



DOI 10.28925/2663-4023.2026.33.1246

УДК 004.5:004.89:004.41

Бондарчук Андрій Петрович

доктор технічних наук, професор,

завідувач кафедри комп'ютерних наук

Київський столичний університет ім. Бориса Грінченка, Київ, Україна

ORCID: 0000-0001-5124-5102

a.bondarchuk@kubg.edu.ua

Криволапов Гліб Ярославович

Голова Наукового товариства студентів, аспірантів, докторантів і молодих вчених

Факультету інформаційних технологій та математики

Київський столичний університет ім. Бориса Грінченка, Київ, Україна

ORCID: 0000-0002-0853-5881

hykryvolapov.fitm25m@kubg.edu.ua

МОДЕЛЬ ГІБРИДНОЇ ВЗАЄМОДІЇ З ПЛАТФОРМОЮ НА ОСНОВІ AI-АГЕНТА

Анотація. У статті представлено концептуальну модель гібридної взаємодії користувача з цифровою платформою, у якій AI-агент функціонує як інтегрований інтерфейсний шар-оркестратор між людиною та функціональними модулями системи. Дослідження зумовлене кризою класичних графічних інтерфейсів сучасних SaaS-екосистем, де надмірна щільність елементів керування створює критичне когнітивне навантаження та знижує продуктивність користувача. У вступі обґрунтовано актуальність переходу від прямої маніпуляції до взаємодії на основі намірів і сформульовано наукову проблему щодо нерозробленості архітектурних підходів, які б формалізували перетворення природномовного запиту у конкретні виклики системи з відображенням результату у звичних елементах графічного інтерфейсу. У контексті аналізу останніх досліджень і публікацій розглянуто еволюцію парадигм людино-машинної взаємодії, обмеження класичних користувацьких інтерфейсів у контексті закону Хіка, концепцію генеративного інтерфейсу та виявлено принципову відмінність запропонованого підходу від типових діалогових систем і чат-ботів. У розділі результатів дослідження описано тривірневу архітектуру, що охоплює графічний інтерфейс користувача як точку входу й активний сенсор контексту, інтелектуальний інтерфейсний шар із модулями аналізу намірів, планування дій і генерації пояснень, а також серверне ядро з модулями пошуку, фільтрації та порівняння об'єктів. Запропоновано логічну діаграму обробки запиту, діаграму послідовності взаємодії та компонентну модель платформи, які системно описують повний цикл від формулювання користувацького наміру природною мовою до автоматичної мутації стану графічних компонентів через виклики Backend API. Окремо в межах дослідження розроблено ієрархічну класифікацію, що охоплює шість послідовних рівнів інтеграції штучного інтелекту у цифрову платформу, де кожен наступний рівень передбачає глибше залучення AI до функціонування системи, а найвищий рівень відповідає повноцінному інтерфейсному шару-оркестратору, який бере на себе координацію всіх взаємодій між користувачем і платформою. У висновках доведено, що запропонована концептуальна модель забезпечує поєднання інтуїтивності природної мови з прозорістю та контрольованістю графічного інтерфейсу, мінімізує операційне тертя та створює методологічну базу для проектування інформаційних застосунків нового покоління.

Ключові слова: AI-агент; гібридна взаємодія; графічний інтерфейс користувача; взаємодія на основі намірів; генеративний інтерфейс, інформаційна система, штучний інтелект.

ВСТУП

Трансформація сучасних цифрових платформ у складні багатокомпонентні екосистеми зумовлює виникнення кризи традиційних патернів прямої маніпуляції, де надмірна щільність елементів графічного інтерфейсу (GUI) стає бар'єром для продуктивності користувача через критичне когнітивне навантаження. У відповідь на цей виклик пропонується архітектурна модель гібридної взаємодії, у якій



AI-агент функціонує як інтегрований інтерфейсний шар, що здійснює семантичну декомпозицію намірів користувача та їх подальшу трансляцію у виклики функціональних модулів системи. На відміну від ізольованих діалогових систем, така модель забезпечує органічну інтеграцію інтелектуального агента в логіку платформи, де природна мова слугує інструментом оркестрації API, а адаптивні компоненти GUI забезпечують необхідний рівень візуальної верифікації, структурованого виводу даних та операційного контролю.

Постановка проблеми. Постійне розширення функціональних можливостей сучасних цифрових платформ закономірно ускладнює їхнє використання. Традиційні графічні інтерфейси вимагають від користувача точного знання системи та виконання багатокрокових операцій, що призводить до швидкого когнітивного перевантаження та зниження продуктивності роботи [1]. У цьому контексті актуальним завданням стає перехід від прямого ручного керування до взаємодії на основі намірів (intent-based outcome specification), коли людина лише формулює бажаний кінцевий результат, а система самостійно визначає алгоритм його досягнення [2]. Розробка підходів, які дозволять системі виступати інтелектуальним посередником між складним функціоналом та потребами користувача, має високу практичну значущість і передбачає створення адаптивних інтерфейсних рішень під конкретний запит [3].

Однак існуючі підходи до застосування технологій штучного інтелекту в програмних продуктах переважно зводяться до впровадження зовнішніх чат-ботів. Такі інструменти здатні генерувати текстові підказки, проте вони здебільшого не інтегровані у внутрішню логіку самої платформи і не можуть безпосередньо керувати її функціональними модулями чи змінювати стан графічних елементів [4]. Основна науково-практична проблема полягає у нерозробленості архітектурних підходів, які б дозволили використовувати AI-агента як повноцінний інтерфейсний шар. Вибір теми даного дослідження зумовлений саме тим, що у сучасному науковому просторі відсутні комплексні моделі гібридної взаємодії, які б формалізували процес перетворення природномовного наміру на конкретні виклики системи з обов'язковим відображенням результату у звичних елементах графічного інтерфейсу [2, 4].

Аналіз останніх досліджень і публікацій. Концептуальний фундамент людино-машинної взаємодії впродовж десятиліть будувався на принципі прямої маніпуляції: користувач впливає на об'єкти – фізично або віртуально – і негайно бачить результат своїх дій [5]. Ця модель довела свою беззаперечну ефективність для систем із лінійною логікою та обмеженими функціональними можливостями. Проте стрімке масштабування сучасних цифрових платформ до рівня складних багатокомпонентних екосистем та SaaS-рішень виявило фундаментальні обмеження цього підходу. У міру зростання кількості функцій графічний інтерфейс перетворюється на лабіринт багаторівневих меню та інструментальних панелей, що, відповідно до закону Хіка, логарифмічно збільшує час на пошук потрібного інструмента [6, 7]. Дослідження показують, що перевантажений інтерфейс і потреба утримувати в пам'яті структуру системи збільшують стороннє когнітивне навантаження на користувача. Людина змушена витратити значні зусилля не на вирішення цільової задачі, а на навігацію інструментарієм, що призводить до падіння продуктивності та зростання кількості помилок [1].

У відповідь на цю кризу наукова спільнота фіксує необхідність радикального переходу до взаємодії на основі намірів. Якоб Нільсен, відомий фахівець із юзабіліті та співзасновник компанії Nielsen Norman Group, визначає цей зсув як першу якісно нову парадигму інтерфейсу за останні шістьдесят років, яка докорінно змінює розподіл ролей між людиною та машиною [2]. Якщо класичний GUI вимагає імперативного підходу – детального опису того, як саме виконати дію (послідовність кліків, викликів меню та налаштувань), то нова парадигма є декларативною [2]. Вона передбачає специфікацію бажаного результату, де користувач лише формулює кінцеву мету природною мовою, а система бере на себе функцію планування дій та вибору оптимального алгоритму їх реалізації.

Попри швидкий розвиток великих мовних моделей (LLM), їх пряме впровадження в програмні системи супроводжується низкою викликів. Дослідники звертають особливу увагу на проблему «виводу»: традиційні діалогові системи (чат-боти) демонструють високу ефективність у розпізнаванні наміру, проте їхній результат часто залишається обмеженим неструктурованим текстовим форматом. Для професійних середовищ, де важлива щільність даних, точність візуалізації та швидкість прийняття рішень, звичайна текстова відповідь не може бути повноцінною заміною функціональному графічному інтерфейсу [4]. Текст позбавляє користувача інструментів прямого контролю, властивих GUI, таких як можливість миттєвого фільтрування, масштабування графіків або редагування окремих параметрів у формі.

Спробою вирішити це протиріччя стала концепція Generative UI, яка передбачає динамічне відтворення адаптивних графічних віджетів безпосередньо під конкретний контекст запиту користувача. Архітектура такого підходу (рис. 1) спирається на три базові компоненти:

- 1) інтеграцію зовнішніх інструментів (Tool access), що забезпечує моделі доступ до вебпошуку чи генерації медіа для підвищення релевантності результату;

2) керуючі системні інструкції (System instructions), які детермінують цілі, специфікації форматування та протоколи уникнення помилок;

3) модулі постобробки (Post-processing), що здійснюють автоматичну валідацію та коригування згенерованого коду перед його остаточним відтворенням.

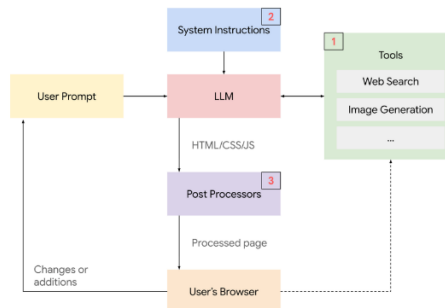


Рис. 1. Архітектура реалізації генеративного інтерфейсу [3]

У такій моделі інтерфейс перестає бути статичною оболонкою і стає плинним середовищем, що формується в реальному часі [3].

Водночас, з позицій інженерії програмного забезпечення, генерація візуальних компонентів без синхронізації з Backend-інфраструктурою та бізнес-логікою не забезпечує цілісного управління платформою [8]. Справжня трансформація вимагає створення системних рішень, де штучний інтелект виступає не як зовнішній консультант, а як вбудований інтерфейсний шар-оркестратор [9]. Цей прошарок має забезпечувати безперервний цикл: від семантичного аналізу природномовного наміру до виконання транзакцій через API системи з подальшою візуалізацією результатів у структурованому графічному вигляді. Саме такий синергетичний підхід дозволяє поєднати інтуїтивність природної мови з надійністю та прозорістю класичних графічних елементів управління.

Наукова новизна даного дослідження полягає у розгляді AI-агента не як допоміжного пошукового інструмента або ізольованого діалогового вікна, а як детермінованого інтерфейсного шару, вбудованого між користувачем і функціональними модулями цифрової платформи. У запропонованій моделі природна мова використовується виключно як механізм введення високорівневого наміру. Далі AI-агент здійснює семантичну декомпозицію цього наміру, формує план дій, ініціює виклики до відповідних API (баз даних, модулів фільтрації, систем порівняння) і модифікує стан графічного інтерфейсу для відображення результату. Принципова відмінність запропонованої архітектури від звичайного чат-бота полягає у характері результату взаємодії. Чат-бот є інформаційно-довідковою системою, яка відповідає на запит текстом, працюючи паралельно з основним функціоналом платформи і не впливаючи на її конфігурацію. Натомість AI-інтерфейсний шар змінює стан самої системи. Він автоматично заповнює форми, активує приховані фільтри, вибудовує таблиці порівняння та здійснює навігацію. AI виступає активним виконавцем усередині платформи, тоді як користувач з оператора інструменту перетворюється на куратора процесу. Детальний розподіл функцій інтелектуального агента в межах платформи представлено у таблиці 1.

Таблиця 1

Ролі AI-агента як інтерфейсного шару

Функція AI-агента	Що робить система	Приклад для платформи
Інтерпретація наміру	Визначає, чого саме прагне досягти користувач	Аналіз запиту «Покажи найбільш релевантні варіанти»
Контекстний аналіз	Враховує поточний стан системи, відкритий розділ та історію сесії	Оцінка того, що користувач вже перебуває на сторінці розширеного пошуку
Перетворення на дії	Визначає необхідний набір функціональних модулів платформи	Оркестрація модулів пошуку, фільтрації та порівняння
Автоматичне налаштування GUI	Заповнює форми або змінює стан графічних елементів	Динамічне встановлення повзунків бюджету, вибір категорій
Виклик API	Передає структурований запит до Backend-інфраструктури	Формування та відправка запиту до бази даних

Продовження таблиці 1

Пояснення результату	Забезпечує прозорість логіки системи для користувача	Формування аргументації: «Цей варіант відповідає бюджету, але розташований далі»
Діалогове уточнення	Дозволяє ітеративно змінювати вибірку без ручного скидання параметрів	Опрацювання контекстного уточнення «Тепер покажи лише дешевші варіанти»
Контроль користувача	Забезпечує можливість ручного скасування або коригування дій AI	Надання доступу до візуальних фільтрів для ручного редагування

Для того щоб системно оцінити еволюційний поступ людино-машинної взаємодії та обґрунтувати архітектурні переваги запропонованого підходу, необхідно провести чітку диференціацію ключових характеристик існуючих інтерфейсних парадигм. Порівняння класичного GUI, чат-ботів та пропонованої моделі інтегрованого AI-інтерфейсу платформи представлено на рисунку 2.

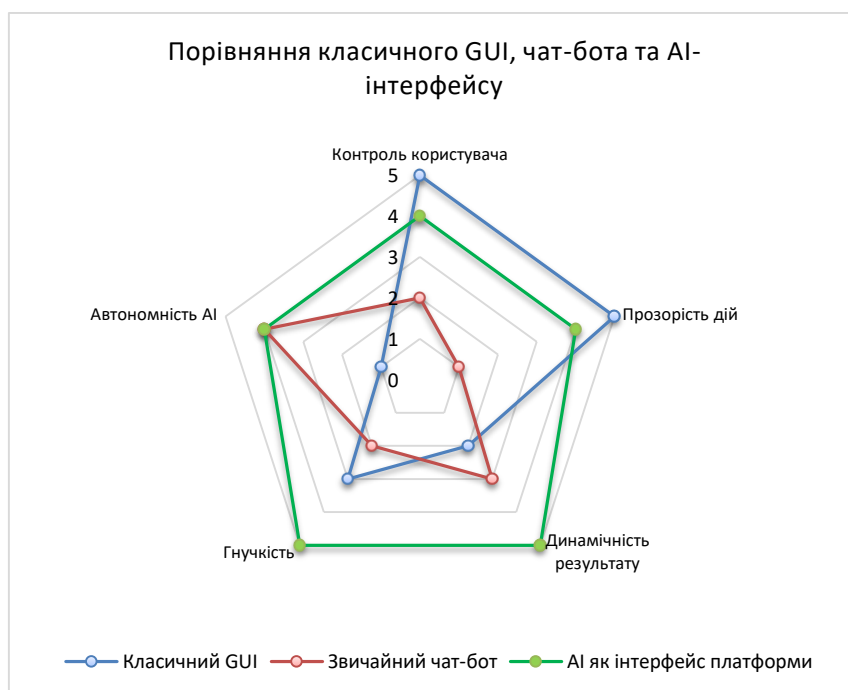


Рис. 2. Параметричне порівняння парадигм взаємодії

Оцінювання кожної моделі здійснено за п'ятибальною шкалою на основі п'яти ключових критеріїв: контроль користувача, прозорість дій, динамічність результату, гнучкість та автономність AI. Форма та загальна площа утворених багатокутників відображають комплексний функціональний потенціал кожного підходу. Графік наочно демонструє, що класичний GUI характеризується різким зміщенням у бік повного ручного контролю та високої прозорості при мінімальній автономності, а профіль базового чат-бота є звуженим через низьку прозорість (ефект «чорної скриньки») та обмежений контроль [10]. Натомість AI-інтерфейс формує найширшу та найбільш збалансовану фігуру, що ілюструє його здатність компенсувати структурні обмеження попередніх парадигм шляхом поєднання високого рівня мультимодальності, динамічності результату та керованої автономності при збереженні прозорості виконання [11, 12].

Мета статті. Зважаючи на відсутність формалізованих підходів до проектування систем, у яких штучний інтелект виступає не окремим автономним застосунком, а активним елементом управління архітектурою взаємодії, метою статті є побудова концептуальної моделі гібридної взаємодії з цифровою платформою, що забезпечує автоматизовану трансформацію природномовних намірів користувача у функціональні дії програмних модулів та динамічну зміну стану графічних елементів через впровадження спеціалізованого AI-інтерфейсного шару.

РЕЗУЛЬТАТИ ДОСЛІДЖЕННЯ

Запропонована архітектурна модель переосмислює традиційну побудову цифрових платформ через введення детермінованого AI-інтерфейсного шару, який виступає посередником між намірами користувача та технічними модулями програмної системи. У межах цього підходу штучний інтелект вбудовується безпосередньо в систему взаємодії як ключовий структурний компонент, що перетворює довільні запити природною мовою на відповідні функціональні можливості платформи в реальному часі. Таке архітектурне рішення дозволяє чітко розмежувати визначення кінцевої цілі та виконання інтерфейсних команд, що є важливим для зниження функціонального перенасичення та надмірного когнітивного навантаження, характерних для сучасних SaaS-екосистем. Завдяки автономному плануванню дій на основі інтерпретації намірів користувача система забезпечує прозорість операційних процесів, перетворюючи статичну ієрархію меню та фільтрів на динамічне середовище, де технологічна складність прихована за інтуїтивною логікою досягнення результату.

Загальна архітектурна композиція системи, у якій AI-агент інтегрований як інтерфейсний шар, детально розкривається через логічну послідовність процесів, представлену на рисунку 3. Концептуальний синтез представленої моделі реалізується через цикл взаємодії, що охоплює спеціалізовані модулі аналізу семантичних намірів, інтелектуального планування та динамічної генерації пояснень. Модуль аналізу намірів виконує гранулярну декомпозицію вхідного сигналу, виділяючи ключові сутності, жорсткі обмеження та пріоритетні вектори пошуку, після чого формалізована структура передається планувальнику для координації послідовних або паралельних викликів до відповідних функціональних модулів через Backend API. Принциповою відмінністю даної моделі є те, що вихідним результатом функціонування AI-шару є не ізольована текстова відповідь, а системна модифікація загального стану графічного інтерфейсу, що супроводжується автоматичним налаштуванням фільтрів, генерацією порівняльних масивів даних та візуалізацією об'єктів у звичних для користувача GUI-компонентах.

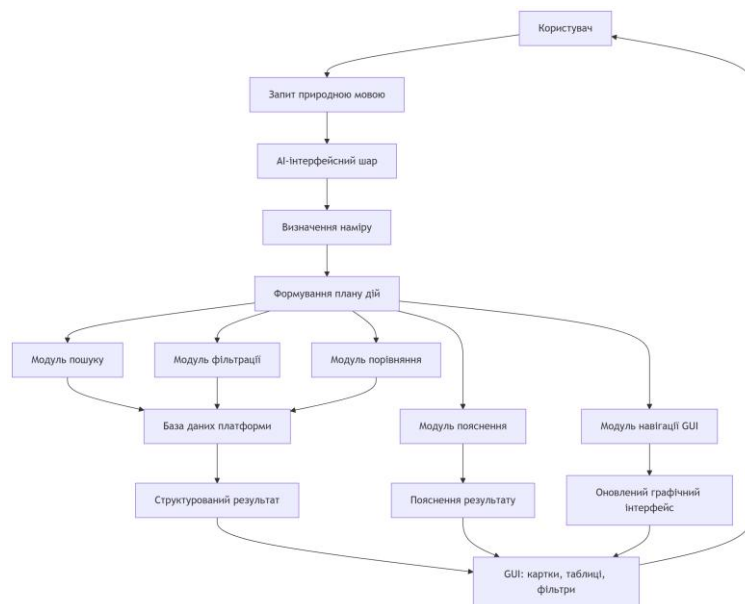


Рис. 3. Логічна діаграма обробки запиту

Детальний аналіз циклічної динаміки взаємодії у запропонованій моделі дозволяє простежити послідовність переходів між намірами користувача та виконавчими механізмами цифрової платформи, що системно відображено на рисунку 4. Процес ініціюється у клієнтському середовищі, де користувач формулює свій намір природною мовою. У цей момент графічний інтерфейс виконує роль не лише приймача вхідного сигналу, а й активного сенсора контексту, передаючи AI-агенту разом із текстом запиту повний набір метаданих про поточний стан системи, відкриті розділи та історію попередніх дій у межах поточної сесії. Отримавши цей агрегований масив даних, інтелектуальний інтерфейсний шар здійснює семантичний аналіз для детермінації цілі та вибору оптимальної стратегії виконання. На відміну від класичних імперативних систем, де користувач змушений самостійно декомпонувати складну задачу на послідовність кліків та викликів меню, у даному рішенні AI-агент бере на себе функцію

планувальника, який автоматично генерує чергу звернень до відповідних функціональних модулів через Backend API платформи. Наступна фаза циклу охоплює етап безпосередньої операційної реалізації наміру та формування структурованої відповіді. Після виконання запитів до бази даних та отримання релевантних відомостей, AI-агент здійснює інтелектуальну обробку результатів, готуючи комплексний пакет інструкцій для графічної оболонки. Це передбачає не просто рендеринг даних, а динамічне налаштування візуальних компонентів: автоматичне встановлення значень у фільтрах і генерацію текстових пояснень, що обґрунтовують логіку вибору системи [3, 4]. Завершальним етапом циклу є виведення результату у звичне для користувача середовище GUI, де реалізується критично важливий механізм верифікації та зворотного зв'язку. Користувач має можливість миттєво оцінити точність виконання запиту та, за необхідності, внести уточнення природною мовою або здійснити пряме ручне редагування параметрів у графічних елементах, що забезпечує безперервність контролю та високу прозорість функціонування застосунку [5, 8].

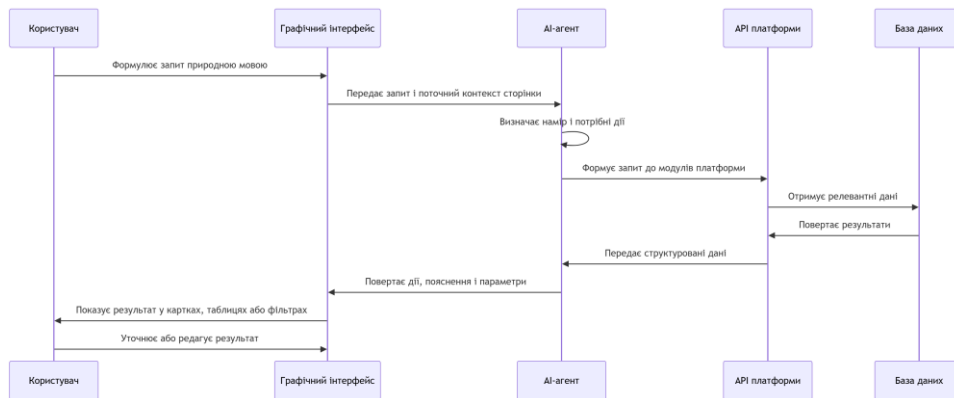


Рис. 4. Діаграма послідовності взаємодії з платформою

Структура системи базується на тривірневій декомпозиції, що охоплює графічний інтерфейс користувача, інтелектуальний інтерфейсний шар та функціональне ядро (Backend API). Графічний інтерфейс у цій моделі виступає як точка входу, що інтегрує поле для введення запитів природною мовою разом із класичними формами та таблицями результатів, забезпечуючи двосторонній зв'язок між людиною та системою. Ключовим архітектурним вузлом є AI-інтерфейсний шар, який виконує роль центрального процесора намірів та здійснює логічне розмежування між суб'єктивним запитом користувача та технічною реалізацією функцій платформи. Внутрішня структура цього шару включає компоненти аналізу наміру та стратегічного планування, які в синергії забезпечують трансформацію неструктурованого тексту у формалізовані послідовності команд для системних модулів (рис. 5).

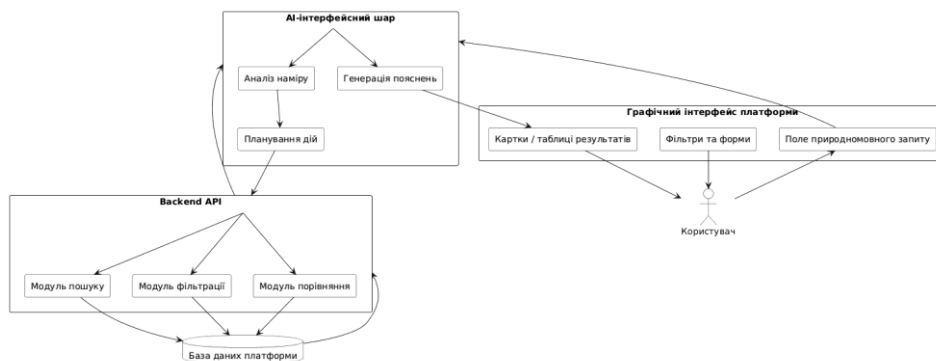


Рис. 5. Компонентна модель платформи

Функціональна стійкість моделі забезпечується через інтеграцію AI-шару із серверним інтерфейсом, який координує роботу спеціалізованих модулів пошуку, фільтрації та порівняння об'єктів. Ці модулі здійснюють пряму взаємодію з базою даних платформи для вилучення релевантної інформації на основі плану, сформованого інтелектуальним агентом. Важливою особливістю компонентної моделі є наявність модуля генерації пояснень у складі AI-шару, який опрацьовує отримані з бази даних результати та формує контекстні обґрунтування для виведення в інтерфейс.



Для систематизації процесу впровадження інтелектуальних агентів розроблено ієрархічну класифікацію з шести рівнів інтеграції, що відображає глибину їхнього залучення у функціональну логіку платформи (табл. 2).

Таблиця 2

Рівні інтеграції AI у цифрову платформу

Рівень	Назва рівня	Опис	Приклад
1	AI як підказка	Система лише пояснює, як виконати дію	«Натисніть кнопку фільтрів»
2	AI як автозаповнення	Система заповнює частину форми або параметрів	Автоматичне встановлення бюджету
3	AI як виконавець дії	Система виконує запит через модулі платформи	Формування вибірки з бази даних
4	AI як пояснювач	Система пояснює логіку результату	Обґрунтування, чому показано певний варіант
5	AI як діалоговий інтерфейс	Користувач уточнює результат у кілька кроків	«Тепер покажи лише дешевші варіанти»
6	AI як інтерфейсний шар	AI поєднує намір користувача з усією логікою платформи	Пошук, фільтри, порівняння, пояснення, навігація

На початкових етапах (рівні 1-2) AI-агент виконує допоміжні реактивні функції підказки або автозаповнення, частково знижуючи операційне тертя в межах традиційної імперативної парадигми. На етапі активного виконання та інтерпретації (рівні 3-4) система набуває здатності самостійно реалізовувати транзакції через Backend API та генерувати контекстні обґрунтування результатів, що суттєво підвищує загальну прозорість взаємодії. Найвищий ступінь архітектурної зрілості досягається на рівнях діалогового інтерфейсу та комплексного інтерфейсного шару (рівні 5-6). Тут забезпечується ітеративне уточнення намірів та повна координація запиту з технічною логікою: від динамічної навігації до адаптивної модифікації графічних компонентів. Саме шостий рівень є базою запропонованого проєкту, оскільки він перетворює багатокомпонентні застосунки на гнучкі адаптивні середовища, мінімізуючи когнітивне навантаження та зберігаючи при цьому точність візуального контролю.

Перевагою запропонованого підходу є те, що користувач не повинен заздалегідь знати структуру платформи. Він може взаємодіяти із системою через природну мову, тоді як AI-агент бере на себе перетворення людського наміру у формалізовані дії. Додатковою перевагою є пояснюваність. Користувач бачить не лише результат, а й логіку його формування: які критерії були враховані, які параметри застосовано та чому система запропонувала саме такий результат.

ВИСНОВКИ ТА ПЕРСПЕКТИВИ ПОДАЛЬШИХ ДОСЛІДЖЕНЬ

У результаті проведеного дослідження було розв'язано важливе науково-практичне завдання щодо подолання кризи когнітивного перевантаження у складних інформаційних системах шляхом переходу від традиційного імперативного керування до взаємодії на основі намірів. Пройдений шлях охоплював системний аналіз обмежень класичних графічних інтерфейсів, зумовлених феноменом функціонального перевантаження, та обґрунтування необхідності впровадження інтелектуального посередника, здатного до семантичної інтерпретації природномовних запитів користувача. На основі вивчення сучасних парадигм людино-машинної взаємодії та архітектурних патернів динамічної генерації інтерфейсів було визначено роль AI-агента як детермінованого інтерфейсного шару, що здійснює оркестрацію функціональних модулів платформи та адаптивну модифікацію GUI-компонентів.

Визначена мета дослідження вважається повністю досягнутою, оскільки в роботі побудовано та формалізовано концептуальну модель гібридної взаємодії з цифровою платформою. Доказом досягнення кінцевого результату є розроблена тривірнева компонентна структура системи, детально описаний алгоритмічний цикл взаємодії та встановлена класифікація рівнів інтеграції штучного інтелекту, де запропонований підхід відповідає найвищому ступеню системної координації. Створена модель забезпечує автоматизовану трансформацію природномовних намірів у конкретні операційні дії через Backend API, що дозволяє ефективно поєднати інтуїтивність мовної взаємодії з точністю та прозорістю візуального контролю. Практична значущість результатів полягає у створенні методологічної бази для проєктування складних інформаційних застосунків нового покоління, які мінімізують операційне тертя та динамічно адаптуються під унікальні потреби кожного завдання в реальному часі.



Проведене дослідження окреслює низку перспективних напрямків для подальших наукових розвідок. У майбутньому буде доцільним здійснити практичну реалізацію запропонованої моделі у вигляді повноцінного програмного прототипу та провести її емпіричне тестування через порівняльні юзабіліті-дослідження з реальними користувачами, що дозволить кількісно підтвердити зниження когнітивного навантаження. Також видається перспективним дослідити питання безпеки та контрольованості детермінованого AI-шару в умовах критично важливих бізнес-процесів, зокрема розробити механізми запобігання некоректній інтерпретації намірів користувача. Окремо буде доцільним перевірити масштабованість архітектури в контексті мультикористувацьких платформ із персоналізованими профілями взаємодії.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Schmidhuber, J., Schögl, S., & Ploder, C. (2021). Cognitive load and productivity implications in human-chatbot interaction. In *2021 IEEE 2nd International Conference on Human-Machine Systems (ICHMS)* (pp. 1-4). IEEE. <https://doi.org/10.1109/ICHMS53169.2021.9582445>
2. Nielsen, J. (2023). *AI: First new UI paradigm in 60 years*. UX Tigers. <https://www.uxtigers.com/post/ai-new-ui-paradigm>
3. Google Research. (2025). *Generative UI: A rich, custom, visual interactive user experience for any prompt*. Google Research Blog. <https://research.google/blog/generative-ui-a-rich-custom-visual-interactive-user-experience-for-any-prompt/>
4. Ordoumpozanis, K., Konstantakis, M., Zoi, S., & Caridakis, G. (2025). Generative AI: A systematic review of related interfaces and interactions. In *Proceedings of the 3rd International Conference of the ACM Greek SIGCHI Chapter* (pp. 39-47). ACM. <https://doi.org/10.1145/3749012.3749052>
5. Shneiderman, B. (2022). Human-centered AI: Ensuring human control while increasing automation. In *Proceedings of the 5th Workshop on Human Factors in Hypertext* (pp. 1-2). ACM. <https://doi.org/10.1145/3538882.3542790>
6. Thompson, D. V., Hamilton, R. W., & Rust, R. T. (2005). Feature fatigue: When product capabilities become too much of a good thing. *Journal of Marketing Research*, 42(4), 431-442. <https://doi.org/10.1509/jmkr.2005.42.4.431>
7. Mishra, S., Guleria, A., & Parikh, V. (2025). Reducing cognitive load in UI design. *International Journal for Research in Applied Science and Engineering Technology*, 13(3), 2744-2446. <https://doi.org/10.22214/ijraset.2025.67917>
8. Dähling, S., Razik, L., & Monti, A. (2021). Enabling scalable and fault-tolerant multi-agent systems by utilizing cloud-native computing. *Autonomous Agents and Multi-Agent Systems*, 35(1), Article 10. <https://doi.org/10.1007/s10458-020-09489-0>
9. Liu, Z., Chen, C., et al. (2024). Make LLM a testing expert: Bringing human-like interaction to mobile GUI testing via functionality-aware decisions. In *Proceedings of the International Conference on Software Engineering (ICSE 2024)*. <https://doi.org/10.1145/3597503.3639180>
10. Khan, S. A., Qadeer, M. A., & Afroz, A. A. (2025). The black box opacity and theoretical understanding of generative AI setbacks. *International Journal of Engineering Research & Technology*, 14(12). <https://doi.org/10.5281/zenodo.18074239>
11. Kryskun, I. M. (2026). Model for evaluating the effectiveness of IT project management using artificial intelligence. *Economics. Management. Business*, 1, 4-10. <https://doi.org/10.31673/2415-8089.2026.012301>
12. Dovzhenko, T. P., & Bondarchuk, A. P. (2025). Analysis of the current stage of artificial intelligence development in telecommunications. *Communication*, 1, 43-48. <https://doi.org/10.31673/2412-9070.2025.019552>

**Andrii Bondarchuk**

Doctor of Technical Sciences, Professor,
Head of the Department of Computer Science,
Borys Grinchenko Kyiv Metropolitan University, Kyiv, Ukraine
ORCID: 0000-0001-5124-5102
a.bondarchuk@kubg.edu.ua

Hlib Kryvolapov

Head of the Scientific society of students, postgraduates, doctoral students and young scientists
of the Faculty of Information Technologies and Mathematics,
Borys Grinchenko Kyiv Metropolitan University, Kyiv, Ukraine
ORCID: 0000-0002-0853-5881
hkryvolapov.fitm25m@kubg.edu.ua

A MODEL OF HYBRID INTERACTION WITH AN AI-AGENT-BASED PLATFORM

Abstract. This article presents a conceptual model of hybrid user interaction with a digital platform, in which an AI agent functions as an integrated interface layer—an orchestrator—between the user and the system’s functional modules. The research is prompted by the crisis of classical graphical interfaces in modern SaaS ecosystems, where an excessive density of control elements creates a critical cognitive load and reduces user productivity. The introduction justifies the relevance of the transition from direct manipulation to intent-based interaction and formulates a research problem regarding the lack of architectural approaches that would formalise the transformation of a natural language query into specific system calls, with the result displayed in familiar graphical interface elements. In the context of an analysis of recent research and publications, the evolution of human-computer interaction paradigms, the limitations of classical user interfaces in the context of Hick’s Law, and the concept of a generative interface are examined, and a fundamental difference between the proposed approach and typical dialogue systems and chatbots is identified. The results section describes a three-tier architecture comprising a graphical user interface as an entry point and active context sensor, an intelligent interface layer with modules for intent analysis, action planning and explanation generation, and a server core with modules for searching, filtering and comparing objects. A logical diagram of query processing, a sequence diagram, and a component-based model of the platform are proposed, which systematically describe the full cycle from the formulation of the user’s intent in natural language to the automatic state transition of graphical components via Backend API calls. Separately, as part of the research, a hierarchical classification has been developed covering six successive levels of artificial intelligence integration into the digital platform, where each subsequent level involves deeper AI involvement in the system’s operation, and the highest level corresponds to a full-fledged interface layer-orchestrator, which takes on the coordination of all interactions between the user and the platform. The conclusions demonstrate that the proposed conceptual model combines the intuitiveness of natural language with the transparency and controllability of a graphical interface, minimises operational friction, and establishes a methodological foundation for the design of next-generation information applications.

Keywords: AI agent; hybrid interaction; graphical user interface; intent-based interaction; generative interface, information system, artificial intelligence.

REFERENCES (TRANSLATED AND TRANSLITERATED)

1. Schmidhuber, J., Schögl, S., & Ploder, C. (2021). Cognitive load and productivity implications in human-chatbot interaction. In *2021 IEEE 2nd International Conference on Human-Machine Systems (ICHMS)* (pp. 1-4). IEEE. <https://doi.org/10.1109/ICHMS53169.2021.9582445>
2. Nielsen, J. (2023). *AI: First new UI paradigm in 60 years*. UX Tigers. <https://www.uxtigers.com/post/ai-new-ui-paradigm>
3. Google Research. (2025). *Generative UI: A rich, custom, visual interactive user experience for any prompt*. Google Research Blog. <https://research.google/blog/generative-ui-a-rich-custom-visual-interactive-user-experience-for-any-prompt/>



4. Ordoumpozanis, K., Konstantakis, M., Zoi, S., & Caridakis, G. (2025). Generative AI: A systematic review of related interfaces and interactions. In *Proceedings of the 3rd International Conference of the ACM Greek SIGCHI Chapter* (pp. 39-47). ACM. <https://doi.org/10.1145/3749012.3749052>
5. Shneiderman, B. (2022). Human-centered AI: Ensuring human control while increasing automation. In *Proceedings of the 5th Workshop on Human Factors in Hypertext* (pp. 1-2). ACM. <https://doi.org/10.1145/3538882.3542790>
6. Thompson, D. V., Hamilton, R. W., & Rust, R. T. (2005). Feature fatigue: When product capabilities become too much of a good thing. *Journal of Marketing Research*, 42(4), 431-442. <https://doi.org/10.1509/jmkr.2005.42.4.431>
7. Mishra, S., Guleria, A., & Parikh, V. (2025). Reducing cognitive load in UI design. *International Journal for Research in Applied Science and Engineering Technology*, 13(3), 2744-2446. <https://doi.org/10.22214/ijraset.2025.67917>
8. Dähling, S., Razik, L., & Monti, A. (2021). Enabling scalable and fault-tolerant multi-agent systems by utilizing cloud-native computing. *Autonomous Agents and Multi-Agent Systems*, 35(1), Article 10. <https://doi.org/10.1007/s10458-020-09489-0>
9. Liu, Z., Chen, C., et al. (2024). Make LLM a testing expert: Bringing human-like interaction to mobile GUI testing via functionality-aware decisions. In *Proceedings of the International Conference on Software Engineering (ICSE 2024)*. <https://doi.org/10.1145/3597503.3639180>
10. Khan, S. A., Qadeer, M. A., & Afroz, A. A. (2025). The black box opacity and theoretical understanding of generative AI setbacks. *International Journal of Engineering Research & Technology*, 14(12). <https://doi.org/10.5281/zenodo.18074239>
11. Kryskun, I. M. (2026). Model for evaluating the effectiveness of IT project management using artificial intelligence. *Economics. Management. Business*, 1, 4-10. <https://doi.org/10.31673/2415-8089.2026.012301>
12. Dovzhenko, T. P., & Bondarchuk, A. P. (2025). Analysis of the current stage of artificial intelligence development in telecommunications. *Communication*, 1, 43-48. <https://doi.org/10.31673/2412-9070.2025.019552>

Отримано редакцією журналу / Received: 26.02.26

Прорецензовано / Revised: 10.03.26

Схвалено до друку / Accepted: 25.06.26



This work is licensed under Creative Commons Attribution-noncommercial-sharealike 4.0 International License.