

УДК 004.8:37.091.3:51

doi: 10.32620/aktt.2026.2.06

О. О. ЄВДОКИМОВ, О. В. ЛУЧШЕВА

Національний аерокосмічний університет

«Харківський авіаційний інститут», Харків, Україна

РОЗРОБКА ПРОТОТИПУ ІНТЕЛЕКТУАЛЬНОЇ ВЕБСИСТЕМИ НАВЧАННЯ МАТЕМАТИКИ НА ОСНОВІ АДАПТИВНОГО СЦЕНАРІЮ

Предметом дослідження є архітектура та функціональна реалізація прототипу веборієнтованої інтелектуальної навчальної системи для адаптивного навчання математики в онлайн-середовищі. **Метою** роботи є розробка та обґрунтування концепції прототипу інтелектуальної вебсистеми, що забезпечує параметризовану генерацію математичних задач, автоматизовану перевірку відповідей, багаторівневий механізм підказок та індивідуальну навчальну траєкторію на основі моделі ймовірного відстеження засвоєння знань. Для досягнення мети вирішувалися такі **завдання**: аналіз сучасних підходів до побудови інтелектуальних навчальних систем та моделей відстеження знань; формування моделі предметної області з використанням графа знань і зв'язків передування; розробка модулів шаблонної генерації задач та автоматизованої перевірки відповідей; формалізація адаптивного навчального сценарію на основі байєсівського оновлення ймовірності засвоєння; проектування ієрархічного механізму контекстних підказок; створення рішень користувачького інтерфейсу для введення математичних виразів та оперативного зворотного зв'язку; оцінка можливостей застосування генеративного штучного інтелекту для автоматизації створення навчального контенту. **Методи** дослідження включають системний аналіз, функціональну декомпозицію, порівняльний огляд літератури з інтелектуальних навчальних систем та комп'ютерного оцінювання в математиці, реалізацію класичної моделі ймовірного відстеження знань, а також прототипування клієнт-серверної вебсистеми (клієнтська частина на базі – Angular). **Висновки**. Розроблений прототип реалізує замкнений цикл адаптивного навчання, що включає динамічну генерацію задач, введення та перевірку відповіді, надання контекстної підказки, оновлення оцінки засвоєння знань і вибір наступного навчального кроку. Система вже функціонує як публічний веб-прототип і створює основу для емпіричної перевірки ефективності адаптивного навчання математики. **Наукова новизна** полягає в комплексній інтеграції в межах однієї вебплатформи: шаблонної генерації задач, автоматизованої перевірки математичних виразів, ієрархічного механізму підказок, моделі ймовірного відстеження знань для керування траєкторією, графа знань із зв'язками передування та обґрунтуванні використання генеративного штучного інтелекту для автоматизації контенту й діагностики помилок.

Ключові слова: інтелектуальна навчальна система; адаптивне навчання; навчання математики; вебсистема; відстеження знань Байєса; генерація задач; підказки; граф знань.

1. Вступ

Стійке зниження рівня математичної підготовки вступників і студентів авіаційно-космічних спеціальностей поєднується з іншим системним викликом – обмеженими можливостями індивідуального супроводу в умовах масового навчання. Парадокс сучасної математичної освіти полягає в тому, що саме предмет, який вимагає послідовного формування понять, постійного тренування та адресного виправлення помилок, найчастіше викладається у форматі, де персоналізований супровід є фрагментарним. Класична постановка цієї проблеми пов'язана з відомим «2-sigma problem» Б. Блума: результати індивідуального навчання в середньому перевищують результати

традиційного групового навчання приблизно на дві стандартні похибки [1].

Інтелектуальні навчальні системи (Intelligent Tutoring Systems, ITS) розглядаються як один із найбільш послідовних шляхів масштабування індивідуалізованого педагогічного впливу. У сучасному розумінні ITS – це системи, що поєднують модель предметної області, модель користувача, механізм педагогічного рішення та засоби інтерфейсу користувача [2]. Оглядіві праці демонструють, що системи покровокої підтримки та змістовного зворотного зв'язку можуть забезпечувати відчутний навчальний ефект порівняно з неадаптивними комп'ютерними засобами [3, 4].



Для авіаційно-космічної галузі якісна математична підготовка набуває особливої актуальності. Точні науки становлять фундамент для опанування складних технічних дисциплін – від аеродинаміки та динаміки польоту до систем керування космічними апаратами та інформаційних технологій. Саме тому помилки користувача зазвичай не є випадковими: вони часто відображають незасвоєні перетворення, хибні операції, порушення порядку дій або неформалізовані стратегії розв'язування.

Водночас сучасні ITS стикаються з низкою обмежень, пов'язаних із масштабованістю, складністю побудови формальних моделей предметної області, а також обмеженою адаптивністю в умовах відкритого онлайн-середовища. З розвитком вебтехнологій і штучного інтелекту виникає потреба у створенні нових класів інтелектуальних навчальних систем, здатних поєднувати адаптивність, автоматичну генерацію навчального контенту та інтеграцію з цифровими освітніми платформами.

У цьому контексті актуальним є розроблення веборієнтованої інтелектуальної навчальної системи, які забезпечують динамічну генерацію задач, автоматичне оцінювання відповідей і формування індивідуальних освітніх траєкторій на основі моделей відстеження знань.

1.1. Мотивація дослідження

Для точних дисциплін веборієнтований формат має кілька принципових переваг: доступність без спеціалізованого програмного забезпечення, зручність коротких тренувальних сесій, можливість швидко повторювати варіативні варіанти одного типу задач і накопичувати дані про взаємодію користувача із системою. Однак сама наявність вебінтерфейсу не робить систему інтелектуальною. Інтелектуальність у цьому контексті означає здатність інтерпретувати хід роботи користувача, диференціювати рівні складності, добирати підказки, фіксувати типові помилки і змінювати траєкторію взаємодії на підставі оцінки стану засвоєння [2, 7].

Прототип розроблявся як проміжна, але функціонально цілісна ланка між статичним електронним курсом і повноцінною інтелектуальною навчальною системою. Його завдання полягає не лише в поданні задач користувачу, а й у створенні керованого сценарію роботи: від генерації чергового варіанта до перевірки відповіді, від локальної підказки – до рішення про перехід на іншу задачу. Такий сценарний підхід добре узгоджується з логікою адаптивного навчання, де навчальний прогрес моделюється через послідовність окремих взаємодій, а не лише через підсумковий тест.

Розроблений прототип реалізує ці принципи у вигляді веборієнтованої клієнт-серверної системи,

що поєднує можливості електронного курсу, автоматизованого тренажера та інтелектуальної навчальної системи. Архітектура системи передбачає розділення рівнів подання, бізнес-логіки та збереження даних, що забезпечує масштабованість, накопичення навчальної аналітики та підтримку індивідуальних освітніх траєкторій.

На відміну від традиційних вебрішень, система використовує ймовірнісні моделі відстеження знань, що дозволяє не лише фіксувати результати виконання задач, а й оцінювати стан засвоєння окремих компонентів знань та динамічно обирати наступні кроки навчання. Такий підхід забезпечує перехід від статичного представлення навчального матеріалу до динамічного керування процесом навчання.

1.2. Аналіз існуючих досліджень

Сучасне поле досліджень ITS сформовано на перетині когнітивної психології, штучного інтелекту, аналітики навчання (learning analytics) та комп'ютерного навчання (computer-based learning). Праця В. Woolf систематизує архітектурні компоненти інтелектуального викладача і показує, що адаптивність системи залежить не від окремої евристики, а від зв'язку між моделлю студента, моделлю задач та механізмом педагогічного вибору [2]. Огляд К. VanLehn засвідчує, що системи покрокового навчання (step-based tutoring systems) зазвичай перевищують за ефективністю системи, орієнтовані на відповідь (answer-based systems), оскільки надають підтримку не лише в кінцевій точці, а й під час проміжних кроків [3]. Мета-аналіз W. Ma та співавторів додатково підтверджує позитивний вплив ITS на навчальні результати у різних предметних галузях [4].

Для моделювання стану знань у системах адаптивного навчання важливе місце посідають підходи відстеження знань (knowledge tracing). Базова модель А. Corbett та J. Anderson пропонує інтерпретований і достатньо компактний спосіб оцінки засвоєння окремого компонента знань через параметри ймовірності навчання (learn), випадкової помилки (slip) та вгадування (guess) [5]. Надалі цей напрям було розвинено в непромерених моделях глибокого відстеження знань (Deep Knowledge Tracing), які краще відтворюють послідовні залежності в довгих навчальних траєкторіях, хоча часто поступаються класичним моделям за рівнем інтерпретованості [6]. Узагальнюючий огляд S. Shen та співавторів показує, що вибір конкретної моделі відстеження знань має залежати від формату задач, доступного обсягу даних та вимог до пояснюваності педагогічного рішення [8].

Окремий напрям становлять системи автоматизованого оцінювання математичних відповідей. У математиці недостатньо зводити оцінювання лише

до тестів із вибором однієї відповіді; набагато важливішими є параметризація задач, аналіз типових помилок стратегій, формувальний характер зворотного зв'язку та можливість багаторазового тренування. Праці С. Sangwin і М. Greenhow доводять, що ефективне комп'ютерне оцінювання (computer-aided assessment) у математичних дисциплінах спирається на ретельно продуману структуру завдань, контроль випадкових параметрів і кодування типових помилок або неправильних правил (malformed rules) [9, 10].

У вітчизняних дослідженнях також розглядаються підходи до побудови інтелектуальних навчальних систем для математики, які поєднують формалізацію задач, їх класифікацію за складністю та використання алгоритмічних моделей розв'язування. У роботі [13] запропоновано розділення задач на прості та складні, а також наведено формальні моделі задач, що створює основу для побудови адаптивних навчальних систем.

Зокрема, у [14] розглянуто концепцію інтелектуальної системи підготовки до STEM-іспитів, яка орієнтована на адаптивне тренування та автоматизований аналіз навчальної діяльності користувача. Це підтверджує актуальність використання ITS у задачах підготовки до стандартизованих математичних іспитів. Попри значний розвиток інтелектуальних навчальних систем, методів відстеження знань та засобів автоматизованого оцінювання, залишається недостатньо дослідженим питання створення веборієнтованих інтелектуальних систем навчання математиці, які поєднують генерацію задач із змінними параметрами, автоматизовану перевірку відповідей, багаторівневий механізм підказок та адаптивне керування навчальною траєкторією на основі інтерпретованих моделей відстеження знань. Саме ця прогалина зумовлює актуальність розроблення прототипу як основи для подальших емпіричних та архітектурних досліджень.

1.3. Мета і завдання дослідження

Метою дослідження є обґрунтування архітектури, функціональної реалізації та сценарної логіки прототипу інтелектуальної вебсистеми навчання математиці, орієнтованої на параметризовану генерацію задач, автоматизовану перевірку відповідей, адаптивну підтримку користувача та формування індивідуальної навчальної траєкторії.

Особливістю дослідження є поєднання адаптивних моделей відстеження знань, зокрема байєсівської моделі відстеження знань (Bayesian Knowledge Tracing, BKT), генерації варіативного навчального контенту та використання засобів генеративного штучного інтелекту в межах єдиної веборієнтованої архітектури.

Для досягнення поставленої мети доцільно розв'язати такі завдання:

- обґрунтувати архітектуру веборієнтованої інтелектуальної навчальної системи та визначити взаємодію її основних компонентів;
- сформулювати модель предметної області, включаючи структуру навчального контенту та граф знань із пререквізитними зв'язками;
- дослідити підходи до параметризованої генерації математичних задач та автоматизованої перевірки відповідей;
- розробити та формалізувати адаптивний навчальний сценарій на основі моделей відстеження знань;
- проаналізувати можливості інтеграції генеративних моделей штучного інтелекту для автоматизації створення навчального контенту;
- дослідити рішення користувацького інтерфейсу та механізми взаємодії користувача із системою;
- оцінити переваги, обмеження та напрями подальшого розвитку прототипу.

2. Матеріали та методи дослідження

Матеріалом дослідження є публічний вебпрототип, доступний за адресою <https://khai-edu.github.io/its-mugna/>. Прототип реалізує не лише клієнтський інтерфейс, а й серверну логіку керування навчальним процесом, що забезпечує параметризовану генерацію задач, автоматизовану перевірку відповідей, збереження стану користувача та накопичення навчальної аналітики. Це дозволяє розглядати систему не як статичний демонстраційний вебресурс, а як функціональний прототип інтелектуальної навчальної системи.

Прототип репрезентує навчальне середовище для роботи з математичними задачами, у якому користувач взаємодіє з системою через браузерний інтерфейс, а логіка подання задач організована як послідовність шаблонізованих кроків. Згідно з реалізаційним описом прототипу, клієнтську частину побудовано на Angular, що є доцільним для односторінкових навчальних застосунків із динамічними станами інтерфейсу.

Методологічно дослідження спирається на системний аналіз, функціональну декомпозицію програмного прототипу, порівняльний огляд підходів до інтелектуальних навчальних систем та відстеження знань, а також на принципи комп'ютерного оцінювання у математичній освіті [2–10].

Оскільки поточна публікація присвячена насамперед архітектурному та сценарному опису прототипу, емпірична педагогічна перевірка ефективності на контрольних та експериментальних групах розглядається як окремий наступний етап дослідження.

2.1. Архітектурне представлення прототипу

Архітектуру доцільно описувати як багаторівневу структуру, у якій клієнтський інтерфейс, генерація задач, перевірка відповідей, підказки та облік навчальної взаємодії утворюють замкнений цикл. На рівні користувача система функціонує як односторінковий застосунок: зміна стану задачі, повідомлення щодо правильності відповіді, перехід до підказок або наступного кроку відбуваються без виходу зі сценарію поточної сесії. На логічному рівні це означає, що навчальна задача подається не як статичний фрагмент контенту, а як об'єкт із параметрами, критеріями перевірки та набором можливих реакцій системи.

У поточній реалізації ця структура набуває форми клієнт-серверної архітектури, у якій клієнтський інтерфейс відповідає за навчальну взаємодію користувача, а серверний рівень – за генерацію задач, перевірку відповідей, збереження даних, обчислення показників засвоєння та підтримку адаптивного сценарію. Такий підхід забезпечує масштабованість прототипу та створює основу для переходу від локальної сценарної логіки до повноцінної інтелектуальної навчальної системи.

З погляду функціональної декомпозиції прототип містить п'ять ключових підсистем: (1) клієнтський рівень взаємодії, (2) модуль шаблонної генерації задач, (3) модуль перевірки відповіді, (4) механізм підказок, (5) рівень збереження стану та аналітики взаємодії. Кожен із цих компонентів може бути реалізований як окремий сервіс або як сукупність

пов'язаних класів/модулів у межах єдиного застосування. Для досліджуваного прототипу принципово важливо, що модульність архітектури допускає як локальне розширення окремих частин, так і поступову міграцію до більш складної серверної моделі без повного переписування клієнтського рівня.

Основні функціональні компоненти прототипу та їх призначення представлено в таблиці 1.

Подана декомпозиція не прив'язує систему до єдиного способу реалізації обчислювальної логіки. У найпростішому варіанті генератор задач і модуль перевірки можуть працювати в межах клієнтського застосунка, якщо йдеться про легкі задачі з наперед визначеним форматом відповіді. У більш складному варіанті ці модулі доцільно винести на серверний рівень, особливо якщо перевірка пов'язана з символічними перетвореннями, кількома допустимими формами відповіді або потребує накопичення аналітики на рівні багатьох користувачів. Отже, архітектура системи є масштабованою не в сенсі наявності вже реалізованої складної інфраструктури, а в сенсі сумісності з нею.

2.2. Інтерфейс системи

Інтерфейс виконує не декоративну, а безпосередньо дидактичну функцію. Для навчання математиці критично важливо, щоб користувачеві були одночасно доступні формулювання задачі, поле введення відповіді, інформація про стан перевірки та можливі підказки. Фрагментація інтерфейсу призводить до зайвого когнітивного навантаження, тоді як цілісне

Таблиця 1

Функціональні компоненти прототипу

Компонент	Основна функція	Опрацьовувані дані	Роль в адаптивності
Клієнтський рівень (Angular)	Відображення задачі, навігація, введення відповіді, показ повідомлень і підказок	Поточний стан сесії, параметри задачі, введені користувачем значення	Забезпечує безперервну інтерактивну взаємодію та зміну сценарію без перезавантаження сторінки
Шаблонна генерація задач	Побудова нового варіанта задачі на основі шаблону та набору параметрів	Шаблони формулювань, домени параметрів, обмеження коректності	Дозволяє варіювати однотипні задачі й підтримувати контрольовану складність
Модуль валідації відповіді	Перевірка введеного результату або кроку розв'язування	Математичний вираз, числовий результат, допустимі форми запису	Є підставою для рішення щодо підказки, повтору або переходу далі
Механізм підказок	Надання контекстного зворотного зв'язку залежно від помилки або стану користувача	Тип відповіді, номер спроби, категорія задачі, обраний режим підтримки	Диференціює ступінь допомоги і формує локальну педагогічну реакцію
Рівень аналітики / модель користувача	Накопичення даних про хід роботи та обчислення оцінки засвоєння	Історія відповідей, час виконання, переходи між задачами, успішність	Підтримує вибір наступного кроку, складності та інтенсивності підказок

робоче поле знижує когнітивні витрати взаємодії. Публічний прототип позначає систему як єдиний освітній простір, де робота з окремою задачею є центральним сценарієм. Загальний вигляд головної сторінки системи показано на рис. 1.

Головна сторінка повинна забезпечувати швидкий вхід у навчальний сценарій, розуміння призначення системи та можливість переходу до задач без тривалої навігації. Для прототипів освітніх систем це особливо важливо, оскільки порогове навантаження на користувача не має перевищувати складність самої навчальної дії. Безпосередній екран роботи з задачею, де реалізовано цілісне робоче поле для розв'язування, подано на рис. 2.

Робоче поле задачі має поєднувати текст умови, елементи введення математичної відповіді, кнопку перевірки та зони для оперативного повідомлення користувача про результат поточної спроби. Така композиція підтримує короткий цикл «спроба – перевірка – корекція», який є принциповим для формувального тренування. Реалізацію цього підходу в робочому полі задачі, а також механізм надання контекстних підказок у процесі розв'язування, показано на рис. 3.

Після некоректної відповіді автоматично активується один із рівнів підказки, залежно від історії

спроб і поточного стану засвоєння. Це дозволяє користувачеві отримувати допомогу саме в момент, коли вона найбільш ефективна – не раніше і не пізніше. Важливим є те, що підказки не просто статичні тексти, а контекстні повідомлення, які можуть включати як пояснення помилки, так і підказки щодо наступного кроку. Це робить їх частиною адаптивного навчального процесу, а не лише допоміжним елементом. Підказки не повинні дублювати повне розв'язання шоразу; доцільно організувати їх ієрархічно – від короткої вказівки на тип помилки до більш докладного пояснення. Така логіка дозволяє уникати надмірної допомоги і зберігає пізнавальну активність користувача.

Особливе значення для математичного інтерфейсу має спосіб введення формул. Якщо система працює лише з текстовим рядком без структурованої підтримки математичної нотації, кількість помилок введення швидко зростає і змішується з власне предметними помилками.

Тому інтерфейс слід розглядати не лише як візуальний шар, а як засіб керування якістю навчальних даних: добре спроектоване поле введення зменшує шум, покращує валідацію відповіді та робить аналітику прогресу більш інформативною.

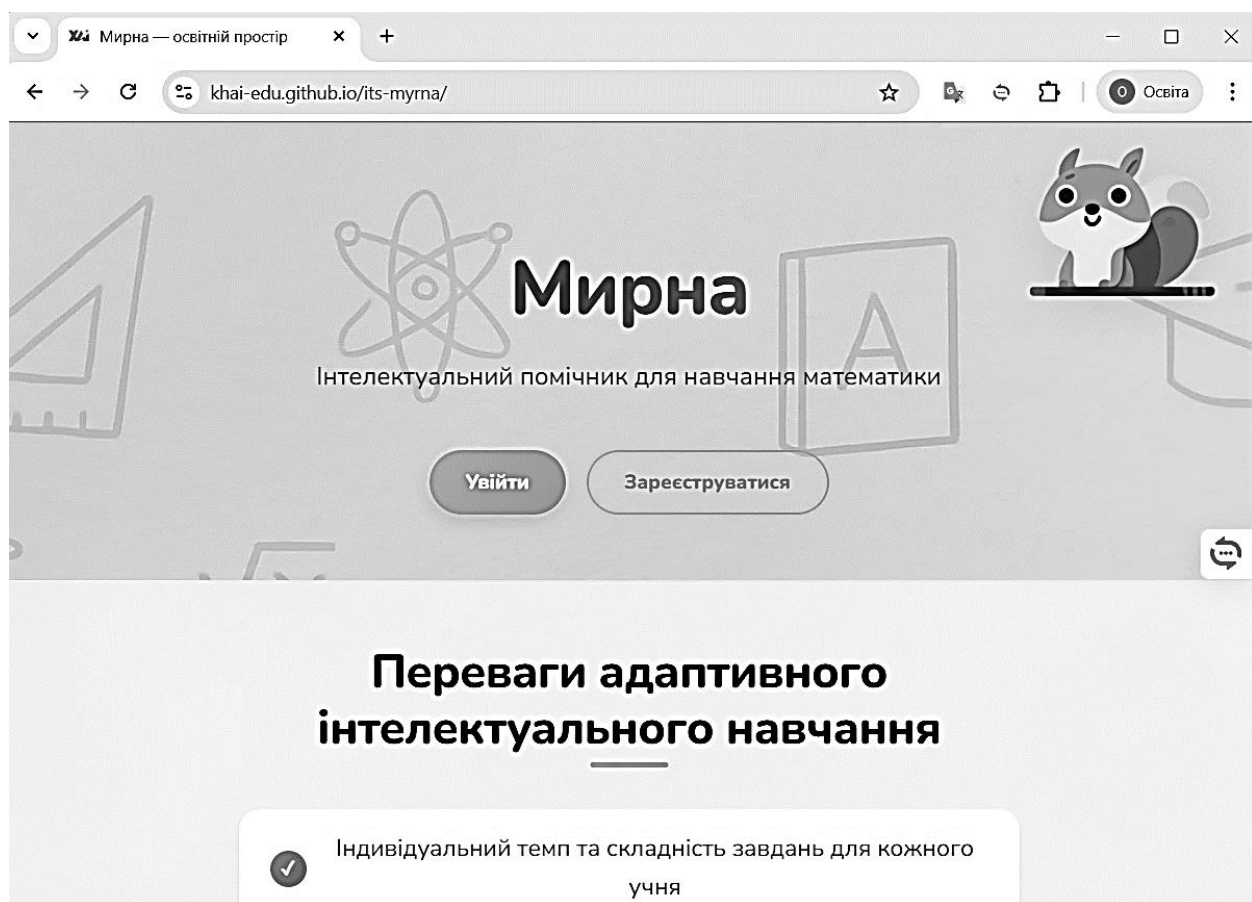


Рис. 1. Головна сторінка системи

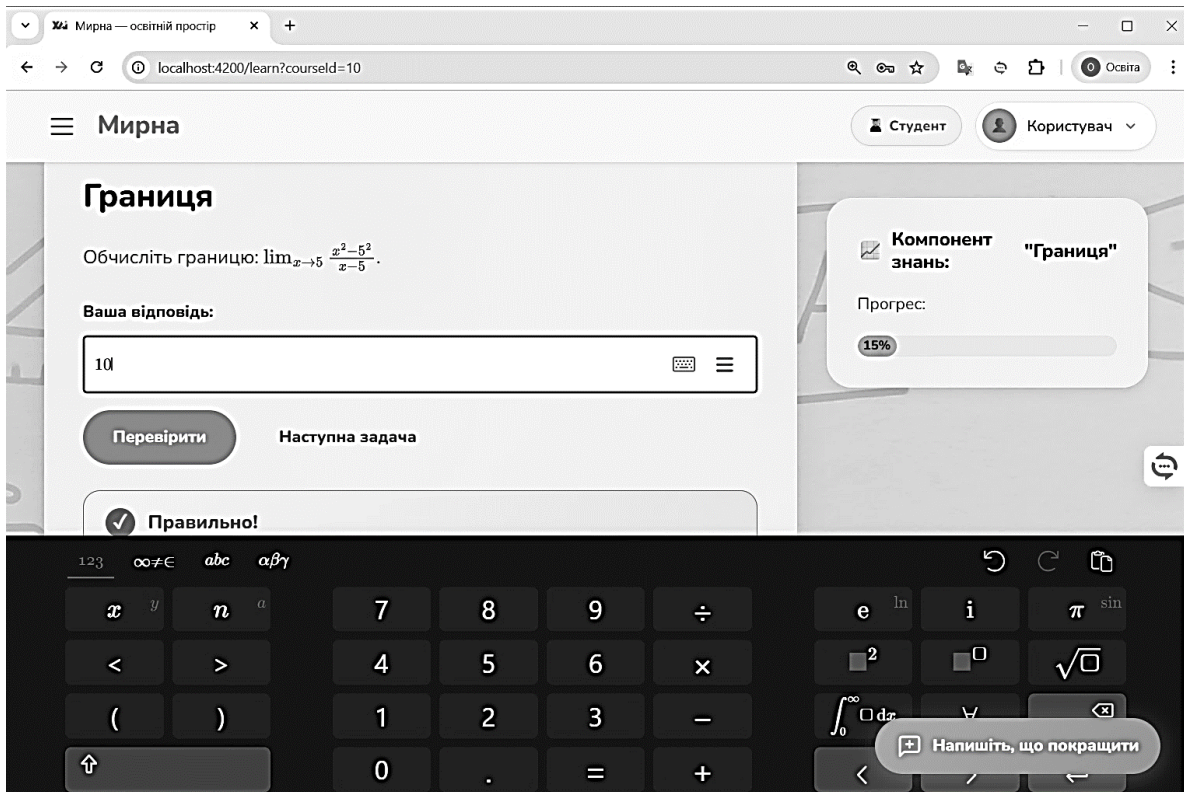


Рис. 2. Інтерфейс розв'язування математичної задачі

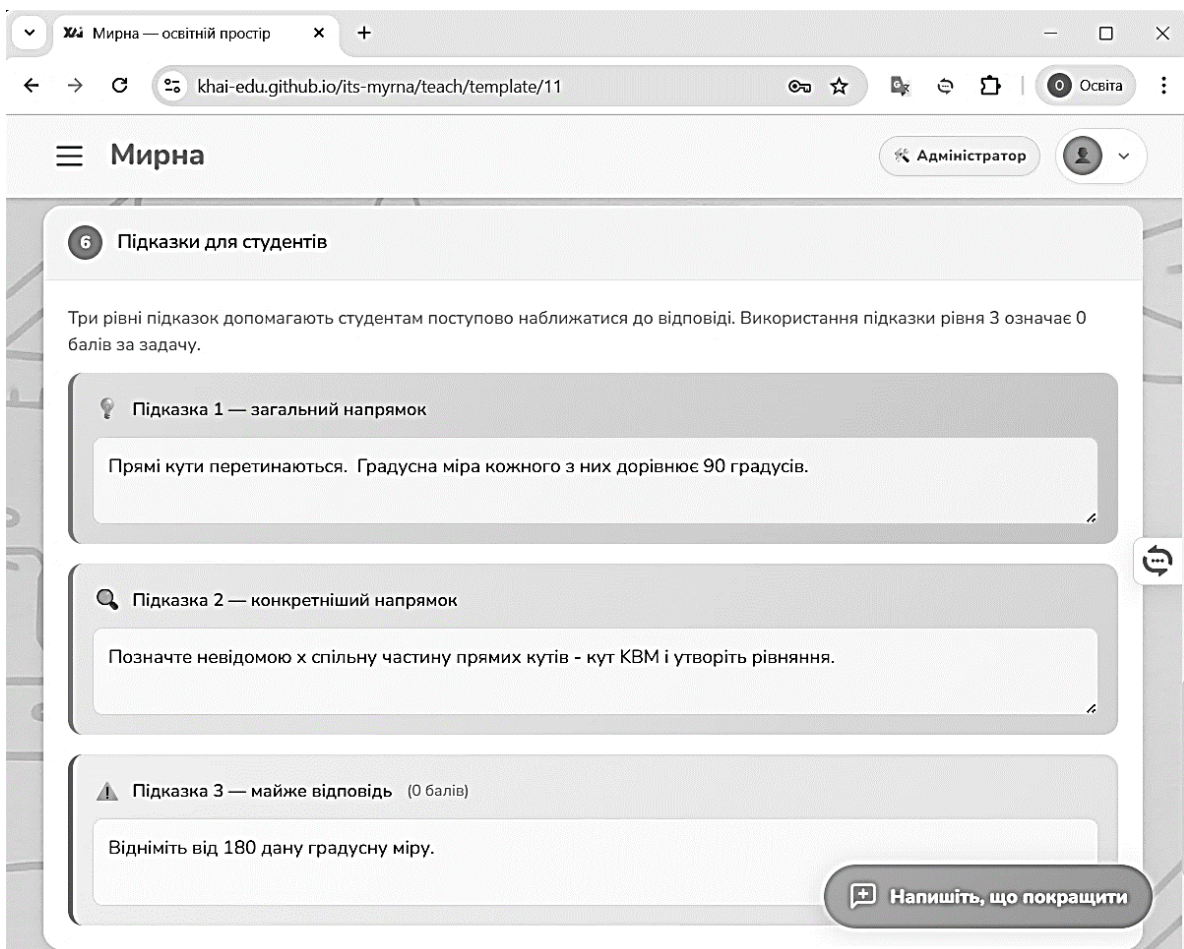


Рис. 3. Механізм підказок у робочому сценарії

2.3. Формалізація адаптивного навчального сценарію та генерації задач

Підходи до формалізації математичних задач у інтелектуальних навчальних системах розглядаються у [13], де задача описується як сукупність правил, вхідних даних, відповідей та алгоритмів розв'язування. Такий формальний підхід дозволяє систематизувати генерацію задач і їх автоматизовану перевірку.

Для переходу від звичайного тренажера до інтелектуальної навчальної системи критичним є механізм, який пов'язує результат поточної дії користувача з вибором наступного кроку. У найпростішому випадку така логіка може бути реалізована правилами виду «правильна відповідь → наступна задача», «неправильна відповідь → повтор або підказка». Проте для накопичувального аналізу навчальної взаємодії доцільно використовувати модель, що оцінює ймовірність засвоєння окремого компонента знань.

Нехай L_t – подія «компонент знань засвоєно» перед урахуванням результату відповіді на t -му кроці, а $C_t \in \{0,1\}$ – результат цієї відповіді, де $C_t = 1$ означає правильну відповідь, а $C_t = 0$ – неправильну. Позначимо через $P(L_t)$ поточну ймовірність засвоєння компонента знань; G – ймовірність випадкового правильного вгадування за відсутності знання; S – ймовірність помилки за наявності знання; T – ймовірність навчального приросту після поточної взаємодії. Тоді після правильної відповіді оновлена ймовірність засвоєння компонента може бути обчислена за формулою класичної моделі ВКТ [5]:

$$P(L_t|C_t = 1) = \frac{P(L_t)(1-S)}{P(L_t)(1-S)+(1-P(L_t))G} \quad (1)$$

Формула (1) задає апостеріорну ймовірність засвоєння після правильної відповіді. Її чисельник $P(L_t)(1-S)$ описує ситуацію, коли користувач уже володіє відповідним компонентом знань і при цьому не допускає помилки. Знаменник є повною ймовірністю правильної відповіді: вона могла бути отримана або завдяки реальному засвоєнню матеріалу, або внаслідок випадкового вгадування за відсутності знання.

Після неправильної відповіді використовується відповідне байєсівське оновлення:

$$P(L_t|C_t = 0) = \frac{P(L_t)S}{P(L_t)S+(1-P(L_t))(1-G)} \quad (2)$$

Формула (2) описує апостеріорну ймовірність засвоєння після неправильної відповіді. У чисельнику $P(L_t)S$ враховано випадок, коли знання вже сформоване, але користувач припускається помилки. Знаменник відображає повну ймовірність неправильної відповіді: вона може виникнути як через помилку

за наявності знання, так і через відсутність знання без випадкового вгадування.

Після цього оцінка знання переноситься на наступний крок з урахуванням навчального приросту:

$$P(L_{t+1}) = P(L_t|C_t) + (1 - P(L_t|C_t))T. \quad (3)$$

Формула (3) відображає можливий навчальний ефект поточної взаємодії. Якщо навіть після врахування відповіді компонент ще не вважається повністю засвоєним, система допускає приріст знання з ймовірністю T . На практиці у формулі (3) використовується значення, отримане за формулою (1) або (2), залежно від того, була відповідь правильною чи неправильною.

У межах запропонованої системи ці співвідношення дозволяють пов'язати історію відповідей із вибором наступної дії. Зокрема, оновлена оцінка $P(L_{t+1})$ може використовуватися як параметр сценарного керування та генерації наступної задачі: за низьких значень доцільним є повернення до простішого варіанта або активація розгорнутої підказки; за середніх – подання нового варіанта задачі того самого типу; за високих – перехід до складнішого матеріалу або до нового підтипу задач.

Окрім ВКТ, для більш складних сценаріїв можуть розглядатися нейромережеві моделі відстеження знань [6, 8], однак на етапі прототипування інтерпретованість класичного підходу має перевагу. Для математичного навчання, де викладач і розробник системи прагнуть бачити причину зміни педагогічного рішення, прозора модель часто є практично кориснішою за складнішу модель з вищою прогностичною точністю, але нижчою пояснюваністю. Правила адаптації навчальної траєкторії на основі поточної оцінки засвоєння компонента знань наведено в таблиці 2.

3. Результати та обговорення

Головним результатом дослідження є цілісне архітектурно-сценарне представлення та реалізація прототипу як вебсистеми для навчання математиці, у якій об'єднано параметризовану генерацію задач, автоматичну перевірку відповіді, механізм підказок, адаптивне оновлення оцінки засвоєння та правила переходу до наступної задачі. На відміну від статичних електронних курсів, система орієнтується на одиницю дії – конкретну задачу, її поточний стан і реакцію системи на результат цієї взаємодії.

3.1. Приклад навчального сценарію

Типовий сценарій роботи користувача системи може бути описаний як п'ятиетапний цикл.

– Генерація задачі. Після входу до навчального середовища користувач отримує параметризований варіант задачі, сформований на основі шаблону та, в адаптивному режимі, обраний з урахуванням поточного рівня засвоєння відповідного компонента знань.

Це знижує ризик механічного відтворення попередніх відповідей і дає змогу зберігати однакову дидактичну мету за змінних числових параметрів.

– Введення відповіді. Користувач працює в єдиному робочому полі, де бачить умову задачі, елементи введення математичної відповіді та засоби підтвердження спроби. Якщо відповідь подається у формульному вигляді, інтерфейс має мінімізувати синтаксичний шум.

– Перевірка. Після надсилання відповіді система запускає модуль валідації. Результатом є або підтвердження правильності, або повідомлення про невідповідність. На цьому етапі важливо відокремлювати предметну помилку від помилки формату введення.

У разі помилки система може не лише фіксувати невідповідність, а й зіставляти відповідь із типізованими діагностичними правилами, що дає змогу уточнювати характер помилкової стратегії.

– Підказки. Якщо відповідь некоректна, активується багаторівневий механізм підтримки: коротка вказівка, уточнювальне повідомлення або більш розгорнуте пояснення. Така ієрархія зближує вебсистему з логікою покрокового навчання (step-based tutoring) [3].

– Перехід до наступної задачі. Вибір наступного кроку виконується за правилом адаптивного сценарію: повторення, варіативне тренування на тому самому рівні або перехід до складнішого підтипу.

Якщо в системі фіксується історія відповідей, то рішення спирається на оцінку засвоєння, яка оновлюється після кожної взаємодії за формулами (1)–(3), та використовується для вибору наступної задачі, повторення матеріалу або переходу до складнішого підтипу.

Загальну послідовність взаємодії компонентів системи в адаптивному сценарії показано на рис. 4.

З погляду педагогічної логіки такий цикл є важливим тим, що зберігає цілісність навчальної взаємодії. Користувач не залишає робочий контекст, не перемикається між кількома розрізненими сервісами і не чекає відкладеного оцінювання. Саме швидка реакція системи на поточну спробу створює відчуття індивідуального супроводу, що є ядром ідеї інтелектуального викладача.

У реалізованому варіанті навчального сценарію користувачеві також відображається поточний стан прогресу, що дозволяє пов'язати окрему відповідь не

лише з локальним результатом перевірки, а й із загальною динамікою засвоєння компонентів знань. Це посилює мотиваційний аспект взаємодії та підтримує усвідомленість навчального процесу.

3.2. Переваги прототипу

Порівняно з традиційними електронними матеріалами або системами, у яких завдання є разовими та мало варіативними, авторська система має кілька принципових переваг.

По-перше, шаблонізація задач підтримує повторюваність тренування без дублювання однакових числових прикладів.

По-друге, модуль перевірки відповіді вбудований у навчальний сценарій і тому виконує не лише контрольну, а й керувальну функцію.

По-третє, інтерфейс підказок може бути прив'язаний до номера спроби і рівня засвоєння, а не бути статичною довідкою.

По-четверте, публічне веброзгортання знижує поріг доступу до системи та полегшує майбутнє збирання навчальної аналітики.

По-п'яте, система не лише фіксує результати виконання задач, а й підтримує ймовірнісне оцінювання рівня засвоєння через ВКТ, що дозволяє перейти від разового контролю до динамічного моделювання навчального прогресу.

По-шосте, структуризація навчального матеріалу через компоненти знань і пререквізитні зв'язки створює основу для обґрунтованої адаптації та візуального представлення індивідуального прогресу користувача.

По-сьоме, інтеграція генеративних моделей штучного інтелекту відкриває можливість автоматизованої підтримки створення шаблонів задач, підказок і попередньої структури графа знань, що знижує трудомісткість підготовки навчального контенту. По-восьме, прототип підтримує рольову організацію навчального середовища, що робить його придатним не лише для індивідуального експериментування, а й для використання в реальному курсі з викладачами, студентами та групами.

Наукова новизна запропонованого підходу в межах даної публікації полягає в узгодженні кількох раніше часто розділених площин: архітектури вебзастосунку, дидактичної логіки математичних вправ, ймовірнісної моделі засвоєння знань, механізму діагностичного зворотного зв'язку, візуалізації структури знань та засобів автоматизованої підтримки створення контенту. Саме ця узгодженість робить прототип придатним до подальшого розвитку в сторону повноцінної ITS, а не лише до покращення окремих інтерфейсних чи сервісних компонентів.

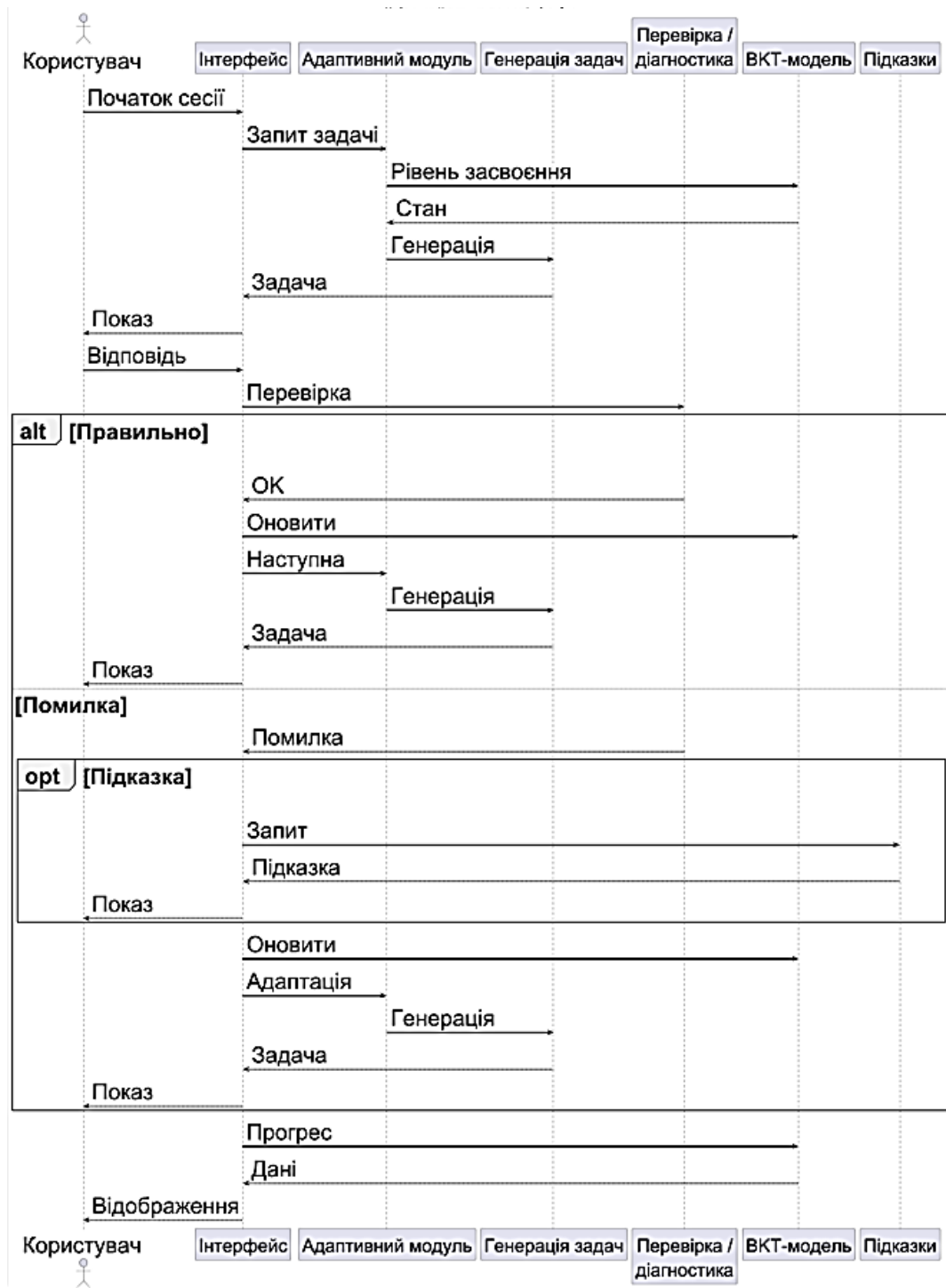


Рис. 4. Діаграма навчального сценарію системи

3.3. Обмеження та напрями масштабування

Разом з тим поточний стан прототипу має і суттєві обмеження. У межах даного рукопису не подано результатів контрольованого експерименту з вимірюванням навчальних приростів, тому педагогічна ефективність системи поки що обґрунтовується архітектурно та теоретично, а не статистично. Крім того,

публічний вебпрототип не дає повної інформації про серверну реалізацію, стратегії збереження даних, повноту покриття типів математичних задач і глибину формальної перевірки еквівалентних відповідей. Зокрема, на поточному етапі перевірка формульних відповідей не повною мірою враховує математичну еквівалентність різних записів одного й того самого виразу, що створює ризик помилково негативних результатів.

Масштабування системи до повноцінного дослідницько-освітнього середовища передбачає щонайменше три напрями розвитку.

Перший – розширення моделі користувача з переходом від базової реалізації відстеження знань до більш складних моделей, що враховують індивідуальні параметри користувача, часові ефекти забування та складніші залежності між компонентами знань [5, 6, 8].

Другий – інтеграція з інструментами символічної математики для перевірки еквівалентних форм відповіді та аналізу проміжних кроків.

Третій – поєднання з інструментами формальної перевірки математичних міркувань. Зокрема, система Lean створює технічну основу для автоматизованої перевірки доказів і поетапного контролю логічної коректності [11]. Останні освітні дослідження показують, що системи формального доведення (proof assistants) можуть стати не лише інструментом формалізації, а й засобом навчання доказу через негайний зворотний зв'язок [12]. Четвертий – розвиток AI-підтримки створення навчального контенту, включаючи автоматичне формування варіантів задач, попереднє генерування діагностичних правил та адаптивне формулювання підказок з урахуванням типових помилок користувача.

У перспективі це відкриває можливість побудови єдиного середовища, де обчислювальні задачі, алгебраїчні перетворення та задачі на доведення підтримуються спільною аналітичною моделлю навчання. Для дисертаційного дослідження такий шлях є особливо продуктивним, оскільки дозволяє поєднати класичні підходи ITS із сучасними можливостями AI-асистованої формальної математики.

3.4. Критерії емпіричної перевірки

Оскільки поточна публікація має архітектурно-проектний характер, наступним логічним кроком є

емпірична перевірка того, як система впливає на навчальні результати, темп проходження вправ і характер помилок користувачів. Для математичних вебсистем недостатньо обмежуватися лише фінальною оцінкою; не менш важливими є показники проміжної успішності, кількість повторних спроб, інтенсивність звернення до підказок та стійкість результату при відкладеному контролі.

Оптимальним дизайном такої перевірки є квазі-експеримент із до- та післятестовим вимірюванням у контрольній та експериментальній групах. Експериментальна група працює з адаптивним сценарієм системи, тоді як контрольна – з неадаптивним набором тих самих типів задач. За наявності журналів подій доцільно аналізувати не лише підсумкові бали, а й траєкторії взаємодії: число і тип помилок, середній час на задачу, переходи між рівнями складності, частоту використання підказок та зміну оцінки засвоєння впродовж сесії. Перелік ключових показників для експериментальної перевірки прототипу та їх очікувані ефекти наведено в таблиці 3.

Окремого виміру потребує користувацька якість інтерфейсу. Для вебсистем навчання математиці вона впливає не опосередковано, а безпосередньо, оскільки погано спроектоване поле введення або заплутана послідовність дій збільшує частку технічних, а не предметних помилок. Тому перевірка системи має поєднувати дидактичні, аналітичні та UX-показники.

Запропонований набір показників важливий ще й тому, що дозволяє розрізнити навчальну та інтерфейсну ефективність. Якщо підсумковий приріст є невеликим, але кількість технічних помилок введення зменшується, то це може свідчити про потребу зміни контентної частини, а не повного перегляду архітектури. І навпаки, високий предметний результат за низьких UX-оцінок означатиме, що система ефективна педагогічно, але ще не оптимізована як масовий вебпродукт.

Для адаптивних навчальних систем особливо

Таблиця 3

Показники для експериментальної перевірки прототипу

Показник	Спосіб вимірювання	Джерело даних	Інтерпретація
Навчальний приріст	Різниця між до- і післятестом	Результати тематичних тестів	Безпосередній ефект використання системи
Час на задачу	Середній час виконання одного варіанта	Журнали подій	Операційна складність і зручність сценарію
Інтенсивність підказок	Частка задач, у яких активовано допомогу	Журнали подій	Потреба в підтримці та рівень автономності
Стійкість засвоєння	Відкладений контроль через певний інтервал	Повторний тест	Тривалість збереження результату
Динаміка оцінки засвоєння	Зміна оцінки засвоєння впродовж сесії	Журнали взаємодії / модель користувача	Швидкість та стійкість набуття окремих компонентів знань
Задоволеність інтерфейсом	Опитувальник SUS / інтерв'ю	Анкети користувачів	Придатність інтерфейсу до масштабування

важливим є аналіз не лише фінального приросту, а й внутрішньої динаміки навчальної траєкторії. Саме тому емпірична валідація системи має враховувати зміну оцінки засвоєння впродовж серії задач, характер використання підказок та структуру помилкових стратегій. Це дозволить оцінити не лише результативність системи, а й якість її адаптивного керування.

3.5. Перевірка реалізації та відтворюваність

Для підвищення відтворюваності результатів подальше дослідження має спиратися на документований карту модулів прототипу, яка дозволить чітко співвіднести архітектурні твердження статті з фактично реалізованими компонентами системи. Це особливо важливо для перевірки того, як у кодовій структурі реалізовано генерацію задач, перевірку відповідей, логіку підказок, адаптивні переходи та накопичення навчальної аналітики.

4. Висновки

Запропонований опис прототипу дає підстави розглядати систему як перспективну основу для інтелектуалізації вебнавчання математиці. Архітектурна декомпозиція показує, що навіть на рівні публічного прототипу доцільно розрізняти клієнтський інтерфейс, генерацію задач, перевірку відповідей, механізм підказок та рівень аналітики, оскільки саме зв'язок між цими компонентами формує основу адаптивного навчального сценарію.

Формалізація сценарію через оцінку стану засвоєння дозволяє перейти від простих правил переходу між задачами до інтерпретованої моделі керування навчальною траєкторією. У цьому контексті ВКТ є методологічно доречним стартовим варіантом для подальшого розвитку системи, оскільки поєднує достатню виразність з прозорістю педагогічного тлумачення.

Практична цінність дослідження полягає в тому, що система вже існує як доступний вебпрототип і може бути використана як база для наступних етапів: кодувального аудиту реалізації, експериментальної перевірки ефективності, інтеграції з системами символічної математики та розширення до середовища навчання формальних доведень.

Основні результати дослідження можна стисло підсумувати так:

- обґрунтовано архітектурну структуру прототипу як веборієнтованої математичної ITS;
- описано інтерфейсні рішення, релевантні для розв'язування математичних задач і надання підказок;

- запропоновано формалізацію адаптивного сценарію на основі оцінки засвоєння компонента знань;

- визначено перспективні напрями масштабування прототипу в бік AI-інтеграції та формальної перевірки математичних міркувань.

Внесок авторів: формулювання задач – Олександр Євдокимов, реалізація вебсистеми – Олександр Євдокимов; аналіз джерел – Оксана Лучшева, написання чернетки статті – Олександр Євдокимов, написання остаточного варіанту статті – Оксана Лучшева.

Конфлікт інтересів

Автори заявляють, що у них немає конфлікту інтересів щодо цього дослідження, фінансового, особистого, авторського чи іншого, який міг би вплинути на дослідження та його результати, представлені в цій статті.

Фінансування

Дослідження проводилося без фінансової підтримки.

Доступність даних

Публічний прототип доступний за адресою <https://khai-edu.github.io/its-myrna/>. Додаткові матеріали щодо архітектурного опису та кодувальної верифікації можуть бути надані автором за обґрунтованим запитом.

Використання штучного інтелекту

Під час підготовки початкової чернетки рукопису були використані генеративні інструменти штучного інтелекту для допоміжного мовного редагування та структуризації тексту; перевірка фактів, інтерпретація результатів і фінальне наукове редагування мають бути виконані автором перед поданням.

Усі автори прочитали і погодили остаточну версію рукопису.

Література

1. Bloom, B. S. *The 2 sigma problem: the search for methods of group instruction as effective as one-to-one tutoring [Text]* / B. S. Bloom // *Educational Researcher*. – 1984. – Vol. 13, № 6. – P. 4–16.
2. Woolf, B. P. *Building intelligent interactive tutors: student-centered strategies for revolutionizing e-learning [Text]* / B. P. Woolf. – Burlington : Morgan Kaufmann, 2010. – 480 p.

3. VanLehn, K. *The relative effectiveness of human tutoring, intelligent tutoring systems, and other tutoring systems [Text]* / K. VanLehn // *Educational Psychologist*. – 2011. – Vol. 46, № 4. – P. 197–221. DOI: 10.1080/00461520.2011.611369.

4. Ma, W. *Intelligent tutoring systems vs. other tutoring systems: a meta-analysis [Text]* / W. Ma, O. O. Adesope, J. C. Nesbit, Q. Liu // *Educational Psychologist*. – 2014. – Vol. 49, № 4. – P. 249–266. DOI: 10.1080/00461520.2014.965891.

5. Corbett, A. T. *Knowledge tracing: modeling the acquisition of procedural knowledge [Text]* / A. T. Corbett, J. R. Anderson // *User Modeling and User-Adapted Interaction*. – 1995. – Vol. 4, № 4. – P. 253–278. DOI: 10.1007/BF01099821.

6. Piech, C. *Deep knowledge tracing [Text]* / C. Piech, J. Bassen, J. Huang, S. Ganguli, M. Sahami, L. J. Guibas, C. Guestrin // *Advances in Neural Information Processing Systems 28 (NIPS 2015)*. – 2015. – P. 505–513.

7. Brusilovsky, P. *Adaptive hypermedia [Text]* / P. Brusilovsky // *User Modeling and User-Adapted Interaction*. – 2001. – Vol. 11. – P. 87–110. DOI: 10.1023/A:1011143116306.

8. Shen, S. *A survey of knowledge tracing: models, variants, and applications [Text]* / S. Shen, F. Liu, H. Weng, Z. Yu // *IEEE Transactions on Learning Technologies*. – 2021. – Vol. 14, № 2. – P. 155–171. DOI: 10.1109/TLT.2021.3058565.

9. Sangwin, C. *Computer aided assessment of mathematics [Text]* / C. Sangwin. – Oxford : Oxford University Press, 2013. – 314 p.

10. Greenhow, M. *Effective computer-aided assessment of mathematics: principles, practice and results [Text]* / M. Greenhow // *Teaching Mathematics and Its Applications*. – 2015. – Vol. 34, № 3. – P. 117–137. DOI: 10.1093/teamat/hrv012.

11. de Moura, L. *The Lean Theorem Prover (System Description) [Text]* / L. de Moura, S. Kong, J. Avigad, F. van Doorn, J. von Raumer // *Automated Deduction – CADE-25 : LNCS 9195*. – Cham : Springer, 2015. – P. 378–388. DOI: 10.1007/978-3-319-21401-6_26.

12. Hanna, G. *Using the Proof Assistant Lean in Undergraduate Mathematics Classrooms [Text]* / G. Hanna, B. Larvor, X. K. Yan // *ZDM – Mathematics Education*. – 2024. – Vol. 56, № 7. – P. 1517–1529. DOI: 10.1007/s11858-024-01577-9.

13. Чухрай, А. *Можливості використання інтелектуальних навчальних систем (ITS) у курсах вищої математики [Текст]* / А. Чухрай, Т. Столяренко, О. Євдокимов, В. Дем'яненко // *Відкриті інформаційні та комп'ютерні інтегровані технології*. – 2025. – № 102. – С. 92–119. DOI: 10.32620/oikit.2024.102.07.

14. Kulik, A. *The concept of intelligent training system for Ukrainian school final STEM exam*

preparation [Text] / A. Kulik, O. Zeleniak, A. Chukhray, O. Prokhorov, O. Yashyna, O. Havrylenko, O. Yevdokymov, A. Torzhkov, O. Zayarnyi // *System research and information technologies*. – 2025. – P. 125–138. DOI: 10.20535/SRIT.2308-8893.2025.2.09.

References

1. Bloom B. S. The 2 sigma problem: the search for methods of group instruction as effective as one-to-one tutoring. *Educational Researcher*, 1984, vol. 13, no. 6, pp. 4–16.

2. Woolf B. P. *Building intelligent interactive tutors: student-centered strategies for revolutionizing e-learning*. Morgan Kaufmann, Burlington, 2010. 480 p.

3. VanLehn K. The relative effectiveness of human tutoring, intelligent tutoring systems, and other tutoring systems. *Educational Psychologist*, 2011, vol. 46, no. 4, pp. 197–221. DOI: 10.1080/00461520.2011.611369.

4. Ma W., Adesope O. O., Nesbit J. C., Liu Q. Intelligent tutoring systems vs. other tutoring systems: a meta-analysis. *Educational Psychologist*, 2014, vol. 49, no. 4, pp. 249–266. DOI: 10.1080/00461520.2014.965891.

5. Corbett A. T., Anderson J. R. Knowledge tracing: modeling the acquisition of procedural knowledge. *User Modeling and User-Adapted Interaction*, 1995, vol. 4, no. 4, pp. 253–278. DOI: 10.1007/BF01099821.

6. Piech C., Bassen J., Huang J., Ganguli S., Sahami M., Guibas L. J., Guestrin C. Deep knowledge tracing. *Advances in Neural Information Processing Systems 28 (NIPS 2015)*, 2015, pp. 505–513.

7. Brusilovsky P. Adaptive hypermedia. *User Modeling and User-Adapted Interaction*, 2001, vol. 11, pp. 87–110. DOI: 10.1023/A:1011143116306.

8. Shen S., Liu F., Weng H., Yu Z. A survey of knowledge tracing: models, variants, and applications. *IEEE Transactions on Learning Technologies*, 2021, vol. 14, no. 2, pp. 155–171. DOI: 10.1109/TLT.2021.3058565.

9. Sangwin C. *Computer aided assessment of mathematics*. Oxford University Press, Oxford, 2013. 314 p.

10. Greenhow M. Effective computer-aided assessment of mathematics: principles, practice and results. *Teaching Mathematics and Its Applications*, 2015, vol. 34, no. 3, pp. 117–137. DOI: 10.1093/teamat/hrv012.

11. de Moura L., Kong S., Avigad J., van Doorn F., von Raumer J. The Lean Theorem Prover (System Description). In: *Automated Deduction – CADE-25. LNCS 9195*. Cham, Springer, 2015, pp. 378–388. DOI: 10.1007/978-3-319-21401-6_26.

12. Hanna G., Larvor B., Yan X. K. Using the Proof Assistant Lean in Undergraduate Mathematics

Classrooms. *ZDM – Mathematics Education*, 2024, vol. 56, no. 7, pp. 1517–1529. DOI: 10.1007/s11858-024-01577-9.

13. Chukhray, A., Stoliarenko, T., Yevdokymov, O., Demyanenko, V. *Mozhlyvosti vykorystannya intelektualnykh navchalnykh system (ITS) u kursakh vyshchoi matematyky [Possibilities of using intelligent tutoring systems (ITS) in higher mathematics courses]. Vidkryti informatsiyi ta kompyuterni intehrovani*

tekhnohiiyi, 2025, no. 102, pp. 92–119. DOI: 10.32620/oikit.2024.102.07 (In Ukrainian).

14. Kulik A., Zeleniak O., Chukhray A., Prokhorov O., Yashyna O., Havrylenko O., Yevdokymov O., Torzhkov A., Zayarnyi O. The concept of intelligent training system for Ukrainian school final STEM exam preparation. *System research and information technologies*, 2025, pp. 125–138. DOI: 10.20535/SRIT.2308-8893.2025.2.09.

Received 05.01.2026, Received in revised form 10.03.2026

Accepted date 15.04.2026, Published date 22.04.2026

DEVELOPMENT OF A PROTOTYPE INTELLIGENT WEB-BASED SYSTEM FOR LEARNING MATHEMATICS THROUGH AN ADAPTIVE SCENARIO

Oleksandr Yevdokymov, Oksana Luchsheva

The subject of the study is the architecture and functional implementation of a prototype web-oriented intelligent tutoring system for adaptive mathematics learning in an online environment. **The purpose of the work** is to develop and substantiate the concept of an intelligent web-based system that provides parameterized generation of mathematical problems, automated answer validation, a multi-level hint mechanism, and an individual learning trajectory based on a probabilistic knowledge mastery tracing model. **To achieve the goal**, the following tasks were addressed: the analysis of modern approaches to building intelligent tutoring systems and knowledge tracing models; the formation of a domain model using a knowledge graph with prerequisite links; the development of modules for template-based problem generation and automated answer validation; the formalization of an adaptive learning scenario based on Bayesian updating of mastery probability; the design of a three-level contextual hint mechanism; the creation of interface solutions for entering mathematical expressions and providing immediate formative feedback; and the evaluation of the potential use of generative artificial intelligence for automating learning content creation. **The research methods** include systems analysis, functional decomposition, a comparative literature review on intelligent tutoring systems and computer-aided mathematics assessment, the implementation of the classical probabilistic knowledge tracing model, and prototyping of a client-server web system (with an Angular frontend). **Conclusions.** The developed prototype implements a closed-loop adaptive learning cycle that includes dynamic problem generation, answer input and validation, the provision of contextual hints, the updating of knowledge mastery assessment, and the selection of the next learning step. The system is already operational as a public web prototype and creates a foundation for the empirical evaluation of the effectiveness of adaptive mathematics learning. The **scientific novelty** lies in the comprehensive integration within a single web platform of: template-based problem generation, automated validation of mathematical expressions, a hierarchical hint mechanism, a probabilistic knowledge tracing model for trajectory management, a knowledge graph with prerequisites, and the justification for the use of generative artificial intelligence in automating content creation and error diagnostics.

Keywords: intelligent tutoring system; adaptive learning; mathematics education; web-based system; Bayesian knowledge tracing; problem generation; hints; knowledge graph.

Євдокимов Олександр Олегович – асп. каф. математичного моделювання та штучного інтелекту, Національний аерокосмічний університет «Харківський авіаційний інститут», Харків, Україна.

Лучшева Оксана Вадимівна – старш. викл. каф. інженерії програмного забезпечення, Національний аерокосмічний університет «Харківський авіаційний інститут», Харків, Україна.

Oleksandr Yevdokymov – PhD Student of the Department of Mathematical Modeling and Artificial Intelligence, National Aerospace University “Kharkiv Aviation Institute”, Kharkiv, Ukraine, e-mail: o.yevdokymov@khai.edu, ORCID: 0009-0008-9687-6344, Scopus Author ID: 58099243700.

Oksana Luchsheva – Senior Lecturer at the Department of Software Engineering, National Aerospace University “Kharkiv Aviation Institute”, Kharkiv, Ukraine, e-mail: o.luchsheva@khai.edu, ORCID: 0000-0003-3855-2815, Scopus Author ID: 57216484591.