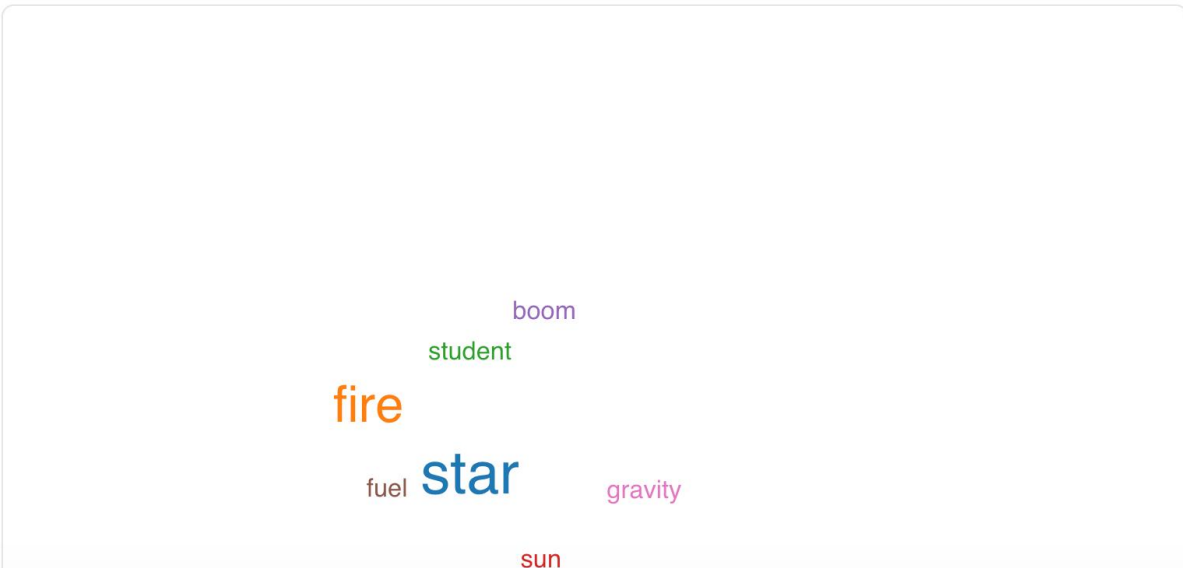
Enter your name:

Sasha

Already submitted

You already submitted for this round. Wait for the tutor to reset the session before sending again.

Group word cloud



A group word cloud visualization showing the following words: 'boom' (purple), 'student' (green), 'fire' (orange), 'fuel' (grey), 'star' (blue), 'gravity' (pink), and 'sun' (red). The word 'star' is the largest and most central, followed by 'fire'.

Рисунок 4.5 — Робочий простір студента із блокуванням повторної відправки

Під час заняття викладач контролює життєвий цикл опитування через вкладку «Students' semantic cloud» (укр. Хмара термів студентів). Інтерфейс дозволяє: бачити кількість студентів, які надали свої відповіді, а також примусово оновити дані або повністю скинути поточний раунд для переходу до наступного запитання (рисунок 4.6).

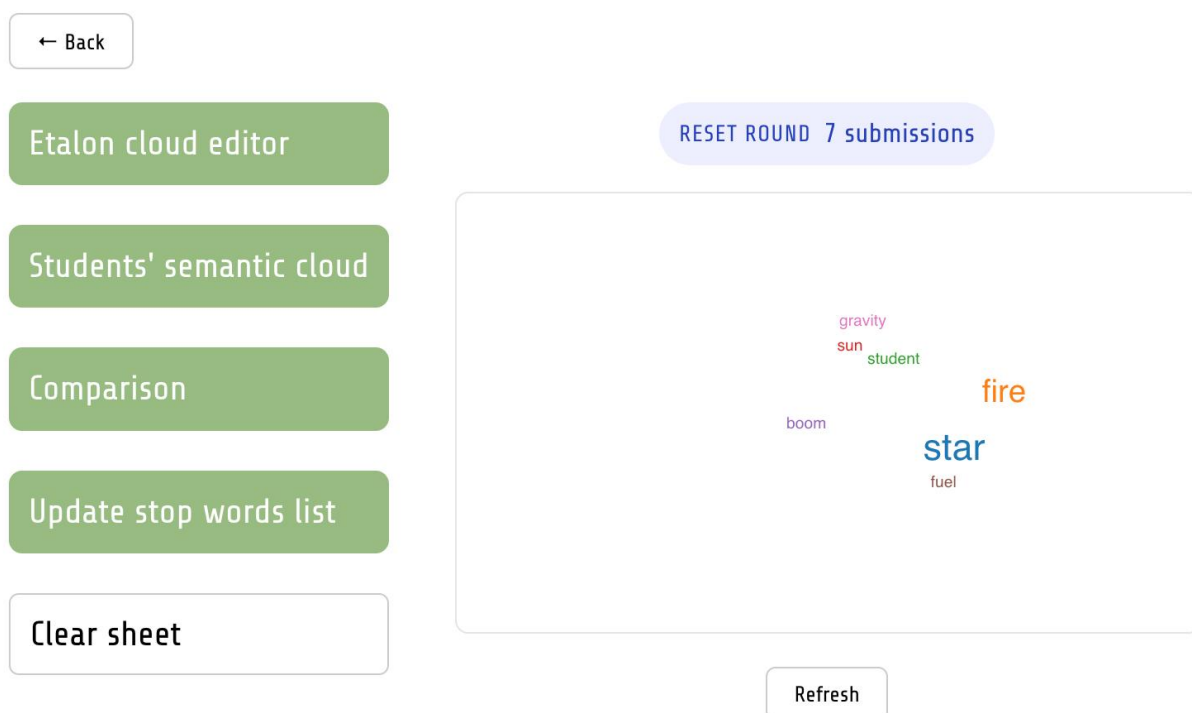


Рисунок 4.6 — Хмара термів студента в інтерфейсі викладача

Ключовий функціонал підтримки прийняття рішень реалізовано в модулі «Comparison» (укр. Порівняння). Коли вибірка стає репрезентативною, викладач ініціює алгоритмічне порівняння студентської та еталонної множин. SOLO Engine розраховує метрики семантичної подібності та виводить на екран діагностований рівень (наприклад, «Level: Prestructural») разом із розгорнутою текстовою рекомендацією (наприклад, «The answer is off-topic. The student lacks even a basic grasp of the terminology») (рисунок 4.7).

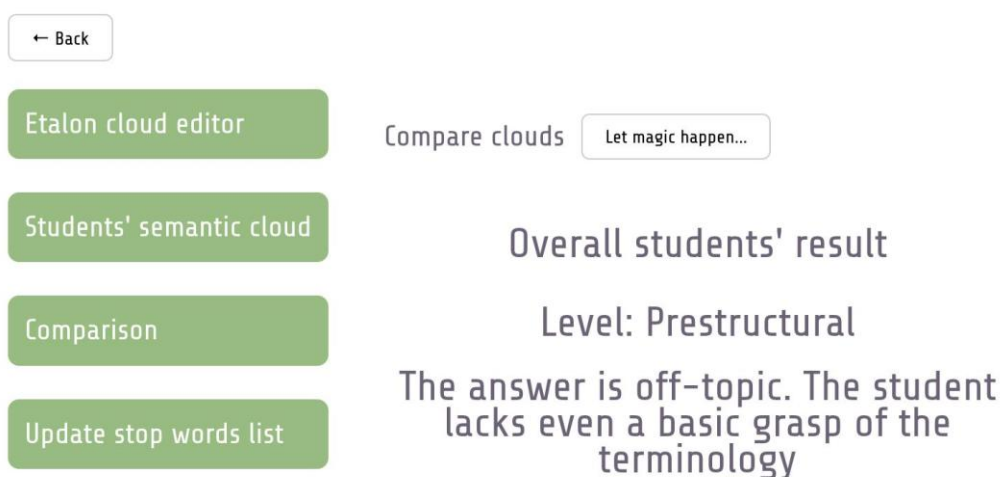


Рисунок 4.7 — Аналітичний звіт системи із зазначенням рівня за таксономією SOLO

Програмний продукт представляє собою веб-додаток, для роботи якого необхідно дотримуватись таких мінімальних системних вимог:

- Стабільне широкопasmове підключення до Інтернету.
- Останні версії одного із браузерів: Chrome, Microsoft Edge, Safari, Mozilla Firefox.
- Операційна система: Windows (останні випуски) або дистрибутиви Linux на базі Debian.
- Двоядерний процесор та 2 ГБ оперативної пам'яті.

## **4.2. Методологія проведення пілотного експерименту**

Для перевірки ефективності розробленого програмного продукту та валідації алгоритмів керованого прийняття рішень на основі даних (англ. Data-Driven Decision Making) було проведено пілотний навчальний експеримент. У дослідженні взяли участь 30 студентів. Варто зауважити, що через обмежений розмір вибірки отримані результати мають індикативний характер і не можуть вважатися статистично значущими для великих генеральних сукупностей, проте вони є достатніми для перевірки працездатності запропонованої моделі. Для забезпечення статистичної достовірності результатів вибірку було поділено на дві рівні групи по 15 осіб: контрольну, де лекція проходила за традиційною методикою, та експериментальну — із застосуванням розробленої системи. Розподіл учасників здійснювався за методом стратифікованої рандомізації: середній академічний бал в обох групах був вирівняний і становив приблизно 6,5 бала з 12 можливих. Це дозволило уникнути концентрації студентів з виключно високим або низьким рівнем підготовки в одній із груп.

Апробація системи відбувалася під час проведення лекційного заняття на тему «Утворення та еволюція зір» з предмета «Астрономія». Методологія експерименту для експериментальної групи передбачала три контрольні зрізи в реальному часі, кожен з яких діагностував певний когнітивний рівень за таксономією SOLO та вимагав від викладача відповідної адаптації матеріалу.

**Етап 1: Діагностика базового розуміння** На початку лекції студентам було поставлено запитання: *«Які основні характеристики або ресурси потрібні небесному тілу, щоб воно вважалося живою зорею, а не просто планетою чи камінням?»* Еталонна хмара викладача містила строгі наукові терміни: *маса, гравітація, гідростатична рівновага, випромінювання, термоядерний синтез, водень, плазма, світність*. Під час генерації студентської хмари алгоритмічне ядро виконувало первинну класифікацію. У відповідях переважали побутові концепти, наприклад, «вогонь», «світло», «космос»; система діагностувала Unistuctural рівень і сигналізувала викладачу про необхідність пояснення фізичної природи явищ (приклад наведений на рисунку 4.8).

The screenshot shows a web interface for 'Etalon cloud editor'. On the left, there is a vertical sidebar with buttons: '← Back', 'Etalon cloud editor', 'Students' semantic cloud', 'Comparison', 'Update stop words list', and 'Clear sheet'. The main area contains a 'Compare clouds' button with the text 'Let magic happen...'. Below this, the 'Overall students' result' is displayed, indicating a 'Level: Unistuctural'. A message states: 'Focusing on establishing interdisciplinary links is recommended. Fragmentary mastery of concepts.' Underneath, there is a 'Stats' section showing 'F1: 0.3333', 'Matches: mass, radiation', 'Precision: 0.500', and 'Recall: 0.250'.

Рисунок 4.8 — Приклад виводу Unistuctural рівня

**Етап 2: Процесна перевірка** У середині лекції, після пояснення динаміки сил всередині зорі, було ініційовано другий зріз: *«Які фізичні чинники протидіють один одному всередині зорі, забезпечуючи її стабільність?»* Очікувана еталонна множина: *гравітаційний тиск, внутрішній тиск, температура, промениста енергія, рівновага, ядро*. На цьому етапі критично важливу роль відіграв конвеєр нормалізації тексту winkNLP. Наприклад, один студент написав «сила тяжіння», а інший — «гравітаційне стискання». Завдяки використанню семантичних векторів система ідентифікувала ці словосполучення як контекстуальні синоніми до еталонного «гравітаційний тиск». Виявлення у хмарі обох протидіючих сил (і тиску, і гравітації) автоматично класифікувалося

як впевнений Multi-structural рівень (приклад наведений на рисунку 4.9).

The screenshot shows a web interface for 'Etalon cloud editor'. On the left is a vertical sidebar with buttons: '← Back', 'Etalon cloud editor', 'Students' semantic cloud', 'Comparison', 'Update stop words list', and 'Clear sheet'. The main area contains the text 'Compare clouds' and a button 'Let magic happen...'. Below this, the results are displayed: 'Overall students' result' with 'Level: Multi-structural'. A message states: 'Material structuring is recommended. Redundant disconnected terminology is present.' Under 'Stats', the F1 score is 0.4286. A list of 'Matches' includes 'gravity, pressure, temperature'. Precision is 0.375 and Recall is 0.500.

Рисунок 4.9 — Приклад виводу Multi-structural рівня

**Етап 3: Синтез та фінальне оцінювання** Наприкінці лекції оцінювалася здатність студентів до синтезу: *«Опишіть процес перетворення масивної зорі на чорну діру: що саме має статися з речовиною та структурою?»* Еталонні концепти: *залізне ядро, наднова, гравітаційний колапс, сингулярність, горизонт подій*. Для досягнення реляційного рівня студентська хмара мала відобразити логічний ланцюжок подій. Використання векторних представлень дозволило системі гнучко оцінювати нетипові формулювання. Наприклад, якщо студент вводив фразу «зоря схлопується», алгоритм зводив токен до леми «схлопуватися», обчислював косинусну відстань у векторному просторі до еталонного концепту «колапс» (з фіксацією показника семантичної близькості на рівні 0.89) і зараховував цей термін як валідний елемент знань.

Головним фактором експерименту була не просто візуалізація цих термінів, а кероване прийняття рішень. Якщо б на другому етапі система показувала низьку F1-міру (рівень Uni-structural), викладач не переходив до розповіді про чорні діри, а зупиняв лекцію для повторного пояснення механізмів гідростатичної рівноваги. У контрольній групі, де такий інструмент був відсутній, викладач не мав об'єктивного зворотного зв'язку і рухався за жорстким планом лекції, залишаючи частину аудиторії з фрагментарним розумінням матеріалу (приклад наведений на рисунку 4.10).

← Back

Etalon cloud editor

Students' semantic cloud

Comparison

Update stop words list

Clear sheet

Compare clouds Let magic happen...

### Overall students' result

Level: Relational

Transition to the next module is recommended. High semantic integrity confirmed.

Stats

F1: 0.9091

Matches: core, supernova, collapse, singularity, horizon

Precision: 0.833

Recall: 1.0

#### Рисунок 4.10 — Приклад виводу Relational рівня

Окрім впровадженого інструментарію, на фінальні показники успішності могли впливати додаткові фактори, зокрема індивідуальний стиль викладання та специфічні особливості конкретних академічних груп, що було враховано при інтерпретації даних.

### 4.3 Аналіз результатів та оцінка ефективності системи

Завершальним етапом апробації став порівняльний аналіз успішності, результати якого дозволили підтвердити ефективність розробленої системи. Для отримання об'єктивних даних та мінімізації суб'єктивного фактора («упередженості експерта») оцінювання результатів експерименту було організовано у два етапи.

Через тиждень після проведення лекції з використанням системи на наступному навчальному занятті обом групам (контрольній та експериментальній) було запропоновано виконати однакову контрольну роботу. Завдання включали як відкриті питання на знання термінології, так і розрахункові задачі на розуміння фізичних процесів еволюції зір. Важливою умовою експерименту стало залучення стороннього викладача для перевірки робіт. Перевірка здійснювалася анонімно: експерт не володів інформацією про те, до якої саме групи (контрольної чи

експериментальної) належить той чи інший студент. Такий підхід забезпечив повну неупередженість та високу достовірність отриманих балів.

Статистичний аналіз результатів цієї контрольної роботи показав, що експериментальна група продемонструвала середній бал на 13,8% вищий, ніж контрольна група. Математичне обґрунтування цього приросту базується на порівнянні сум балів, отриманих студентами: для контрольної групи сума балів склала 116 з середнім балом  $\sim 7,73$ . Тоді як для експериментальної — 132 з середнім балом  $\sim 8,8$ . Оскільки оцінювання здійснювалося за 12-бальною шкалою, такий приріст свідчить про якісний перехід групи до вищого рівня розуміння матеріалу. Реальний розподіл оцінок у групах наведено у таблиці 4.1.

Таблиця 4.1 — Розподіл оцінок за результатами контрольної роботи

Оцінка (12-бальна шкала)	Кількість учнів (Контрольна група)	Кількість учнів (Експериментальна група)
12 балів	0	1
11 балів	1	0
10 балів	1	3
9 балів	3	5
8 балів	4	4
7 балів	3	1
6 балів	2	1
3 бали	1	0
<b>СЕРЕДНІЙ БАЛ</b>	<b>7,73</b>	<b>8,8</b>

Наочний розподіл оцінок за результатами підсумкового контролю представлено на гістограмі (рисунок 4.11).

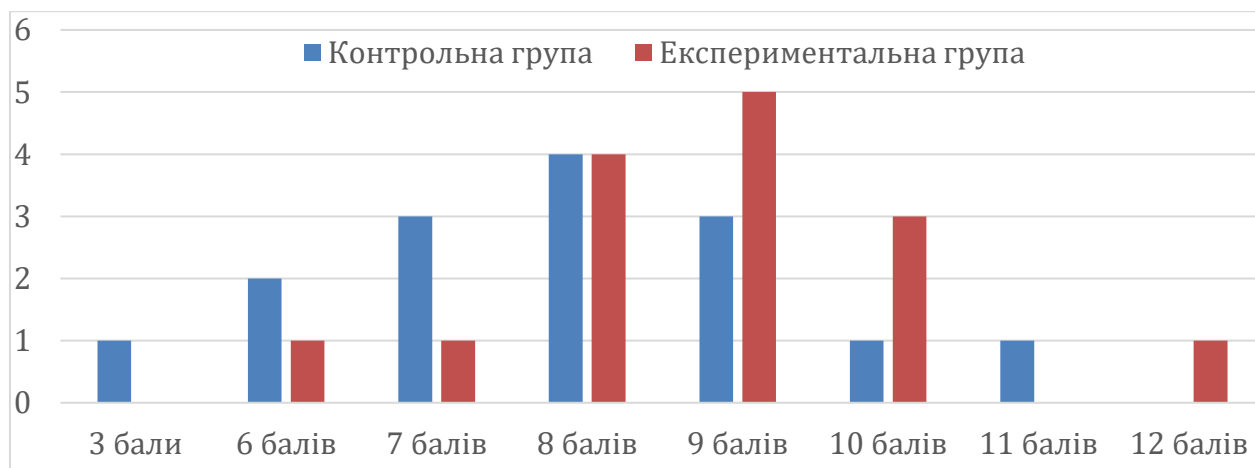


Рисунок 4.11 — Гістограма розподілу оцінок у контрольній та експериментальній групах

Основним чинником успіху стала реалізація концепції керованого прийняття рішень. Протягом експерименту було зафіксовано випадок, коли під час другого зрізу (запитання про стабільність зорі) система діагностувала рівень SOLO Unistuctural у більшості студентів. Це стало критичним сигналом для викладача: замість запланованого переходу до теми еволюції масивних зір лекцію було призупинено для додаткового роз'яснення балансу сил гравітації та внутрішнього тиску. Без використання програмного продукту цей когнітивний розрив залишився б непоміченим до етапу підсумкового контролю, що призвело б до фрагментарного розуміння всієї наступної теми.

Додатковим стимулюючим фактором виявилася активізація вищих когнітивних функцій студентів. Процес формулювання відповіді, за якого студент мав самостійно виділити ключові терміни для їхнього семантичного зв'язку з еталоном, сприяв глибшому запам'ятовуванню. Студенти не просто пасивно слухали матеріал, а здійснювали активний пошук семантичних відповідностей, що корелює з принципами конструктивістського навчання.

Під час апробації було також оцінено стійкість системи до ризиків, зокрема до спроб введення нерелевантних даних («шуму»). Аналіз логів показав, що в окремих випадках студенти вводили некоректні або жартівливі терміни. Проте інтегрований лінгвістичний конвеєр на базі бібліотеки winkNLP успішно ідентифікував та відфільтрував ці токени як стоп-слова або нерелевантні частини мови. Це дозволило зберегти чистоту колективної хмари слів та забезпечити об'єктивність підсумкових метрик Precision та Recall.

Таким чином, розроблена система доводить свою спроможність функціонувати як об'єктивний асистент викладача. Поєднання математичної точності алгоритму BM25, гнучкості семантичних векторів та навчальної глибини таксономії SOLO дозволило трансформувати традиційну лекцію на адаптивний освітній процес, що динамічно підлаштовується під актуальний рівень знань аудиторії.

Водночас варто зауважити, що отримані результати можуть залежати від додаткових факторів, зокрема індивідуального стилю викладання та специфічних

особливостей конкретної групи, що потребує врахування при подальшому масштабуванні дослідження.

## **Висновок до розділу 4**

Практична апробація розробленої системи оцінює ефективність її впровадження в освітній процес. Програмний продукт, реалізований у форматі MVP, демонструє стабільну роботу всіх ключових компонентів: від модулів авторизації та динамічного управління стоп-словами до аналітичного ядра SOLO Engine.

Аналіз користувацьких інтерфейсів підтверджує гнучкість системи. Можливість ручного коригування вагових коефіцієнтів у редакторі еталонної хмари та версіювання моделей знань дозволяють викладачу адаптувати інструментарій під специфіку будь-якої технічної чи гуманітарної дисципліни. Своєю чергою, механізм контролю стану сесій для студентів забезпечує чистоту вхідних даних, що є необхідною умовою для отримання об'єктивної аналітики.

Пілотний практичний експеримент за участю 30 студентів підтверджує гіпотезу про позитивний вплив керованого прийняття рішень на результативність навчання. Статистичний аналіз фіксує приріст середнього бала експериментальної групи на 13,8 % (8,8 бала проти 7,73 у контрольній групі) за 12-бальною шкалою. Отримані дані свідчать, що використання системи дозволяє викладачу вчасно ідентифікувати когнітивні розриви (зокрема рівень Uni-structural за таксономією SOLO) та оперативно коригувати сценарій лекції, що запобігає фрагментарному засвоєнню матеріалу.

Водночас апробація визначає межі застосування поточної версії системи. Через пілотний характер дослідження та обмежений розмір вибірки отримані результати мають орієнтовний характер. Спостереження також фіксують, що на підсумкові показники успішності, окрім використання програмного продукту,

можуть впливати додаткові чинники, такі як стиль викладання та рівень попередньої підготовки групи.

У цілому, результати експериментальної перевірки показують, що розроблена система успішно виконує функцію інтелектуального асистента викладача, забезпечуючи перехід від пасивного слухання до керованої взаємодії на основі семантичного аналізу даних у реальному часі.

## ВИСНОВКИ

У результаті виконання магістерської роботи вирішено задачу розроблення формалізованого підходу до керованого прийняття рішень у гейміфікованому цифровому середовищі на основі аналізу текстових відповідей користувачів.

У ході дослідження проаналізовано існуючі підходи до формувального оцінювання, системи аудиторного опитування та засоби візуалізації колективних відповідей. Встановлено, що наявні рішення забезпечують збір і часткову обробку даних, однак не реалізують формалізований механізм переходу від аналізу текстових відповідей до автоматизованого прийняття рішень у реальному часі, що є критичним для динамічних гейміфікованих середовищ.

Розроблено модель представлення знань користувачів у вигляді множини термінів, що формується на основі текстових відповідей після їх попередньої обробки. Для цього використано методи обробки природної мови, зокрема токенізацію, фільтрацію службових слів та лематизацію, що забезпечують уніфікацію текстових даних та підвищують коректність подальшого аналізу.

Для визначення значущості термінів обґрунтовано застосування алгоритму Окарі BM25, який враховує частоту появи терміна та нормалізацію за довжиною тексту, що є особливо важливим для аналізу коротких текстових відповідей. Для оцінювання відповідності множини термінів користувачів еталонній моделі використано асиметричні метрики точності, повноти та F-міри, що дозволяють окремо оцінити повноту охоплення знань, їх точність та узгодженість.

Розроблено формалізований алгоритм керованого прийняття рішень, який визначає відображення множини текстових відповідей у множину керуючих впливів. Алгоритм включає етапи попередньої обробки тексту, формування множини термінів, їх зважування, порівняння з еталонною моделлю та інтерпретації отриманих метрик.

Для інтерпретації результатів використано таксономію SOLO як шкалу відображення кількісних показників у якісні рівні структури знань. На її основі

реалізовано систему формалізованих правил, що встановлює відповідність між значеннями метрик і типами керуючих впливів, зокрема повторення матеріалу, уточнення понять, структуризації знань або переходу до складніших завдань.

У рамках роботи спроектовано та реалізовано програмну систему, що інтегрує запропоновані моделі та алгоритми. Система побудована за принципами клієнт-серверної архітектури з розподілом функціональності між клієнтською та серверною частинами. Серверна частина реалізована із використанням середовища виконання JavaScript та платформи Node.js і забезпечує обробку текстових даних, обчислення метрик та виконання алгоритму прийняття рішень. Клієнтська частина реалізована із використанням бібліотеки React і відповідає за взаємодію з користувачем, візуалізацію результатів та відображення хмари термінів. Обмін даними між компонентами системи здійснюється за допомогою веб-інтерфейсу прикладного програмування на основі протоколу HTTP. Для обробки тексту використано зовнішні сервіси аналізу мовних даних, що забезпечують лематизацію та нормалізацію термінів.

Проведено експериментальну перевірку ефективності запропонованого підходу на вибірці з 30 учасників. Отримано зростання результативності експериментальної групи на 13,8 %, що свідчить про потенційну ефективність застосування алгоритму керованого прийняття рішень. Водночас отримані результати мають попередній характер і потребують підтвердження на більших вибірках із застосуванням методів статистичної перевірки.

Наукова новизна отриманих результатів полягає у розробленні моделі керованого прийняття рішень, що реалізує алгоритмічний перехід від результатів аналізу текстових відповідей студентів до визначення подальших дій навчального процесу шляхом відображення кількісних показників подібності множин термінів у якісні рівні структури знань на основі формалізованих правил.

Практичне значення: створено програмну систему, що реалізує алгоритм керованого прийняття рішень та забезпечує автоматизовану обробку коротких текстових відповідей із подальшим формуванням рекомендацій для адаптації сценарію заняття.

До обмежень роботи належать використання обмеженої вибірки, спрощене представлення семантики тексту на основі множини термінів та залежність результатів від якості еталонної моделі. Запропонований підхід орієнтований на короткі текстові відповіді та може потребувати адаптації для інших типів даних.

Подальші дослідження доцільно спрямувати на розширення семантичних моделей представлення знань, автоматизацію формування еталонної моделі, застосування методів машинного навчання для оптимізації правил прийняття рішень та проведення масштабних експериментів.

Таким чином, поставлена в роботі задача вирішена, а запропонований підхід підтверджує можливість формалізації процесу керованого прийняття рішень на основі аналізу текстових відповідей та його практичної реалізації у вигляді програмної системи.

## СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. European Commission. Digital Education Action Plan 2021–2027: Resetting education and training for the digital age. Brussels, 2020.
2. UNESCO. Education 2030: Incheon Declaration and Framework for Action for the Implementation of Sustainable Development Goal 4. Paris : UNESCO, 2016.
3. Биков В. Ю., Вернигора С. М., Гуржій А. М. та ін. Проектування і використання відкритого хмароорієнтованого освітньо-наукового середовища закладу вищої освіти // Інформаційні технології і засоби навчання. 2019. Т. 74, № 6. С. 1–19.
4. Black P., Wiliam D. Assessment and classroom learning // Assessment in Education. 1998. Vol. 5, No. 1. P. 7–74.
5. Shute V. J. Focus on formative feedback // Review of Educational Research. 2008. Vol. 78, No. 1. P. 153–189.
6. Brooks B. J., Gilbuena D. M., Krause S. J., Koretsky M. D. Using word clouds for fast, formative assessment of students' short written responses // Chemical Engineering Education. 2014. Vol. 48, No. 4. P. 190–198.
7. Kitchens M. B. Word clouds: An informal assessment of student learning // College Teaching. 2014. Vol. 62, No. 3. P. 113–114.
8. Atenstaedt R. Word cloud analysis of the BJGP: 5 years on // British Journal of General Practice. 2017. Vol. 67, No. 658. P. 231–232.
9. Corbett A. T., Anderson J. R. Knowledge tracing: Modeling the acquisition of procedural knowledge // User Modeling and User-Adapted Interaction. 1994. Vol. 4. P. 253–278.
10. Piech C. та ін. Deep knowledge tracing // Advances in Neural Information Processing Systems. 2015. P. 505–513.
11. Embretson S. Item response theory for psychologists // Quality of Life Research. 2000.
12. Rasch G. Probabilistic models for some intelligence and attainment tests. Copenhagen : Danish Institute for Educational Research, 1960.

13. Blei D. M., Ng A. Y., Jordan M. I. Latent Dirichlet allocation // Journal of Machine Learning Research. 2003. Vol. 3. P. 993–1022.
14. Biggs J. B., Collis K. F. Evaluating the Quality of Learning: The SOLO Taxonomy (Structure of the Observed Learning Outcome). New York : Academic Press, 1982. 245 p.
15. Галицька-Дідух Т. В., Гречановська О. В., Батарейна І. О., Гурська В. А. Гейміфікація як засіб індивідуалізації навчання XXI століття. Київ : Futurity Publishing, 2025. 184 с.
16. Namari J., Koivisto J., Sarsa H. Does gamification work? A literature review of empirical studies on gamification // Proceedings of the 47th Hawaii International Conference on System Sciences (HICSS). 2014. P. 3025–3034.
17. Chi M. T. H., Wylie R. The ICAP framework: Linking cognitive engagement to active learning outcomes // Educational Psychologist. 2014. Vol. 49, No. 4. P. 219–243.
18. Kotlyar S., Levchenko O., Balaban O., Tkachenko H. Media Systems as a Communicative Model of Current Educational Practices // International Journal of Computer Science and Network Security. 2022. Vol. 22, No. 1. P. 125–132.
19. Mand S. K., Cico S. J., Haas M. R. C. [та ін.]. Let's get active: The use of technology-enhanced audience interaction to promote active learning // AEM Education and Training. 2024. Vol. 8, Suppl. 1. P. S50–S55.
20. Kahoot! AI integration and Deep Learning applications in classroom management / Kahoot! Official Blog. 2024. URL: <https://kahoot.com/blog/ai-integration/> (дата звернення: 05.04.2026).
21. Using Mentimeter for Eliciting the Students' Responses in Formative Assessment Practice in Jombang // Journal of English Language Teaching. 2024. Vol. 11, No. 2. P. 85–95.
22. What is Mentimeter? Platforms for interactive engagement / University of Bath Teaching Hub. 2023. URL: <https://teachinghub.bath.ac.uk/mentimeter/> (дата звернення: 06.04.2026).

23. Slido AI Summaries and automation features in town-hall meetings / Slido Blog. 2024. URL: <https://blog.slido.com/ai-summaries/> (дата звернення: 07.04.2026).
24. Coaldrake K. What are the best real-time polling and quiz tools to use for class interaction? / EdTech Review. 2023. URL: <https://edtechreview.in/polling-tools/> (дата звернення: 07.04.2026).
25. ClassInsight: Developing an Agentic AIED Tool for Responsive Teaching and Learning // Proceedings of the International Conference on Artificial Intelligence in Education. 2024. P. 112–120.
26. Tips and Trends: Microsoft Forms: Interactive Polling for Presentations and Instruction / ACRL Instruction Section. 2023. URL: <https://acrl.ala.org/IS/instruction-tools/> (дата звернення: 08.04.2026).
27. The 10 Best AI Question Generators for Quizzes / Vevox. 2024. URL: <https://www.vevox.com/blog/ai-question-generators> (дата звернення: 08.04.2026).
28. Interactive surveys during online lectures for IT specialties / Zhytomyr Polytechnic State University study // Вісник Житомирського державного технологічного університету. Серія: Технічні науки. 2023. № 2. С. 45–52.
29. Hearst M. A., Pedersen E., Patil L., Lee E., Laskowski P., Franconeri S. An Evaluation of Semantically Grouped Word Cloud Designs // IEEE Transactions on Visualization and Computer Graphics. 2019. Vol. 26, No. 9. P. 2796–2810.
30. Nault K., Nieto-Taborda M., Ruhi U. Linking ICAP and Bloom's Taxonomy for formatively assessing and actively engaging students in e-learning environments // Computers & Education. 2025. Vol. 195. 104712.
31. Manning C. D., Raghavan P., Schütze H. Introduction to Information Retrieval. Cambridge : Cambridge University Press, 2008. 482 p.
32. Filatova O., Hearst M. A. Evaluating Word Clouds as an Analytical Tool // Journal of Information Visualization. 2020. Vol. 19, No. 4. P. 250–265.
33. Jurafsky D., Martin J. H. Speech and Language Processing. 3rd ed. Stanford : Stanford University, 2023. 624 p.

34. WinkJS. winkNLP: Developer friendly Natural Language Processing. URL: <https://winkjs.org/> (дата звернення: 03.04.2026).
35. Short-Text Semantic Similarity (STSS): Techniques, Challenges, and Future Directions // Applied Sciences. 2023. Vol. 13, No. 5. 3205.
36. Term Weighting for Feature Extraction on Twitter: A Comparison Between BM25 and TF-IDF // Proceedings of the International Conference on Data Mining (ICDM). 2019. P. 450–458.
37. Practical BM25 - Part 2: The BM25 Algorithm and its Variables / Elastic Blog. URL: <https://www.elastic.co/blog/practical-bm25-part-2-the-bm25-algorithm-and-its-variables> (дата звернення: 07.04.2026).
38. Munzner T. Visualization Analysis and Design. Boca Raton : A K Peters/CRC Press, 2014. 428 p.
39. Han J., Kamber M., Pei J. Data Mining: Concepts and Techniques. 3rd ed. Waltham : Morgan Kaufmann, 2011. 744 p.
40. Viégas F. B., Wattenberg M., Feinberg J. Participatory Visualization with Wordle // IEEE Transactions on Visualization and Computer Graphics. 2009. Vol. 15, No. 6. P. 1137–1144.
41. Experimental study on short-text clustering using transformer-based semantic similarity measure // PeerJ Computer Science. 2023. URL: <https://peerj.com/articles/cs-2078/> (дата звернення: 08.04.2026).
42. O'Connor B. T. F-scores, Dice, and Jaccard set similarity / AI and Social Science. 2012. URL: <https://brenocon.com/blog/2012/04/f-scores-dice-and-jaccard-set-similarity/> (дата звернення: 08.04.2026).
43. Accuracy, Precision, Recall or F1? / Towards Data Science. 2020. URL: <https://towardsdatascience.com/accuracy-precision-recall-or-f1-331bfeb326fc> (дата звернення: 08.04.2026).
44. Evaluation metrics and statistical tests for machine learning // PubMed Central (PMC). 2024. URL: <https://pmc.ncbi.nlm.nih.gov/articles/PMC10937649/> (дата звернення: 08.04.2026).

45. Vite: Next Generation Frontend Tooling. URL: <https://vitejs.dev/> (дата звернення: 11.04.2026).
46. D3-cloud: Word cloud layout in JavaScript / Jason Davies. URL: <https://github.com/jasondavies/d3-cloud> (дата звернення: 11.04.2026).
47. Express - Fast, unopinionated, minimalist web framework for Node.js. URL: <https://expressjs.com/> (дата звернення: 11.04.2026).
48. Google Sheets API Overview / Google for Developers. URL: <https://developers.google.com/sheets/api> (дата звернення: 11.04.2026).
49. Firebase Admin SDK - Introduction / Google for Developers. URL: <https://firebase.google.com/docs/admin/setup> (дата звернення: 11.04.2026).
50. Virtual DOM and Internals / React Docs. URL: <https://legacy.reactjs.org/docs/faq-internals.html> (дата звернення: 11.04.2026).
51. Reconciliation / React Core Algorithm Concepts. URL: <https://legacy.reactjs.org/docs/reconciliation.html> (дата звернення: 11.04.2026).
52. Sidorov D. Analyzing virtual DOM performance in front-end frameworks // Наука і техніка сьогодні. Серія «Техніка». 2024. № 8(36). С. 840–858. DOI: 10.52058/2786-6025-2024-8(36)-840-858.
53. React Fiber Architecture and Concurrency Patterns / Andrew Clark. URL: <https://github.com/acdlite/react-fiber-architecture> (дата звернення: 13.04.2026).
54. Tilkov S., Vinoski S. Node.js: Using JavaScript to Build High-Performance Network Programs // IEEE Internet Computing. 2010. Vol. 14, No. 6. P. 80–83.
55. Node.js Internal Architecture: Libuv and the Event Loop / Node.js Documentation. URL: <https://nodejs.org/en/docs/guides/event-loop-timers-and-nexttick/> (дата звернення: 12.04.2026).
56. Gupta R. K., Panchal Y. Building Scalable Web Apps with Node.js and Express: Design and develop a robust, scalable, high-performance web application using Node.js, Express.js, TypeScript, and Redis. New Delhi : Orange Education Pvt Ltd, 2024. 390 p.
57. JSON Web Token (JWT) / OWASP Cheat Sheet Series. URL: <https://cheatsheetseries.owasp.org/> (дата звернення: 13.04.2026).

58. Firebase Admin SDK: Verify ID Tokens documentation / Google for Developers. URL: <https://firebase.google.com/docs/auth/admin/verify-id-tokens> (дата звернення: 13.04.2026).
59. Gunawardena B. JWTs & CSRF Tokens: When and Where to use JSON Web Tokens / Medium. 2023. URL: <https://medium.com/@gunawardena.buddika/jwts-csrf-tokens-465e5d4f91cf> (дата звернення: 15.04.2026).
60. Cross-Site Request Forgery Prevention / OWASP Foundation. URL: <https://owasp.org/www-community/attacks/csrf> (дата звернення: 13.04.2026).

# **ДОДАТОК А**

## **ХМАРИ ТЕРМІНІВ ЯК МЕТОД ФОРМУВАЛЬНОГО АНАЛІЗУ ЗНАНЬ У СЕРВІСНО-ОРІЄНТОВАНИХ ОСВІТНІХ СИСТЕМАХ**

Стаття у фаховому технічному журналі  
Таврійський науковий вісник. Серія: Технічні науки №2 2026

Аркушів 25

Київ 2026

УДК 378.147:004.89

В. О. Тихоход  
кандидат технічних наук  
доцент кафедри цифрових технологій в енергетиці  
Національний технічний університет України  
"Київський політехнічний інститут імені Ігоря Сікорського"  
ORCID: 0000-0002-1525-4770

О. В. Шпадківський  
магістр кафедри цифрових технологій в енергетиці  
Національний технічний університет України  
"Київський політехнічний інститут імені Ігоря Сікорського"  
ORCID: 0009-0001-5460-0672

## **ХМАРИ ТЕРМІНІВ ЯК МЕТОД ФОРМУВАЛЬНОГО АНАЛІЗУ ЗНАНЬ У СЕРВІСНО-ОРІЄНТОВАНИХ ОСВІТНІХ СИСТЕМАХ**

*У дослідженні запропоновано метод формувального аналізу знань і адаптації навчального сценарію у сервісно-орієнтованих освітніх системах, що ґрунтується на агрегуванні коротких текстових відповідей студентів у вигляді термінологічного представлення академічної групи. Метод орієнтований на використання в гейміфікованих цифрових освітніх середовищах вищих навчальних закладів і забезпечує отримання оперативного аналітичного сигналу для прийняття рішень у реальному або квазі-реальному часі.*

*Теоретико-методологічною основою дослідження є концепція переходу від моделі «освіта як продукт» (Education-as-a-Product) до моделі «освіта як сервіс» (Education-as-a-Service, EaaS), у межах якої агреговане термінологічне представлення розглядається як функціональний елемент сервісної архітектури навчання. Хмари термінів у запропонованому підході використовуються як*

*аналітичний інструмент агрегування колективних відповідей, що дозволяє діагностувати домінуючі поняття та поточний стан структури знань академічної групи.*

*Для забезпечення відтворюваності та наукової валідності метод спроектовано з використанням парадигми *Design Science Research (DSR)*. Інтерпретація результатів формульованого аналізу здійснюється із застосуванням таксономії *SOLO* для оцінювання якості структури знань та рамки *ICAP* для аналізу характеру когнітивної взаємодії студентів. Отримані результати створюють теоретичне підґрунтя для алгоритмізації методу та подальшої емпіричної валідації адаптивних гейміфікованих сценаріїв навчання в цифрових освітніх системах.*

**Ключові слова:** *вища освіта, гейміфікація, хмара термінів, освіта як сервіс, Education-as-a-Service, формульоване оцінювання, навчальна аналітика, дослідження на основі наукового проектування, Design Science Research, таксономія SOLO, рамка ICAP.*

V. O. Tykhokhod

PhD in Technical Sciences

Associate Professor at the Department of Digital Technologies in Energy  
National Technical University of Ukraine “Igor Sikorsky Kyiv Polytechnic  
Institute”

ORCID: 0000-0002-1525-4770

О. В. Шпадіківський

Master, Department of Digital Technologies in Energy  
National Technical University of Ukraine “Igor Sikorsky Kyiv Polytechnic  
Institute”

ORCID: 0009-0001-5460-0672

# WORD CLOUDS AS A METHOD OF FORMATIVE KNOWLEDGE ANALYSIS IN SERVICE-ORIENTED EDUCATIONAL SYSTEMS

The study proposes a method for formative knowledge analysis and learning scenario adaptation in service-oriented educational systems, based on the aggregation of students' short textual responses into a terminological representation of the academic group. The method is designed for use in gamified digital educational environments of higher education institutions and provides a real-time or quasi-real-time operational analytical signal for decision-making.

The theoretical and methodological foundation of the research is the concept of transitioning from the "Education-as-a-Product" model to the "Education-as-a-Service" (EaaS) model, within which the aggregated terminological representation is considered a functional element of the service architecture of learning. In the proposed approach, word clouds are used as an analytical tool for aggregating collective responses, allowing for the diagnosis of dominant concepts and the current state of the academic group's knowledge structure.

To ensure reproducibility and scientific validity, the method was designed using the Design Science Research (DSR) paradigm. The interpretation of the formative analysis results is carried out using the SOLO taxonomy to assess the quality of the knowledge structure and the ICAP framework to analyze the nature of students' cognitive engagement. The obtained results create a theoretical foundation for the algorithmization of the method and the subsequent empirical validation of adaptive gamified learning scenarios in digital educational systems.

**Keywords:** higher education, gamification, word cloud, education as a service, Education-as-a-Service, formative assessment, learning analytics, Design Science Research, SOLO taxonomy, ICAP framework.

## Постановка проблеми

Сучасна вища освіта функціонує в умовах цифрової трансформації, що зумовлює зміну логіки проектування освітніх процесів та інформаційних систем. Традиційна модель «освіта як продукт» передбачає створення відносно статичних навчальних курсів із фіксованою структурою та відкладеним підсумковим оцінюванням. Такий підхід обмежує можливості оперативної адаптації навчального сценарію в умовах динамічного інформаційного середовища.

Перехід до сервісної парадигми (Education-as-a-Service, EaaS) передбачає трактування освітнього процесу як сервісно-орієнтованої інформаційної системи, у якій рішення щодо адаптації навчального сценарію ґрунтуються на даних, що надходять у реальному або квазі-реальному часі [1–3]. У межах такої архітектури ключову роль відіграють механізми формувального оцінювання та навчальної аналітики, здатні забезпечувати безперервний зворотний зв'язок і підтримувати обґрунтовані управлінські рішення.

Водночас ефективність адаптивного управління безпосередньо залежить від валідності метрик, що використовуються як індикатори навчальної активності. Одним із джерел таких метрик є гейміфікація освітнього процесу, яка розглядається як інструмент підвищення залученості студентів [4-6]. Проте поверхнєве використання ігрових механік без інтеграції їх у когнітивну структуру навчання породжує системні ризики: ефект надмірного виправдання, спотворення мотиваційної структури та «геймінгування» індикаторів успішності [4]. У таких умовах поведінкові метрики не відображають якості сформованих знань і можуть призводити до хибних управлінських рішень.

Таким чином, у сервісно-орієнтованих цифрових освітніх системах виникає суперечність між потребою в оперативному, інтерпретованому та обчислювально ефективному механізмі формувального аналізу знань і відсутністю формалізованої моделі такого механізму, інтегрованої у замкнений контур адаптації навчального сценарію.

Розв'язання цієї суперечності потребує інструмента, який одночасно стимулює когнітивну активність студентів, агрегує результати цієї діяльності в

інтерпретоване представлення та придатний до інтеграції в сервісну архітектуру цифрової платформи.

### **Аналіз останніх досліджень і публікацій**

Аналіз сучасних цифрових індустрій засвідчує стійку тенденцію переходу від продуктових моделей створення цінності до сервісних [1–3]. У продуктивній парадигмі цифровий артефакт розглядається як завершений результат інженерної діяльності, життєвий цикл якого значною мірою обмежується моментом постачання кінцевому користувачеві. Натомість перехід до сервісної моделі змінює логіку проектування: програмний продукт трансформується в довготривалу послугу, що функціонує в режимі постійної взаємодії з користувачем, накопичує експлуатаційні дані та адаптується до змін контексту використання. У таких системах ключовими стають механізми моніторингу, аналітики поведінки та швидких циклів зворотного зв'язку, які забезпечують ітеративне вдосконалення сервісу. Показовим прикладом такої трансформації є ігрова індустрія, де перехід від моделі «гра як продукт» до «гра як сервіс» зумовив зміну архітектури ігор, способів взаємодії з користувачами та підходів до аналізу даних [7]. Важливою характеристикою сервісних цифрових систем є наявність інтерпретованих проміжних представлень стану користувача, які дозволяють ухвалювати рішення без застосування складних прогнозних моделей і фактично виконують роль інтерфейсу між даними та рішеннями.

Аналогічні процеси спостерігаються й у вищій освіті, де традиційна модель «освіта як продукт» поступово втрачає ефективність у цифровому середовищі [1, 2]. Концепція «освіта як сервіс» (Education-as-a-Service, EaaS) пропонує розглядати освітній процес як сервісну інформаційну систему [3], орієнтовану на безперервну взаємодію зі здобувачами освіти та підтримувану цифровими платформами, які збирають дані про активність студентів, аналізують їх і використовують результати для адаптації сценаріїв навчальної взаємодії [1, 3]. У межах EaaS ключовими компонентами виступають модулі навчальної аналітики, механізми формувального оцінювання та інтерфейси підтримки прийняття рішень. На відміну від традиційних підсумкових форм контролю, формувальне оцінювання у сервісній

моделі виконує функцію безперервного «сенсора» стану навчальної системи, що дає змогу викладачеві оперативно коригувати перебіг заняття на основі аналітичного зворотного зв'язку [8, 9].

У цьому контексті значну увагу приділено гейміфікації як підходу до підвищення залученості студентів і підтримки активної взаємодії у цифровому середовищі [4–6]. Водночас дослідження підкреслюють, що поверхнєве застосування ігрових атрибутів (бали, бейджі, таблиці лідерів) без інтеграції у когнітивну структуру навчання не забезпечує сталого освітнього ефекту та може мати негативні наслідки. Зокрема описуються ризики ефекту надмірного виправдання, за якого зовнішні стимули витісняють внутрішню мотивацію, а також маніпулятивна адаптація поведінки до формальних правил оцінювання, коли дії оптимізуються під кількісні показники успішності, що призводить до спотворення даних навчальної аналітики та зниження обґрунтованості рішень щодо адаптації навчального сценарію [4]. Окремо відзначається, що надмірна орієнтація на конкурентні механіки може підвищувати тривожність і знижувати залученість окремих груп студентів, що суперечить принципам інклюзивності та довготривалої взаємодії, критичним для сервісних освітніх систем. Унаслідок цього більш перспективним вважається підхід конструктивної гейміфікації, у межах якого ігрові механіки інтегруються в зміст і когнітивну структуру завдань, зміщуючи фокус із зовнішніх винагород на процеси аналізу, синтезу та рефлексії.

Формувальне оцінювання у сучасних дослідженнях визначається як процес отримання та використання інформації про поточний стан навчальної діяльності з метою її корекції та підвищення ефективності навчання в процесі його здійснення [8, 9]. На відміну від підсумкового оцінювання, воно орієнтоване на підтримку навчальної діяльності в реальному часі та прийняття педагогічних рішень на основі оперативного аналітичного зворотного зв'язку. У цифрових освітніх середовищах це підсилює роль навчальної аналітики як засобу агрегування та інтерпретації даних про діяльність студентів, а також актуалізує вимогу до інструментів, які одночасно є обчислювально ефективними, інтерпретованими та придатними для використання під час заняття.

Окремий напрям досліджень стосується використання хмар слів/термінів у навчанні. Проведений огляд свідчить, що наявні роботи описують переважно дидактичні або інструментальні аспекти, не формалізуючи хмари термінів як відтворений аналітичний компонент сервісно-орієнтованого контуру адаптації навчального сценарію. У проаналізованих джерелах застосування хмар слів у навчанні узагальнюється до трьох типових сценаріїв. Перший сценарій — оперативний зворотний зв'язок у формульованому оцінюванні, коли хмара слів використовується як швидкий спосіб узагальнення коротких письмових відповідей і подальшого обговорення ключових понять у групі [10,11]. Другий сценарій — дидактичне застосування як інструмента навчальної активності (активація попередніх знань, підсумування, рефлексія, фіксація термінології) у межах окремих занять або тем, де акцент робиться на організаційній стороні використання інструмента без формалізованого контуру прийняття рішень [10, 11]. Третій сценарій — критичний аналіз обмежень хмар слів, зокрема втрату контексту, чутливість до якості тексту та ризику спрощення інтерпретації, що підкреслює потребу процедур передобробки й обережного трактування результатів [12]. У сукупності ці роботи підтверджують прикладну корисність хмар термінів як інтерпретованого представлення колективних відповідей, але не пропонують їх системної інтеграції в сервісний контур адаптації.

Разом із прикладними інструментами в освітній аналітиці активно розвиваються формалізовані параметризовані та латентні моделі, що забезпечують статистичне вимірювання або прогнозування результатів навчання. До таких підходів належать моделі відстеження засвоєння знань (knowledge tracing), що моделюють ймовірність оволодіння навичкою на основі послідовності відповідей студента та призначені для оцінювання латентних станів і прогнозування майбутньої успішності [13, 14]. Іншим класом є психометричні моделі теорії відповіді на завдання (IRT) та модель Раша, які дозволяють оцінювати параметри складності завдань і латентні характеристики здобувачів освіти на основі ймовірнісних моделей вимірювання [15, 16], проте передбачають калібрування параметрів і наявність стандартизованих вибірок даних. Для аналізу відкритих

текстових відповідей застосовуються методи тематичного моделювання, зокрема латентний розподіл Діріхле (Latent Dirichlet Allocation, LDA) [17], однак такі підходи зазвичай потребують достатнього корпусу текстів і додаткової інтерпретації, що ускладнює оперативне використання в межах заняття. Отже, попри високу прогностичну та вимірну ефективність формалізованих моделей, їх застосування в межах окремого заняття в режимі квазіреального часу є методично та технічно обмеженим. У цьому контексті хмари термінів залишаються інструментом, що поєднує інтерпретованість, обчислювальну простоту та придатність до оперативного використання в сервісно-орієнтованому контурі адаптації.

Водночас у проаналізованих публікаціях не виявлено опису використання хмар термінів як аналітичного компонента сервісно-орієнтованого замкненого контуру, що передбачає агрегування відповідей, їх інтерпретацію та ухвалення рішень щодо адаптації сценарію. Такий контур не формалізовано як відтворювану процедуру та не забезпечено узгодженим підходом до інтерпретації агрегованого сигналу в логіці адаптації навчального сценарію.

Таким чином, наукова прогалина полягає у відсутності систематизованого методу інтеграції хмар термінів у замкнений контур формуального аналізу знань із визначеними принципами інтерпретації та сценарної адаптації. Це зумовлює необхідність розроблення відповідного методу в межах сервісно-орієнтованої архітектури освіти.

### **Мета дослідження**

Метою дослідження є розроблення та теоретико-методологічне обґрунтування методу формуального аналізу знань і адаптації навчального сценарію в сервісно-орієнтованих освітніх системах на основі агрегованого термінологічного представлення колективних текстових відповідей студентів.

У межах сервісної парадигми освіти (Education-as-a-Service, EaaS) запропонований підхід розглядається як функціональний компонент замкненого аналітичного контуру, у якому результати навчальної взаємодії агрегуються,

інтерпретуються та використовуються для прийняття рішень щодо подальшої адаптації навчального сценарію.

Досягнення поставленої мети передбачає формалізацію ролі хмар термінів у сервісно-орієнтованій архітектурі цифрового освітнього середовища, визначення теоретично обґрунтованих підходів до інтерпретації агрегованого сигналу та обґрунтування методологічної рамки Design Science Research (DSR) для проєктування й оцінювання відповідного освітнього артефакта [18, 19].

### **Основна частина**

**Метод формувального аналізу знань на основі хмар термінів.** Для оперативної адаптації навчального сценарію в цифрових освітніх середовищах необхідний метод формувального аналізу знань, що забезпечує отримання інтерпретованого аналітичного сигналу про актуальний стан засвоєння навчального матеріалу академічною групою без порушення динаміки заняття. Такий сигнал має формуватися на основі результатів навчальної діяльності студентів, бути стійким до мовного шуму та нерівномірної участі, а також придатним для використання у реальному або квазі-реальному часі.

У межах цього дослідження пропонується метод формувального аналізу знань і адаптації навчального сценарію, що ґрунтується на агрегуванні коротких текстових відповідей студентів у вигляді термінологічного представлення навчальної групи. Під хмарою термінів розуміється агреговане представлення нормалізованих термінів предметної області, виділених із колективних текстових відповідей студентів, яке використовується як аналітичний сигнал у контурі формувального оцінювання. Надалі хмара термінів розглядається як форма представлення агрегованого аналітичного сигналу, отриманого внаслідок обробки текстових відповідей.

Сутність методу полягає у перетворенні множини індивідуальних коротких відповідей на узагальнене термінологічне представлення, що відображає домінантні поняття, актуалізовані академічною групою під час виконання навчального завдання. Отриманий сигнал використовується для діагностики

поточного стану структури знань групи та підтримки прийняття педагогічних рішень щодо подальшої адаптації навчального сценарію.

Таким чином, хмара термінів у запропонованому підході інтерпретується не як засіб візуалізації, а як інтерпретований аналітичний модуль у структурі сервісно-орієнтованої освітньої інформаційної системи.

Хмара термінів у межах запропонованого підходу реалізується як обчислювальна модель низької складності, призначена для агрегування колективних когнітивних сигналів. Обчислення ваг термінів та їх агрегування здійснюються з лінійною складністю відносно обсягу вхідного тексту, що забезпечує обчислювальну ефективність і придатність до інтеграції в цифрові платформи без значного навантаження на ресурси.

На відміну від параметризованих та латентних моделей навчальної аналітики [13–17], запропонований підхід не потребує попереднього навчання або калібрування параметрів і не оперує латентними змінними, що забезпечує прозорість та безпосередню інтерпретованість результату в межах заняття.

Модель описується як функція перетворення множини неструктурованих текстових входів у впорядковане представлення термінів із відповідними ваговими коефіцієнтами. Попри концептуальну простоту, такий підхід дозволяє отримати інформаційно насичене представлення стану колективного розуміння навчального матеріалу без застосування складних методів машинного навчання.

Ключовою перевагою хмар термінів є їх інтерпретованість: кожен елемент агрегованого представлення має чітке семантичне значення, що дає змогу викладачеві співвіднести результат із навчальними цілями та очікуваними результатами навчання.

Для забезпечення коректності агрегування необхідно враховувати вплив орфографічних помилок, синонімії та варіативності граматичних форм. У межах запропонованого підходу передбачаються базові процедури передобробки тексту, зокрема:

- нормалізація регістру;
- лематизація;

- уніфікація синонімів;
- вилучення стоп-слів;
- контроль мінімальної частки участі студентів.

Такі процедури спрямовані на зниження ризику штучної фрагментації понять і підвищення надійності агрегованого сигналу.

Водночас важливо враховувати контекст виконання завдання (етап заняття, тип питання, обмеження часу), оскільки інтерпретація однакової хмари термінів може відрізнитися залежно від ситуації навчальної взаємодії.

У межах запропонованого підходу формувальний аналіз знань визначається як ітеративна процедура інтерпретації агрегованих результатів навчальної діяльності студентів із метою оперативного коригування навчального сценарію. Такий аналіз орієнтований не на підсумкове вимірювання досягнень, а на виявлення поточного стану структури знань і характеру когнітивної взаємодії.

Сформоване агреговане термінологічне представлення використовується як інформаційна підстава для прийняття рішення щодо адаптації сценарію в реальному або квазі-реальному часі. Зокрема, викладач може:

- змінити темп подачі матеріалу;
- скоригувати акценти пояснення;
- ініціювати дискусію щодо домінантних або проблемних термінів;
- запропонувати додаткове завдання для уточнення розуміння.

У цьому сенсі хмара термінів виступає компонентом підтримки прийняття рішень, інтегрованим у навчальну взаємодію через інтерфейс цифрової платформи.

Прийняття рішень може здійснюватися в режимі підтримки викладача або бути частково автоматизованим. Тригери переходів у контурі адаптації визначаються як умови, сформовані на основі інтерпретації агрегованого сигналу, тоді як правила адаптації встановлюють відповідність між інтерпретованим станом знань групи та допустимими сценарними діями.

Оперативність побудови хмари термінів забезпечує мінімальну затримку в контурі зворотного зв'язку, що є критично важливим для ефективності формуального оцінювання в сервісній моделі навчання [1–3].

## **Контур інтеграції методу в сервісно-орієнтоване цифрове середовище.**

Інтеграція хмар термінів у цифрове освітнє середовище здійснюється в межах замкненого контуру формувального аналізу та адаптації навчального сценарію. Під замкненим контуром у цьому дослідженні розуміється ітеративна структура зі зворотним зв'язком, у межах якої результати інтерпретації агрегованого аналітичного сигналу впливають на перебіг наступного циклу навчальної взаємодії та визначають відповідні сценарні рішення.

У запропонованому підході короткі текстові відповіді студентів, отримані в межах навчального завдання, передаються до аналітичного модуля цифрової платформи. Платформа забезпечує збір, збереження та первинну обробку відповідей, після чого виконується передобробка тексту, нормалізація термінів і їх агрегування з формуванням агрегованого термінологічного представлення. Отримане представлення функціонує як інтерпретований аналітичний сигнал, що відображає домінуючі поняття та структуру актуалізованих знань академічної групи.

Логіку замкненого контуру та розподіл ролей між учасниками процесу подано на рисунку 1. Аналітичний модуль здійснює перетворення множини індивідуальних відповідей на агрегований сигнал, тоді як його інтерпретація відбувається з урахуванням дидактичного контексту заняття. У межах сервісно-орієнтованої архітектури цей модуль функціонує як проміжний елемент між даними навчальної взаємодії та сценарними рішеннями.

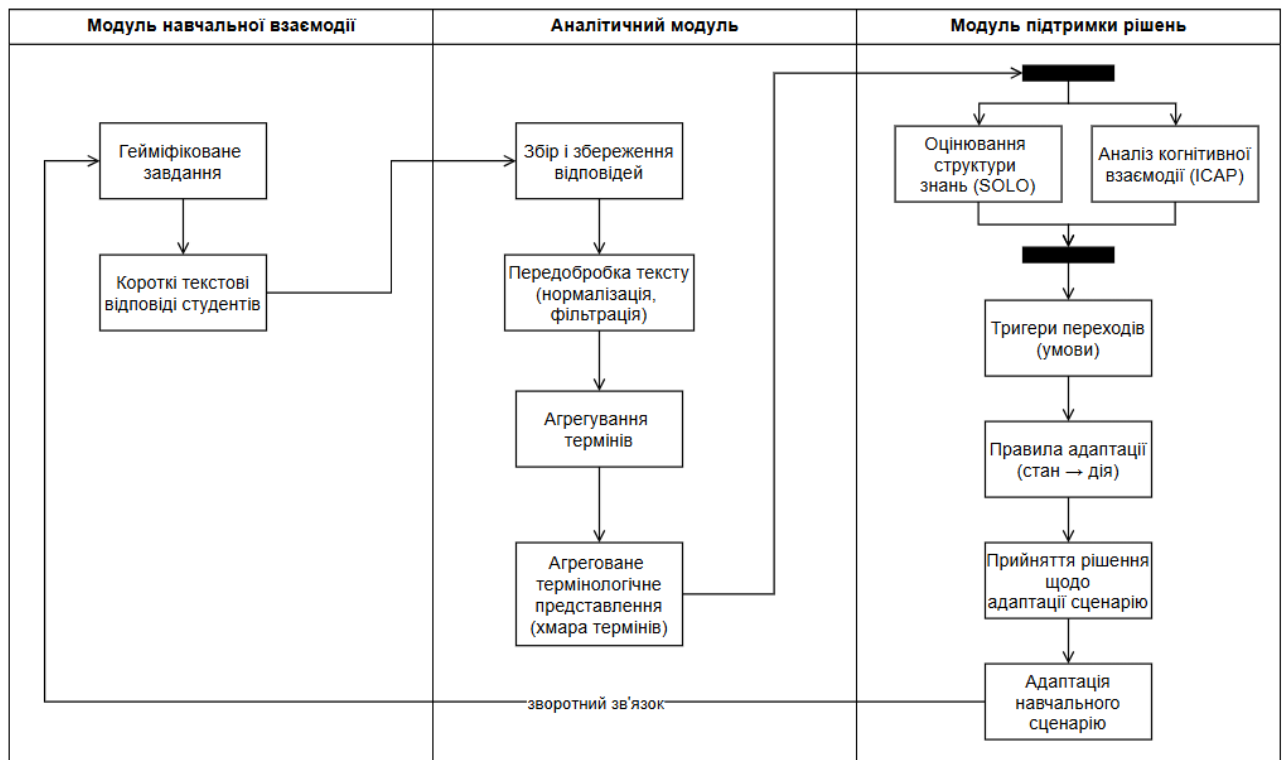


Рисунок 1 – Контур формуального аналізу знань і адаптації навчального сценарію в цифровому освітньому середовищі

Інтерпретація агрегованого сигналу в запропонованому контурі здійснюється із застосуванням таксономії SOLO для оцінювання структури знань [20] та рамки ICAP для аналізу когнітивного залучення студентів [21–23]. Використання цих рамок забезпечує узгоджене формування тригерів переходів між сценарними станами та підтримує прийняття рішень щодо адаптації навчального сценарію в реальному або квазі-реальному часі.

Правила адаптації задають відповідність між інтерпретованим станом групи та допустимими сценарними діями. Такі дії можуть включати зміну темпу пояснення, уточнення ключових концептів, ініціювання дискусії або постановку додаткових завдань. Остаточне рішення може залишатися за викладачем у режимі підтримки прийняття рішень або бути частково автоматизованим у межах цифрової платформи.

Виконана адаптація впливає на подальший перебіг заняття та ініціює наступну ітерацію контуру, що забезпечує безперервність формуального аналізу в межах сервісної моделі навчання. Оперативність побудови хмари термінів та мінімальна обчислювальна складність методу дозволяють реалізувати цей контур

із незначною затримкою, що є критично важливим для підтримки ефективного зворотного зв'язку під час навчальної взаємодії.

Таким чином, хмара термінів у запропонованому підході функціонує як інтерпретований аналітичний компонент сервісно-орієнтованої освітньої інформаційної системи, інтегрований у замкнений контур адаптації навчального сценарію та узгоджений із принципами формування й оцінювання й навчальної аналітики.

**Методологічний контур проєктування та оцінювання гейміфікованих компонентів освітнього середовища.** Для забезпечення відтворюваності та наукової коректності впровадження запропонованого методу доцільно розглядати його як інженерний артефакт у межах методології дослідження на основі наукового проєктування (Design Science Research, DSR) [18, 19]. DSR передбачає ітеративний процес формулювання проблеми, проєктування артефакта, теоретичного обґрунтування його структури та визначення критеріїв оцінювання ефективності. На відміну від описових педагогічних підходів, ця методологія дозволяє трактувати освітні механізми як штучно створені системні конструкції з визначеними входами, виходами та контуром зворотного зв'язку. У межах даного дослідження таким артефактом виступає метод формування аналізу знань на основі агрегованого термінологічного представлення, інтегрований у сервісно-орієнтовану архітектуру цифрового освітнього середовища.

Оцінювання артефакта в логіці DSR здійснюється за низкою критеріїв: інтерпретованість сформованого аналітичного сигналу; відтворюваність процедур передоброби та агрегування; обчислювальна ефективність і масштабованість; придатність до інтеграції в цифрову платформу; а також здатність підтримувати прийняття сценарних рішень на основі чітко визначених тригерів. Такий підхід забезпечує узгодження теоретичних засад гейміфікації та формування оцінювання з інженерною логікою проєктування інформаційних систем.

Інтерпретація результатів формування аналізу в запропонованому підході здійснюється із застосуванням таксономії SOLO та рамки ICAP. Таксономія SOLO дозволяє оцінювати якість структури знань за ступенем інтегрованості понять і

зв'язків між ними [20]. На відміну від таксономії Блума, яка класифікує переважно типи когнітивних дій і цілі навчання, SOLO орієнтована на характеристику сформованої структури знання. Це є принципово важливим у контексті аналізу хмар термінів, оскільки агреговане термінологічне представлення відображає не лише наявність окремих понять, а й їхню узгодженість та взаємозв'язок у межах колективного розуміння. Домінування розрізнених термінів може відповідати мультиструктурному рівню розуміння, тоді як узгоджене використання взаємопов'язаних понять — реляційному рівню, що свідчить про вищу інтегрованість знань.

Рамка ICAP забезпечує аналіз когнітивного залучення студентів у процесі формування хмар термінів, класифікуючи залучення за типом когнітивної діяльності, а не за формальними показниками активності [21–23]. На відміну від метрик поведінкової аналітики (наприклад, часу взаємодії з платформою або кількості виконаних дій) та таксономії Блума, ICAP дозволяє інтерпретувати фактичний характер взаємодії як активний, конструктивний або інтерактивний. Це є принципово важливим для оцінювання ефективності гейміфікованих механік, оскільки зовнішні прояви активності не завжди відображають глибину когнітивної обробки інформації. У межах запропонованого методу генерація власних термінів і участь у групових обговореннях інтерпретуються як прояви конструктивного та інтерактивного режимів залучення, які згідно з теоретичними положеннями ICAP асоціюються з вищими результатами навчання.

Поєднання SOLO та ICAP створює можливість комплексної оцінки як структурних аспектів навчання (результат сформованих знань), так і процесуальних характеристик когнітивної діяльності (характер залучення). У межах замкненого контуру формувального аналізу ці інтерпретаційні рамки використовуються для формування тригерів переходів між сценарними станами та забезпечують узгодженість між аналітичним сигналом і педагогічним рішенням.

Таким чином, методологічний контур дослідження поєднує інженерну рамку DSR із інтерпретаційними підходами SOLO та ICAP, що дозволяє розглядати

запропонований метод як формалізований, відтворюваний та теоретично обґрунтований компонент сервісно-орієнтованої цифрової освітньої системи.

**Обмеження методу та вимоги до інтеграції.** Попри концептуальну простоту й інтерпретованість запропонованого методу, його застосування в цифровому освітньому середовищі пов'язане з низкою методичних і технічних обмежень, які необхідно враховувати під час проектування та впровадження. Аналіз таких обмежень є важливим елементом дослідження в логіці DSR [18, 19], оскільки дозволяє визначити умови відтворюваності артефакта та межі його коректного застосування.

Першим обмеженням є втрата контексту при агрегуванні коротких текстових відповідей. Хмара термінів відображає частотний розподіл нормалізованих понять, однак не фіксує семантичні зв'язки між ними в межах окремих висловлювань. Це може призводити до спрощеної інтерпретації результатів або до втрати нюансів аргументації. Зниження цього ризику потребує чіткого формулювання завдань і поєднання агрегованого представлення з додатковим якісним аналізом окремих відповідей.

Другим обмеженням є чутливість до якості вхідних текстових даних. Орфографічні помилки, синонімія, варіативність граматичних форм і використання неточних термінів можуть призводити до штучної фрагментації понять та викривлення вагових коефіцієнтів. У зв'язку з цим обов'язковою умовою інтеграції методу є застосування процедур передобробки тексту (лематизація, нормалізація регістру, уніфікація синонімів, вилучення стоп-слів), а також контроль мінімальної частки участі студентів.

Третім обмеженням є ризик поверхневої інтерпретації результатів без урахування дидактичного контексту. Однакова конфігурація домінантних термінів може мати різні значення залежно від етапу заняття, типу завдання або попереднього досвіду групи. Отже, аналіз агрегованого сигналу повинен здійснюватися з урахуванням конкретної навчальної ситуації та не може бути повністю автоматизованою без експертного контролю.

Четвертим обмеженням є потенційна нерівномірність участі студентів у формуванні хмари термінів. Домінування активної підгрупи може спотворювати агрегований сигнал і створювати ілюзію колективного розуміння, що не відповідає реальному стану групи. Це потребує впровадження механізмів забезпечення мінімальної участі або використання додаткових індикаторів розподілу відповідей.

Узагальнення основних проблем інтерпретації та відповідних заходів зниження ризиків подано в таблиці 1.

Таблиця 1 Основні проблеми інтерпретації хмар термінів і заходи зниження ризику

<b>Проблема</b>	<b>Потенційний ризик</b>	<b>Заходи зниження</b>
Втрата контексту	Спрощена інтерпретація змісту відповідей	Поєднання агрегованого аналізу з вибірковим якісним аналізом
Лінгвістична варіативність	Фрагментація понять, викривлення частот	Лематизація, нормалізація, уніфікація синонімів
Нерівномірна участь	Домінування окремих студентів	Контроль мінімальної частки участі
Автоматизована інтерпретація без контексту	Хибні сценарні рішення	Збереження експертного контролю викладача

Крім зазначених обмежень, важливо враховувати технічні вимоги до інтеграції методу в цифрову платформу. До них належать: підтримка збору коротких текстових відповідей у синхронному режимі; наявність модулів передобробки тексту; забезпечення мінімальної затримки обробки даних; можливість відображення агрегованого сигналу в інтерфейсі викладача; а також

реалізація механізму фіксації сценарних переходів для подальшого аналізу ефективності адаптацій.

Таким чином, запропонований метод є інтерпретованим і обчислювально простим, однак його ефективність залежить від дотримання процедур передоброби, коректної інтерпретації результатів та узгодження з дидактичним контекстом. Визначення цих обмежень дозволяє окреслити межі застосування методу та створює основу для подальшої емпіричної валідації в реальних цифрових освітніх середовищах.

### **Висновки та напрями подальших досліджень**

У роботі запропоновано метод формульовального аналізу знань і адаптації навчального сценарію в сервісно-орієнтованих освітніх системах на основі агрегованого термінологічного представлення колективних текстових відповідей. Метод концептуалізовано як артефакт у межах методології DSR і інтегровано в замкнений контур, що включає такі етапи: збір відповідей, їх агрегування, інтерпретацію результатів, сценарний перехід та наступну ітерацію.

Аналіз публікацій показав, що наявні підходи до гейміфікації, формульовального оцінювання та навчальної аналітики або зосереджені на мотиваційних механіках, або ґрунтуються на параметризованих і латентних моделях вимірювання та прогнозування. Використання хмар термінів у навчанні переважно описується як дидактичний або інструментальний прийом без формалізації їх як компонента сервісно-орієнтованого контуру адаптації. Запропонований підхід заповнює зазначену прогалину шляхом формалізації процедур агрегування, інтерпретації та інтеграції в архітектуру цифрового освітнього середовища.

Методологічне поєднання DSR із рамками SOLO та ICAP забезпечує узгоджене трактування структурних характеристик результату (SOLO) і процесуальних характеристик когнітивної взаємодії (ICAP), що використовується для формування тригерів і правил сценарних переходів. Це підвищує

відтворюваність інтерпретації агрегованого сигналу та зменшує залежність контуру від неформалізованих оцінок.

Практична придатність методу визначається обчислювальною простотою, прозорістю та можливістю інтеграції без навчання або калібрування моделей. Визначені обмеження (втрата контексту, чутливість до якості текстових даних, нерівномірність участі, ризики автоматизованої інтерпретації без урахування дидактичного контексту) задають умови коректного застосування та вимоги до інтеграції в цифрову платформу.

Подальші дослідження доцільно спрямувати на емпіричну валідацію контуру в реальних цифрових освітніх середовищах, формалізацію правил сценарних переходів для часткової автоматизації, а також аналіз впливу різних гейміфікованих механік на стабільність термінологічного представлення та параметри когнітивного залучення.

#### **Список використаних джерел**

1. European Commission. Digital Education Action Plan 2021–2027: Resetting education and training for the digital age. Brussels, 2020.
2. UNESCO. Education 2030: Incheon Declaration and Framework for Action for the Implementation of Sustainable Development Goal 4. Paris : UNESCO, 2016.
3. Биков В. Ю., Вернигора С. М., Гуржій А. М., Новохатько Л. М., Спірін О. М., Шишкіна М. П. Проектування і використання відкритого хмароорієнтованого освітньо-наукового середовища закладу вищої освіти // Інформаційні технології і засоби навчання. 2019. Т. 74, № 6. С. 1–19. DOI: <https://doi.org/10.33407/itlt.v74i6.3499>.
4. Hamari J., Koivisto J., Sarsa H. Does gamification work? A literature review of empirical studies on gamification // Proceedings of the 47th Hawaii International Conference on System Sciences (HICSS). 2014. P. 3025–3034.
5. Лугова Т. А. Геймдизайн-орієнтований підхід до розробки навчальних дисциплін закладів вищої освіти // Інформаційні технології і засоби навчання. 2021. Т. 81, № 1. С. 235–255. DOI: <https://doi.org/10.33407/itlt.v81i1.3265>.

6. Горбань О. В. Освітлогічні аспекти відеоігрової діяльності у контексті сучасної освіти // Освітлогія. 2023. № 12. С. 7–19. DOI: <https://doi.org/10.28925/2226-3012.2023.12.1>.
7. Dubois L. E., Weststar J. Games-as-a-service: Conflicted identities on the new front-line of video game development // *New Media & Society*. 2022. Vol. 24, No. 5. P. 1157–1175. DOI: <https://doi.org/10.1177/1461444821995815>.
8. Black P., Wiliam D. Assessment and classroom learning // *Assessment in Education*. 1998. Vol. 5, No. 1. P. 7–74. DOI: <https://doi.org/10.1080/0969595980050102>.
9. Shute V. J. Focus on formative feedback // *Review of Educational Research*. 2008. Vol. 78, No. 1. P. 153–189. DOI: <https://doi.org/10.3102/0034654307313795>.
10. Brooks B. J., Gilbuena D. M., Krause S. J., Koretsky M. D. Using word clouds for fast, formative assessment of students' short written responses // *Chemical Engineering Education*. 2014. Vol. 48, No. 4. P. 190–198.
11. Kitchens M. B. Word clouds: An informal assessment of student learning // *College Teaching*. 2014. Vol. 62, No. 3. P. 113–114. DOI: <https://doi.org/10.1080/87567555.2013.807216>.
12. Atenstaedt R. Word cloud analysis of the BJGP: 5 years on // *British Journal of General Practice*. 2017. Vol. 67, No. 658. P. 231–232. DOI: <https://doi.org/10.3399/bjgp17X690833>.
13. Corbett A. T., Anderson J. R. Knowledge tracing: Modeling the acquisition of procedural knowledge // *User Modeling and User-Adapted Interaction*. 1994. Vol. 4. P. 253–278. DOI: <https://doi.org/10.1007/BF01099821>.
14. Piech C. та ін. Deep knowledge tracing // *Advances in Neural Information Processing Systems*. 2015. P. 505–513.
15. Embretson S. Item response theory for psychologists // *Quality of Life Research*. 2000. DOI: <https://doi.org/10.1023/B:QURE.0000021503.45367.F2>.
16. Rasch G. Probabilistic models for some intelligence and attainment tests. Copenhagen : Danish Institute for Educational Research, 1960.

17. Blei D. M., Ng A. Y., Jordan M. I. Latent Dirichlet allocation // Journal of Machine Learning Research. 2003. Vol. 3. P. 993–1022.
18. Hevner A. R., March S. T., Park J., Ram S. Design science in information systems research // MIS Quarterly. 2004. Vol. 28, No. 1. P. 75–105. DOI: <https://doi.org/10.2307/25148625>.
19. Peffers K., Tuunanen T., Rothenberger M. A., Chatterjee S. A design science research methodology for information systems research // Journal of Management Information Systems. 2007. Vol. 24, No. 3. P. 45–77. DOI: <https://doi.org/10.2753/MIS0742-1222240302>.
20. Biggs J. B., Collis K. F. Evaluating the quality of learning: The SOLO taxonomy. New York : Academic Press, 1982.
21. Chi M. T. H., Wylie R. The ICAP framework: Linking cognitive engagement to active learning outcomes // Educational Psychologist. 2014. Vol. 49, No. 4. P. 219–243. DOI: <https://doi.org/10.1080/00461520.2014.965823>.
22. Nault K., Nieto-Taborda M., Ruhi U. Linking ICAP and Bloom's taxonomy: A comprehensive framework to assess students' cognitive engagement in the context of GenAI // Proceedings of the 5th International Conference on Artificial Intelligence and Education (ICAIE). Suzhou, China, 2025. P. 460–464. DOI: <https://doi.org/10.1109/ICAIE64856.2025.11158527>.
23. Chi M. T. H., Kang S., Yaghmourian D. L. Why students learn more from dialogue- than monologue-videos: Analyses of peer interactions // Journal of the Learning Sciences. 2017. Vol. 26, No. 1. P. 10–50. DOI: <https://doi.org/10.1080/10508406.2016.1204546>.

## **References**

- European Commission. (2020). *Digital Education Action Plan 2021–2027: Resetting education and training for the digital age*.
- UNESCO. (2016). *Education 2030: Incheon Declaration and Framework for Action for the Implementation of Sustainable Development Goal 4*. UNESCO.

Bykov, V. Y., Vernyhora, S. M., Hurzhii, A. M., Novokhatko, L. M., Spirin, O. M., & Shyshkina, M. P. (2019). Design and use of an open cloud-based educational and research environment of a higher education institution. *Information Technologies and Learning Tools*, 74(6), 1–19. <https://doi.org/10.33407/itlt.v74i6.3499>

Hamari, J., Koivisto, J., & Sarsa, H. (2014). Does gamification work? A literature review of empirical studies on gamification. In *Proceedings of the 47th Hawaii International Conference on System Sciences (HICSS)* (pp. 3025–3034).

Luhova, T. A. (2021). Game design-oriented approach to the development of higher education disciplines. *Information Technologies and Learning Tools*, 81(1), 235–255. <https://doi.org/10.33407/itlt.v81i1.3265>

Horban, O. V. (2023). Educational aspects of video game activity in the context of modern education. *Osvitoloziia*, 12, 7–19. <https://doi.org/10.28925/2226-3012.2023.12.1>

Dubois, L. E., & Weststar, J. (2022). Games-as-a-service: Conflicted identities on the new front-line of video game development. *New Media & Society*, 24(5), 1157–1175. <https://doi.org/10.1177/1461444821995815>

Black, P., & Wiliam, D. (1998). Assessment and classroom learning. *Assessment in Education*, 5(1), 7–74. <https://doi.org/10.1080/0969595980050102>

Shute, V. J. (2008). Focus on formative feedback. *Review of Educational Research*, 78(1), 153–189. <https://doi.org/10.3102/0034654307313795>

Brooks, B. J., Gilbuena, D. M., Krause, S. J., & Koretsky, M. D. (2014). Using word clouds for fast, formative assessment of students' short written responses. *Chemical Engineering Education*, 48(4), 190–198.

Kitchens, M. B. (2014). Word clouds: An informal assessment of student learning. *College Teaching*, 62(3), 113–114. <https://doi.org/10.1080/87567555.2013.807216>

Atenstaedt, R. (2017). Word cloud analysis of the BJGP: 5 years on. *British Journal of General Practice*, 67(658), 231–232. <https://doi.org/10.3399/bjgp17X690833>

Corbett, A. T., & Anderson, J. R. (1994). Knowledge tracing: Modeling the acquisition of procedural knowledge. *User Modeling and User-Adapted Interaction*, 4, 253–278. <https://doi.org/10.1007/BF01099821>

Piech, C., et al. (2015). Deep knowledge tracing. In *Advances in Neural Information Processing Systems* (pp. 505–513).

Embretson, S. (2000). Item response theory for psychologists. *Quality of Life Research*. <https://doi.org/10.1023/B:QURE.0000021503.45367.F2>

Rasch, G. (1960). *Probabilistic models for some intelligence and attainment tests*. Danish Institute for Educational Research.

Blei, D. M., Ng, A. Y., & Jordan, M. I. (2003). Latent Dirichlet allocation. *Journal of Machine Learning Research*, 3, 993–1022.

Hevner, A. R., March, S. T., Park, J., & Ram, S. (2004). Design science in information systems research. *MIS Quarterly*, 28(1), 75–105. <https://doi.org/10.2307/25148625>

Peppers, K., Tuunanen, T., Rothenberger, M. A., & Chatterjee, S. (2007). A design science research methodology for information systems research. *Journal of Management Information Systems*, 24(3), 45–77. <https://doi.org/10.2753/MIS0742-1222240302>

Biggs, J. B., & Collis, K. F. (1982). *Evaluating the quality of learning: The SOLO taxonomy*. Academic Press.

Chi, M. T. H., & Wylie, R. (2014). The ICAP framework: Linking cognitive engagement to active learning outcomes. *Educational Psychologist*, 49(4), 219–243. <https://doi.org/10.1080/00461520.2014.965823>

Nault, K., Nieto-Taborda, M., & Ruhi, U. (2025). Linking ICAP and Bloom's taxonomy: A comprehensive framework to assess students' cognitive engagement in the context of GenAI. In *Proceedings of the 5th International Conference on Artificial Intelligence and Education (ICAIE)* (pp. 460–464). <https://doi.org/10.1109/ICAIE64856.2025.11158527>

Chi, M. T. H., Kang, S., & Yaghmourian, D. L. (2017). Why students learn more from dialogue- than monologue-videos: Analyses of peer interactions. *Journal of the Learning Sciences*, 26(1), 10–50. <https://doi.org/10.1080/10508406.2016.1204546>

# **ДОДАТОК Б**

## **ПЕРЕХІД ВІД МОДЕЛІ GAME-AS-A-PRODUCT ДО GAME-AS-A-SERVICE**

III Всеукраїнська науково-практична конференція «Системні науки та інформатика»

Аркушів 7

Київ 2024



III Всеукраїнська науково-практична конференція

# **«Системні науки та інформатика»**

**Збірник доповідей**

25–29 листопада 2024 року

Київ, Україна

<b>Дослідження комп'ютерної симуляції фізичних процесів для нестандартних моделей простору</b>	<b>374</b>
<i>Титарчук В.І., Безносик О.Ю.</i>	
<b>Оцінка сумісності кандидата та компанії на основі семантичного аналізу історії наймів та резюме</b>	<b>380</b>
<i>Товстенко Н.А., Казаков В.В., Письменний І.О.</i>	
<b>Використання LLM агентів в першій ланці медичної допомоги</b>	<b>386</b>
<i>Тринус Н.В., Кислий Р.В.</i>	
<b>Автоматизація розпізнавання точок відмови розподіленої системи на основі аналізу трасування запитів, логів та метрик</b>	<b>392</b>
<i>Хитрий М.С., Письменний І.О.</i>	
<b>Безпека блокчейн систем в мікросервісній архітектурі</b>	<b>395</b>
<i>Хуторянський Д.О., Булах Б.В.</i>	
<b>Виконання розподілених обчислень з віртуалізацією середовища користувача</b>	<b>399</b>
<i>Шаблій В.С., Булах Б.В.</i>	
<b>Дослідження відкритих платформ слухопротезування для розробки набору числових алгоритмів, оптимізованих для використання в енергоефективних системах цифрової обробки сигналів</b>	<b>405</b>
<i>Шляхов Д.С., Кірюша Б.А.</i>	
<b>Перехід від моделі Game-as-a-Product до Game-as-a-Service</b>	<b>410</b>
<i>Шпаджківський О.В.</i>	

#### **Секція 04**

##### Системи і методи штучного інтелекту

<b>Пошук відповідності між точками та їх відстеження</b>	<b>415</b>
<i>Бірук С.В., Шаповал Н.В.</i>	
<b>Напівкероване моделювання впливу для електронної комерції</b>	<b>419</b>
<i>Заїка Б.Ю., Шаповал Н.В.</i>	
<b>Виявлення шахрайської платіжної діяльності на основі методів штучного інтелекту</b>	<b>427</b>
<i>Пишнюк А.О., Кузнєцова Н.В.</i>	
<b>Використання рекомендаційної системи у транспортній галузі на основі методів штучного інтелекту</b>	<b>434</b>
<i>Поліщук М.С., Гуськова В.Г.</i>	
<b>Моніторинг росту рослин на основі зображень методами глибокого навчання</b>	<b>440</b>
<i>Сандрацький В.В., Новогрудська Р.Л., Шаповал Н.В.</i>	
<b>Напівкероване навчання зорового трансформера для задачі сегментації дорожнього трафіку в неструктурованому середовищі</b>	<b>444</b>
<i>Шабо О., Шаповал Н.</i>	
<b>Кластеризація текстових даних на основі вкладення слів</b>	<b>450</b>
<i>Шутяк Д.О., Подколзін Г.Б.</i>	
<b>Рекомендація хештегів методами глибокого навчання на основі мультимодальних даних</b>	<b>457</b>
<i>Яковлев С.О., Шаповал Н.В.</i>	

# ПЕРЕХІД ВІД МОДЕЛІ GAME-AS-A-PRODUCT ДО GAME-AS-A-SERVICE

Шпадківський О.В.

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

[vaivipir@gmail.com](mailto:vaivipir@gmail.com)

Дослідження аналізує перехід від ігрової моделі *Game-as-a-Product* (GaaP) до ігрової моделі *Game-as-a-Service* (GaaS). Метою дослідження є аналіз ключових аспектів даного переходу, використовуючи методи порівняння, аналізу та узагальнення. На основі отриманих даних, було визначено, що модель GaaS стимулюється розвитком хмарних сервісів, інтернет-технологій, мобільних платформ та штучного інтелекту. Останнє, у свою чергу, дозволяє якісно персоналізувати ігровий досвід. Важливою складовою є безперервне залучення гравців, яке забезпечується за допомогою зворотнього зв'язку, мікротранзакцій, підписок та внутрішньоігрових покупок.

Ключові слова: *Game-as-a-Product* (GaaP), *Game-as-a-Service* (GaaS), персоналізація ігрового досвіду, ігрова індустрія.

## 1. ВСТУП

Протягом останніх десятиліть ігрова індустрія зазнала суттєвих трансформацій, обумовлених як технологічними інноваціями, так і еволюцією бізнес-моделей. Однією з найбільш значущих змін став перехід від концепції *Game-as-a-Product* (далі – GaaP, "гра як продукт") до *Game-as-a-Service* (далі – GaaS, "гра як сервіс").

Перехід від GaaP до GaaS відображає новітній підхід до розробки ігор, заснований не лише на створенні завершеного продукту, але й на формуванні гнучкої та динамічної екосистеми. Ця екосистема постійно оновлюється і адаптується до потреб користувачів, що дозволяє залучати нову аудиторію та підтримувати інтерес існуючих гравців протягом тривалого часу.

**Актуальність дослідження** обумовлюється наявною потребою у розгляді та детальному аналізі особливостей переходу від ігрової моделі *Game-as-a-Product* (далі – GaaP, "гра як продукт") до ігрової моделі *Game-as-a-Service* (далі – GaaS, "гра як сервіс"), котрий є однією з найбільш значущих для ігрової індустрії за останнє десятиліття. **Метою дослідження** є розкрити та проаналізувати основні аспекти переходу від ігрової моделі GaaP до ігрової моделі GaaS.

## **2. ПЕРЕХІД ВІД ГРИ ЯК ПРОДУКТУ ДО ГРИ ЯК СЕРВІСУ: ТЕОРЕТИЧНИЙ АНАЛІЗ**

Еволюція ігрової індустрії від ігрової моделі GaaS до GaaS відображає глибокі трансформації, зумовлені технологічними інноваціями, змінами в бізнес-моделях та зростаючими вимогами споживачів. Спочатку ігри поширювалися через фізичні носії, такі як картриджі та диски, і були завершеними продуктами без можливості оновлення після релізу. Розвиток персональних комп'ютерів та Інтернету в 1990-х роках започаткував перехід до моделей з регулярними оновленнями та підписками, як у випадку з Ultima Online та EverQuest [1]. Запуск цифрових платформ, зокрема Steam у 2003 році, радикально змінив спосіб дистрибуції ігор, дозволивши розробникам безпосередньо поширювати оновлення та додатковий контент. Мобільна революція початку 2010-х років та поява платформ App Store і Google Play впровадили моделі монетизації на кшталт free-to-play з мікротранзакціями, що сприяло популярності ігор на зразок Angry Birds та Clash of Clans. Високошвидкісний Інтернет та технології серверної інфраструктури дозволили створювати масштабні багатокористувацькі проекти, такі як World of Warcraft, що успішно використовували модель підписки та регулярних оновлень контенту [2]. Хмарні технології та стрімінгові сервіси, такі як Playstation Now та Xbox Cloud розширили можливості GaaS, надаючи доступ до високоякісних ігор без потреби в потужному апаратному забезпеченні. Еволюція бізнес-моделей, включаючи free-to-play, підписки, мікротранзакції та сезонні пропуски, відіграла ключову роль у переході до GaaS, забезпечуючи стабільніший дохід для розробників [3]. Соціальні медіа та стрімінгові платформи створили нові канали для взаємодії з аудиторією, підсилюючи важливість активних спільнот у успіху сучасних ігор. Інтеграція технологій штучного інтелекту дозволила створювати адаптивні та персоналізовані ігрові середовища, підвищуючи рівень залученості гравців [5]. Перехід до GaaS обумовлений розвитком цифрової дистрибуції та високошвидкісного Інтернету, зміною споживчих уподобань, еволюцією бізнес-моделей, технологічними інноваціями, соціальними факторами та глобалізацією ринку [4]. Порівняння моделей GaaS та GaaS показує фундаментальні зміни у підходах до розробки, монетизації та взаємодії з гравцями: GaaS пропонує регулярні оновлення, різноманітні моделі монетизації, довготривалу взаємодію та акцент на соціальних аспектах, тоді як GaaS базується на одноразових продажах завершених продуктів. Хоча GaaS залишається актуальною для певних жанрів та аудиторій, GaaS стає домінуючою моделлю, відповідаючи сучасним вимогам гравців до динамічного та інтерактивного контенту. Цей перехід не лише змінив спосіб створення та розповсюдження ігор, але й вплинув на саму природу ігрового досвіду. Зокрема, поглибив зв'язок між гравцями, стаючи більш соціально орієнтованим. Окрім того, досвід гравця став більш інтерактивним та персоналізованим.

## **3. ТЕХНОЛОГІЧНІ ТЕНДЕНЦІЇ У СФЕРІ GAAS**

Модель GaaS спричинила значні інновації у розробці та видавництві ігор, впровадивши нові підходи, що підвищують якість продукту та покращують досвід

гравців. Однією з ключових інновацій є динамічний контент та адаптивність ігрового середовища, де регулярні оновлення та розширення надають гравцям нові можливості, підтримуючи їхній інтерес до гри. Адаптивність досягається за допомогою штучного інтелекту та алгоритмів машинного навчання, які аналізують поведінку гравців і відповідно налаштовують складність або пропонують індивідуальні завдання, що підвищує персоналізацію ігрового процесу. Впровадження методологій Agile (базується на ітеративній розробці) та DevOps(опирається на збирання, налаштування та розгортання програмного забезпечення) у розробці ігор дозволяє здійснювати поетапну розробку з можливістю регулярного вдосконалення, швидко реагуючи на зміни та побажання гравців. DevOps сприяє безперервному вдосконаленню та автоматизації процесів розробки та розгортання, що знижує час між ітераціями та спрощує внесення змін без перерв у роботі гри. Прямий зворотній зв'язок з гравцями через соціальні мережі, форуми та внутрішньоігрові опитування забезпечує швидке адаптування гри до потреб аудиторії та підвищує рівень залученості користувачів, створюючи довірчі відносини та спільне відчуття причетності до процесу розробки. Інтеграція аналітики та великих даних дозволяє розробникам відстежувати поведінкові патерни гравців, ухвалювати обґрунтовані рішення щодо модифікації контенту та покращувати користувацький досвід, а також краще сегментувати аудиторію для пропозиції індивідуалізованих можливостей. Технологічна підтримка для регулярних оновлень, з використанням хмарних технологій та сервісів безперервної інтеграції та доставки, забезпечує оперативне оновлення ігрового контенту без значних інфраструктурних витрат та підтримує стабільність гри.

Також у контексті GaaS запроваджуються нові бізнес-моделі та методи монетизації, які забезпечують стабільний потік доходів та подовжують життєвий цикл гри. Мікротранзакції дозволяють гравцям здійснювати невеликі покупки всередині гри, надаючи можливості для персоналізації та покращення ігрового досвіду. Підписні сервіси забезпечують стабільний потік доходів через регулярну плату за доступ до ексклюзивного контенту, спеціальних функцій або бонусів, мотивуючи гравців регулярно повертатися до гри. Внутрішньоігрові покупки та платний контент розширюють функціональні можливості та забезпечують постійне надходження доходів навіть після початкового завантаження гри. Модель free-to-play з платним контентом має широкий потік нових гравців, забезпечуючи високий рівень залучення та підтримуючи баланс між безкоштовним та платним контентом.

Вплив таких технологічних трендів як штучний інтелект, доповнена та віртуальна реальність, хмарні технології та Інтернет речей, значно розширив можливості GaaS. Штучний інтелект приєє у персоналізації ігрового досвіду та підвищує інтерактивність. Індивідуалізація та “гнучкість” контенту досягається через ретельний аналіз широкого набору маркерів, які відзначають внутрішньоігрову поведінку гравців. Доповнена та віртуальна реальність створюють нові формати взаємодії, надаючи користувачам відчуття присутності у віртуальних світах та збагачуючи реальне середовище інтерактивними елементами. Хмарні технології забезпечують зберігання великого обсягу даних, знижують витрати на інфраструктуру та дозволяють запускати ігри на віддалених серверах, розширюючи можливості GaaS. Інтеграція з Інтернетом речей відкриває нові перспективи для

інтеграції ігрового процесу з фізичними пристроями, підвищуючи взаємодію між віртуальним та фізичним світом.

#### **4. ВПЛИВ GaaS НА ІГРОВУ ІНДУСТРІЮ ТА КОРИСТУВАЧІВ**

Впровадження моделі «гра як сервіс» (GaaS) суттєво змінило поведінку гравців, їхні очікування та взаємодію з ігровим контентом, переходячи від одноразового проходження до тривалої та регулярної участі. Гравці тепер очікують постійних оновлень, нових функцій та унікальних подій, що підтримують їхній інтерес. Підвищена роль соціальної взаємодії стала невід'ємною складовою GaaS, де гравці прагнуть співпраці, конкуренції та взаємодії зі спільнотою, що підсилює їхню залученість та значущість ігрових досягнень. Бажання персоналізації та індивідуалізації контенту стимулює гравців до кастомізації персонажів і придбання ексклюзивних предметів, що підвищує рівень залученості та створює додаткові джерела доходу для розробників. Зростання вимог до якості обслуговування та підтримки формує новий стандарт обслуговування, який вимагає від розробників оперативної реакції на запити аудиторії та забезпечення надійності сервісу.

Етичні та соціальні аспекти GaaS викликають дискусії щодо впливу на психічне та фізичне здоров'я користувачів, особливо в контексті можливої залежності від ігор [6]. Використання механік, що стимулюють тривалу взаємодію, підкреслює необхідність відповідального підходу до дизайну ігор, включаючи надання гравцям інструментів для контролю часу, проведеного у грі. Монетизація через мікротранзакції та внутрішньоігрові покупки породжує питання щодо етичності, особливо якщо вона створює нерівність між платними та безкоштовними користувачами [7]. На наш погляд, важливим є дотримання балансу, котрий гарантував би чесність ігрового процесу та зберігав довіру гравців. Використання даних гравців для персоналізації контенту піднімає питання конфіденційності, вимагаючи від розробників дотримання етичних стандартів при зборі та зберіганні інформації. Майбутні перспективи GaaS пов'язані з розширенням можливостей завдяки новим технологіям, таким як штучний інтелект, хмарні сервіси, доповнена реальність та Інтернет речей, які сприятимуть створенню більш інтерактивних та персоналізованих ігор. Однак розробники зіткнуться з викликами перенасичення ринку та посилення конкуренції, що вимагатиме інноваційного підходу та унікального контенту для привернення уваги аудиторії. Забезпечення сталого доходу та лояльності гравців стане критичним, оскільки підтримка інтересу аудиторії потребуватиме регулярних оновлень, високого рівня обслуговування та уважного відстеження потреб користувачів. Етичні виклики, пов'язані з монетизацією та можливістю залежності від ігор, вимагають від розробників відповідального підходу до дизайну ігрових механік та впровадження опцій для контролю часу у грі. Таким чином, модель GaaS продовжує еволюціонувати, впливаючи на поведінку гравців та встановлюючи нові стандарти в індустрії ігор. Успіх майбутніх проєктів залежатиме від здатності розробників збалансувати інновації, етичні аспекти та задоволення потреб гравців, підтримуючи довіру та лояльність аудиторії в умовах зростаючої конкуренції.

## 5. ВИСНОВКИ

Перехід від моделі Game-as-a-Product (GaaP) до Game-as-a-Service (GaaS) став однією з найбільших змін у сучасній ігровій індустрії, що кардинально змінила підходи до розробки, монетизації та взаємодії з користувачами. Впровадження GaaS дозволило розробникам створити динамічні та адаптивні ігри, які постійно оновлюються і підтримують інтерес гравців протягом тривалого періоду. Такий підхід також забезпечує стабільний потік доходів через різноманітні форми монетизації, такі як мікротранзакції, підписки та внутрішньоігрові покупки.

У результаті дослідження вдалось визначити, що, що успішність моделі GaaS значною мірою залежить від впровадження новітніх технологій, таких як штучний інтелект, доповнена та віртуальна реальність, хмарні сервіси та Internet of Things. Ці інновації сприяють створенню персоналізованого, інтерактивного досвіду, що відповідає сучасним очікуванням гравців. Технологічний прогрес також відкриває нові можливості для розробників, зокрема, у створенні інтеграцій з повсякденними девайсами та забезпеченні доступу до високоякісного контенту навіть на менш потужних пристроях.

Однак разом із перевагами модель GaaS створює й певні виклики. Серед основних проблем можна виділити необхідність підтримки високих стандартів обслуговування, вирішення питань конфіденційності та забезпечення балансу між платним і безкоштовним контентом. Гравці стали більш обізнаними і критично налаштованими до агресивної монетизації, що підвищує важливість відповідального підходу до дизайну ігор, включаючи можливості для контролю часу в грі та мінімізацію ризиків залежності.

Таким чином, майбутній успіх проєктів GaaS залежатиме від здатності розробників підтримувати баланс між економічними інтересами та потребами аудиторії. Важливим аспектом стане врахування етичних питань, пов'язаних із тривалою взаємодією гравців із контентом та монетизаційними стратегіями. Розробники, які зможуть зберігати довіру аудиторії та постійно вдосконалювати ігровий процес, мають високі шанси на досягнення лояльності користувачів і здобуття довготривалого успіху.

Тож модель GaaS не лише змінює спосіб створення ігор, але й формує нові стандарти взаємодії між розробниками та гравцями, які визначатимуть майбутнє ігрової індустрії.

## ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Kobayashi, Nobushige, Koyama, Yuhsuke, 2020, The Early History of the Hobbyist Production Field of Video Games and its Impacts on the Establishment of Japan's Video Game Industries, 73–82 p.
2. Kerr A. Global games: production circulation and policy in the networked era. Routledge, 2016, p. 65.
3. Davidovici-Nora M. Innovation in business models in the video game industry: free-to-play or the gaming experience as a service. Computer Games Journal, 2013, vol. 2, no. 3, pp. 22–51.
4. Wilhelmsson U., Wang W., Zhang R., et al. Shift from game-as-a-product to game-as-a-service research trends. Service Oriented Computing and Applications, 2022, vol. 16, pp. 79–81.
5. Morton D. What Netflix's entrance in gaming means for big and small gamers alike. Forbes, 2021. Available at: <https://www.forbes.com/sites/forbestechcouncil/2021/11/08/what-netflixs-entrance-in-gaming-means-for-big-and-small-gamers-alike/> (дата звернення: 14.11.2024)
6. Mohammad, S., Jan, R. A., & Alsaedi, S. L. (2023). Symptoms, Mechanisms, and Treatments of Video Game Addiction. Cureus, 15(3).
7. Lelonek-Kulet, B., Bartczuk, R. P., & Wiechetek, M. (2020). Pay for play – behavioural patterns of pay-to-win gaming. Computers in Human Behavior, 115(1).