УДК 004.8

**А.В. Семенюк**

# СТВОРЕННЯ ТА ВИКОРИСТАННЯ LLM-АГЕНТІВ, ІНТЕГРАЦІЯ АГЕНТА З ПОТРІБНИМ СЕРЕДОВИЩЕМ

Вінницький національний технічний університет

**Анотація**

*У доповіді розглянуто концепцію агентів на основі великих мовних моделей (LLM-агентів) та підходи до їх створення і використання. Наведено архітектуру та принципи роботи LLM-агентів, які поєднують потужність великих мовних моделей із зовнішніми інструментами та модулями пам'яті для розв'язання складних завдань. Окремо проаналізовано теоретичні основи побудови таких агентів (планування дій, механізми міркування, пам'ять) та наведено практичні приклади їх застосувань. Розглянуто процес інтеграції LLM-агента з зовнішнім середовищем – підключення до баз знань, сервісів та інших систем – що дозволяє агенту виконувати дії у реальному світі. Висвітлено переваги використання LLM-агентів у різних доменах та окреслено пов'язані виклики і перспективи розвитку.*

**Ключові слова:** LLM-агент, великі мовні моделі, штучний інтелект, планування, пам'ять, інтеграція, інструменти.

**Abstract**

*The report examines the concept of large language model-based agents (LLM agents) and approaches to their creation and usage. It presents the architecture and working principles of LLM agents, which combine the power of large language models with external tools and memory modules to solve complex tasks. The theoretical foundations of building such agents (action planning, reasoning mechanisms, memory) are analyzed, and practical examples of their applications are provided. The process of integrating an LLM agent with its external environment – connecting to knowledge bases, services, and other systems – is considered, enabling the agent to perform actions in the real world. The advantages of using LLM agents in various domains are highlighted, and related challenges and development prospects are outlined.*

**Key words:** LLM agent, large language models, artificial intelligence, planning, memory, integration, tools.

## Вступ

Розвиток сучасних технологій штучного інтелекту призвів до появи LLM-агентів – систем, що використовують великі мовні моделі як «мозок» для вирішення складних завдань у різних предметних сферах. На відміну від традиційних моделей, які лише генерують текст, LLM-агенти здатні самостійно планувати дії та виконувати їх за допомогою зовнішніх інструментів. Це відкриває можливості для автономних інтелектуальних систем, які можуть взаємодіяти з середовищем і виконувати практичні функції – від пошуку інформації до проведення транзакцій [7].

**Метою дослідження є** аналіз підходів до створення та використання LLM-агентів, зокрема їх архітектури та методів інтеграції з зовнішнім середовищем, для підвищення ефективності автоматизації складних завдань.

**Об'єктом дослідження є** процес розробки і функціонування агентів на основі великих мовних моделей. Предмет дослідження – архітектурні компоненти LLM-агентів (модулі планування, пам'яті, інтерфейси інструментів) та технології їх інтеграції із середовищем контексту.

**Наукова новизна** роботи полягає в узагальненні сучасних підходів до побудови LLM-агентів – нового напряму в галузі штучного інтелекту та в систематизації знань про їх взаємодію із зовнішніми ресурсами.

**Практичне значення** отриманих результатів полягає у можливості використання LLM-агентів для створення інтелектуальних програмних систем, здатних автономно виконувати багатоетапні завдання

## LLM-агенти: архітектура, принципи роботи та переваги використання

LLM-агент (від англ. Large Language Model agent) – це програмний агент, у якому в ролі ядра виступає велика мовна модель, здатна розуміти природну мову та генерувати осмислені відповіді. На відміну від звичайного використання мовної моделі (коли модель лише видає завершений результат на запит), LLM-агент поєднує виклики до однієї або кількох мовних моделей з викликами до інших систем та інструментів для виконання складних задач, часто розбиваючи їх на декілька кроків міркувань. Інакше кажучи, LLM-агент – це автономна система, що використовує можливості LLM для планування і виконання дій у певному середовищі шляхом інтеграції з зовнішніми сервісами та даними [1].

Архітектура. Архітектура типового LLM-агента є модульною та складається з кількох ключових компонентів, що взаємодіють. Центральним компонентом є велика мовна модель (LLM) – вона виконує роль ядра агента, обробляючи запити користувача, приймаючи рішення і генеруючи текстові висновки.

Навколо LLM вибудовуються допоміжні модулі: модуль планування забезпечує розбиття складного завдання на підзадачі та визначає послідовність дій для їх виконання; модуль пам'яті відповідає за збереження контексту – як короткострокового (діалогу чи поточного сеансу), так і довгострокового (нагромадженого досвіду агенту); інтерфейс інструментів надає агенту можливості для взаємодії із зовнішніми системами – наприклад, виклик API, виконання коду, доступ до баз даних тощо. Усі ці компоненти працюють у тісній зв'язці: мовна модель аналізує ситуацію і формує план дій, використовуючи пам'ять для врахування попереднього контексту, після чого через інтерфейс інструментів виконує потрібні дії (отримання даних, запуск обчислень тощо) і на основі результатів коригує подальші кроки. Таким чином, LLM-агент функціонує як циклічна система типу «Обдумування – Дія – Спостереження» [5].

Необхідність і переваги використання LLM-агентів. Застосування агентів, керованих великими мовними моделями, обумовлене їх здатністю долати обмеження звичайних AI-систем і вирішувати більш комплексні завдання. По-перше, LLM-агенти переходять від пасивної обробки запитів до активного розв'язання проблем: вони не лише генерують текст-відповідь, а й можуть самостійно здійснювати дії для досягнення поставленої мети. Це якісно новий рівень можливостей ШІ, що дозволяє створювати, напряму, розумних помічників для користувачів або автономних ботів-агентів, спроможних виконувати практичні завдання в цифровому середовищі.

## Основи створення та застосування LLM-агентів: методи планування, пам'ять

Планування. Однією з ключових особливостей LLM-агентів є вбудований механізм планування – здатність розбивати складні цілі на простіші кроки та вибудовувати послідовність дій для їх досягнення. Для реалізації такого планування використовуються спеціальні підходи у налаштуванні мовної моделі. Зокрема, техніка «ланцюжка міркувань» (Chain-of-Thought, CoT) полягає у тому, що модель в явному вигляді спонукається думати крок за кроком, формуючи проміжні логічні висновки перед остаточною відповіддю. Це дозволяє агенту детальніше опрацьовувати складні питання, розділяючи їх на підзадачі, і підвищує прозорість процесу. Розвитком цієї ідеї є підхід «дерево міркувань» (Tree-of-Thought, ToT), що передбачає розгалужене дослідження можливих шляхів рішення: модель генерує кілька альтернативних «гілок» дій та оцінює їх, обираючи найперспективнішу траєкторію вирішення задачі.

Ще один важливий метод – це планування з рефлексією (від англ. plan & reflect), коли агент може переглядати і коригувати свій план дій на основі зворотного зв'язку та отриманих результатів. Для цього в архітектурі агента реалізуються механізми самоперевірки і критики. Яскравим прикладом є метод ReAct (Reason + Act) – підхід, при якому мовна модель інтегрує етапи логічних міркувань і виконання дій, чергуючи їх у циклічному порядку.

Підхід Reflexion пропонує оснащувати агента довготривалою динамічною пам'яттю та правилом оцінки власних дій: після кожної спроби розв'язання задача аналізується – якщо агент припустився

помилки або зайшов у тупик, він може розпочати спочатку, уникаючи попередніх помилкових кроків [4]. Комбінація подібних стратегій планування і самонавчання дозволяє LLM-агентам поступово покращувати свої результати на довгих послідовностях дій, наближаючись до поведінки, подібної людському плануванню з методом проб і помилок. Моделі пам'яті в LLM-агентах.

Для успішного виконання багатоетапних сценаріїв агенту потрібна пам'ять, яка перевищує можливості базової мовної моделі з її обмеженим вікном контексту.

Пам'ять. В LLM-агентах використовуються два основних типи пам'яті [6]. Короткострокова пам'ять зберігає дані про поточний контекст діалогу чи задачі – фактично, це ті ж вбудовані можливості контекстуальної пам'яті LLM (in-context learning), завдяки яким модель пам'ятає попередні репліки користувача або проміжні висновки в рамках одного сеансу. Однак обсяг такої пам'яті обмежений розміром контекстного вікна моделі (як правило, кількома тисячами токенів), і після його перевищення старі дані «забуваються». Довгострокова пам'ять розширює можливості агента за рахунок зовнішніх сховищ – як-от векторні бази даних, файли чи спеціалізовані бази знань. Агент може зберігати туди інформацію про попередні сесії, факти, які він дізнався в ході роботи, результати старих обчислень тощо, а при потребі – відшукати і відновити потрібні відомості через механізм пошуку по пам'яті (наприклад, знаходячи найбільш релевантні векторні подання записів). Таким чином забезпечується ефект майже необмеженої пам'яті: агент з часом може накопичувати знання і досвід, до яких звертається у нових ситуаціях. На практиці часто застосовується гібридна пам'ять, що об'єднує обидва рівні – оперативний контекст і довготривале сховище – для максимальної ефективності.

## Інтеграція LLM-агента з середовищем

Однією з найбільш важливих характеристик LLM-агента є його здатність взаємодіяти із зовнішнім середовищем виконання. Під середовищем у даному контексті розуміється сукупність ресурсів поза межами самої мовної моделі, з якими агент може обмінюватися даними або на які може впливати. Це можуть бути бази знань, веб-сервіси, файлові системи, інтернет-сайти, інші модулі штучного інтелекту або навіть фізичні пристрої. Інтеграція агента з таким середовищем відбувається через підключення інструментів – спеціальних функцій або API, що надають агенту потрібний інтерфейс для дій у зовнішньому світі.

Механізми підключення інструментів. У архітектурі LLM-агента модуль інструментів виконує роль «адаптера» між мовною моделлю та зовнішніми системами. З погляду реалізації, інструментом може бути будь-яка функція або сервіс, виклик якого запрограмований розробником і описаний для агента. Прикладами інструментів є: веб-пошук (агент формує пошуковий запит і отримує результати з інтернету), інтерпретатор коду (агент генерує та виконує фрагмент коду для обчислень чи трансформації даних), доступ до баз даних (виконання SQL-запиту для отримання потрібних даних), сервіси третіх сторін (наприклад, виклик API сервера погоди, платіжної системи, сервісу відправки електронної пошти тощо), системні команди (створення файлу, читання документа, завантаження зображення) та інші. Власне, набори інструментів можуть бути дуже широкими – сучасні агенти здатні комбінувати декілька дій: наприклад, агент може здійснити веб-пошук даних про фінансові ринки, запустити скрипт для аналізу отриманих цифр, зберегти результати в базі даних, а потім сформувати підсумковий звіт – і все це автоматично на основі початкового завдання користувача.

Для того щоб LLM-агент міг коректно використовувати інструменти, він зазвичай забезпечується описом можливостей кожного інструмента (так званою специфікацією API). Під час роботи модель на основі цього опису вирішує, який інструмент і коли застосувати. Сучасні рішення, такі як згадувані плагіни або бібліотеки на зразок LangChain, фактично реалізують шаблон, де LLM генерує спеціальний формат виходу – виклик функції – якщо вважає за потрібне скористатися інструментом. Відповідний виклик API перехоплюється середовищем виконання агента (програмною оболонкою), який власне і здійснює реальну дію у зовнішній системі, після чого підставляє результат назад в агент як нову спостереження [3]. Такий цикл може повторюватися, дозволяючи агенту виконувати серію дій.

LLM-агенти проєктуються під різні середовища, залежно від сфери застосування. У випадку програмних агентів, що працюють у межах інформаційних систем, середовищем є корпоративні бази даних, внутрішні сервіси та знання організації. Наприклад, агент для технічної підтримки інтегрується з CRM-системою компанії: через інструменти він може читати дані користувачів, створювати записи про звернення, оновлювати статуси заявок. В іншому сценарії, агент-аналітик підключений до хмарного середовища з фінансовими даними та бібліотеками аналізу – отримуючи завдання, він може автоматично виконувати складні обчислення на виділеному сервері і повертати результат аналізу. Важливою є також інтеграція з людським інтерфейсом – агенти можуть працювати як чат-боти в месенджерах, голосові асистенти або частина веб-додатку. Це теж частина середовища: користувацький інтерфейс, через який агент отримує вхідні дані і презентує результат. Розробники приділяють увагу створенню зручного інтерактивного середовища, де агент може не лише реагувати на команди, але й запитувати уточнення, виводити проміжні результати, зберігати чернетки і т.п.

Багатоагентні системи. Коли задачі стають дуже складними або різноплановими, доцільно використовувати кілька спеціалізованих агентів, які спільно вирішують проблему. Інтеграція агента з середовищем у такому випадку включає координацію між агентами – створення спільного робочого простору, де агенти обмінюються повідомленнями чи результатами роботи. Існують напрацювання щодо «оркестровки» мультиагентних систем: один із агентів може виступати менеджером, розподіляючи завдання між іншими («майстер-слуга» архітектура), або агенти можуть працювати паралельно, комунікуючи через спільну пам’ять чи канал зв’язку. Наприклад, система CrewAI на базі LangChain пропонує шаблон, де група агентів виконує різні ролі (планувальник, виконавець, перевіряючий тощо) для досягнення спільної мети [3]. Хоча це виходить за рамки одиничного LLM-агента, така інтеграція демонструє гнучкість концепції: LLM-агенти можуть стати будівельними блоками складніших автономних систем, здатних виконувати комплексні проекти.

Безпека та обмеження. Інтеграція агента з реальним середовищем потребує врахування питань безпеки, надійності та етики. Агент, що має доступ до зовнішніх систем, потенційно може здійснювати небажані дії або отримати конфіденційну інформацію. Тому практично реалізовані агенти обмежуються конкретними дозволеними інструментами і діями, а розробники впроваджують запобіжні механізми – валідацію вихідних команд, фільтрацію контенту, аудит і логування дій агента [2]. Наприклад, перш ніж виконати згенерований агентом код, система може перевірити його на відповідність політикам безпеки. Такі заходи покликані запобігти ситуаціям, коли помилка або зловмисна інструкція призведе до шкідливих наслідків. Попри ці виклики, тренд на інтеграцію LLM-агентів набирає обертів: з розвитком технологій з’являються все кращі засоби для безпечного підключення агентів до середовищ, включно з підтримкою авторизації, шифрування, «пісочниць» для виконання коду тощо.

## Висновки

LLM-агенти представляють собою новий етап розвитку штучного інтелекту – перехід від просто генерування тексту до автономного вирішення завдань шляхом активної взаємодії з зовнішнім середовищем. Об’єднавши великі мовні моделі з інструментами, пам’яттю та модулями планування, такі агенти здатні виконувати складні багатокрокові операції, які раніше вимагали участі людини. Наведений аналіз показав, що LLM-агенти мають низку важливих переваг: вони можуть залучати актуальні дані та зовнішні обчислювальні ресурси для підвищення точності та релевантності відповідей [2], зберігати контекст взаємодії у довготривалій пам’яті, адаптувати свої дії до змін середовища і потреб користувача, а також вчитися на власному досвіді. Вже сьогодні продемонстровано ефективність використання таких агентів у різних сферах – від клієнтських сервісів і автоматизації бізнес-процесів до наукового аналізу та програмної інженерії [2]

Водночас, впровадження LLM-агентів ставить і нові виклики. Забезпечення надійності й контрольованості поведінки агента є критично важливим завданням: необхідно запобігати помилковим або небезпечним діям, враховувати етичні аспекти прийняття рішень ШІ. Технічні складності

багатокомпонентної архітектури (налагодження планувальника, управління пам’яттю, інтеграція великої кількості інструментів) вимагають ретельного підходу до розробки і тестування таких систем.

Проте активні дослідження у цьому напрямі тривають, і вже з’являються рішення для підвищення прозорості роботи агентів, засоби моніторингу та «відладки» їхніх дій, а також більш досконалі алгоритми міркування. Очікується, що LLM-агенти відіграватимуть все більшу роль у програмних системах нового покоління. Подальший прогрес у галузі йде шляхом покращення моделей (наприклад, поява мультимодальних агентів, які працюють не лише з текстом, а й з зображеннями, звуком тощо), розвитку алгоритмів планування та навчання агентів у симуляціях реального світу. В перспективі LLM-агенти можуть стати основою для створення справді інтелектуальних автономних систем, що будуть здатні підлаштовуватися під середовище, самовдосконалюватися і безпечно співіснувати з людьми, виконуючи роль універсальних помічників у різних сферах життя.

## СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ


1. Understanding LLM agent architectures (DataStax, 2025) – Електронний ресурс. Режим доступу: https://www.datastax.com/guides/understanding-llm-agent-architectures
2. Rizqi Mulki. «LLM Agents and Tool Use: Building AI Systems That Can Act» – Medium, 2025. Електронний ресурс. Режим доступу: https://medium.com/@rizqimulkisrc/llm-agents-and-tool-use-building-ai-systems-that-can-act-0bfabf6d6b88
3. Дем’єн Березенко. «Еволюція ботів з ШІ: використовуємо можливості агентів, моделей RAG і LLM» – DOU, 2024. Електронний ресурс. Режим доступу: https://dou.ua/forums/topic/49083/
4. Lilian Weng. «LLM Powered Autonomous Agents» – Lil’Log Blog, 2023. Електронний ресурс. Режим доступу: https://lilianweng.github.io/posts/2023-06-23-agent/
5. LLM agents: The ultimate guide 2025 – SuperAnnotate, 2025. Електронний ресурс. Режим доступу: https://www.superannotate.com/blog/llm-agents
6. LLM Agents – Prompt Engineering Guide (Dair AI) – Електронний ресурс. Режим доступу: https://www.promptingguide.ai/research/llm-agents
7. Understanding AI & LLM Agents: Architecture, Security, & Deployment – Skyflow Blog, 2023. Електронний ресурс. Режим доступу: https://www.skyflow.com/post/understanding-llm-agents



***Семенюк Андрій Васильович*** - Інститут докторантури та аспірантури, Вінницький національний технічний університет, м. Вінниця, e-mail: andrew.semeniuk.university@gmail.com

***Semeniuk Andrew V.*** - Institute of doctoral and postgraduate studies, Vinnytsia National Technical University, Vinnytsia, e-mail: andrew.semeniuk.university@gmail.com