

УДК 81'33:004.891.2

DOI <https://doi.org/10.24919/2308-4863/103-24>

Володимир ПАСІЧНИК,

orcid.org/0000-0002-5231-6395

доктор технічних наук, професор,
професор кафедри інформаційних систем та мереж
Національного університету «Львівська політехніка»
(Львів, Україна) vrasichnyk@gmail.com

Максим ЯРОМИЧ,

orcid.org/0009-0005-3299-6695

аспірант кафедри прикладної лінгвістики
Національного університету «Львівська політехніка»
(Львів, Україна) yatax0312@gmail.com

Артем АСАУЛЮК,

orcid.org/0009-0004-0720-7212

аспірант кафедри прикладної лінгвістики
Державного вищого навчального закладу «Ужгородський національний університет»
(Ужгород, Україна) artem.asauliuk@uzhnu.edu.ua

МУЛЬТИАГЕНТНИЙ ПІДХІД ДО ОНТОЛОГІЧНО-КЕРОВАНОГО ГЕНЕРУВАННЯ НАУКОВИХ ТЕКСТІВ: ТЕОРЕТИЧНІ ЗАСАДИ

Статтю присвячено теоретичному обґрунтуванню мультиагентного підходу до онтологічно-керованого генерування наукових текстів засобами великих мовних моделей. Актуальність теми зумовлена тим, що великі мовні моделі вже стали реальним інструментом підготовки й опрацювання наукових текстів, тоді як методики їх контрольованого використання лише формуються. Вимоги до наукового тексту значно ширші за граматичну правильність: важать точність термінології, логічна несуперечливість, коректність часових зв'язків та можливість встановити джерельну основу тверджень. У режимі одноетапного генерування ці вимоги контролювати складно, оскільки модель відразу формує фінальний текст, не залишаючи проміжних станів, придатних для перевірки. Показано, що продуктивнішим є розділення процесу на джерельний, термінологічний, концептуальний, темпоральний, композиційний, генеративний і контрольний рівні, кожен із яких залишає власний артефакт. Схарактеризовано становлення мультиагентної парадигми в межах досліджень розподіленого штучного інтелекту та причини її актуалізації після появи великих мовних моделей. Простежено перехід від класичного програмного агента, поведінку якого задавали алгоритмами й логічними моделями, до агента, роль якого визначається природномовною системною інструкцією, доступом до інструментів і механізмами пам'яті. Розглянуто принципи взаємодії та координації агентів і обґрунтовано, чому для роботи з науковим текстом важливим є структурований, а не вільний обмін даними між ролями. Виконано порівняння одноетапного та мультиагентного підходів за критеріями декомпозиції, простежуваності, спеціалізації й вбудованого контролю якості. Проаналізовано сучасні LLM-мультиагентні системи як приклади різних архітектурних рішень. Окреслено місце мультиагентності як моделі організації філологічної роботи з науковим текстом, за якої мовне генерування відокремлюється від термінологічних, концептуальних і композиційних рішень, що йому передують.

Ключові слова: великі мовні моделі, мультиагентні системи, онтологічно-кероване генерування, науковий текст, простежуваність, декомпозиція, агентна роль.

Volodymyr PASICHNYK,

orcid.org/0000-0002-5231-6395

*Doctor of Technical Sciences, Professor,
Professor at the Department of Information Systems and Networks
Lviv Polytechnic National University
(Lviv, Ukraine) vpasichnyk@gmail.com*

Maksym YAROMYCH,

orcid.org/0009-0005-3299-6695

*PhD student at the Department of Applied Linguistics
Lviv Polytechnic National University
(Lviv, Ukraine) yamax0312@gmail.com*

Artem ASAULIUK,

orcid.org/0009-0004-0720-7212

*PhD student at the Department of Applied Linguistics
Uzhhorod National University
(Uzhhorod, Ukraine) artem.asauliuk@uzhnu.edu.ua*

MULTI-AGENT APPROACH TO ONTOLOGY-GUIDED GENERATION OF SCIENTIFIC TEXTS: THEORETICAL FOUNDATIONS

The article provides a theoretical substantiation of the multi-agent approach to ontology-guided generation of scientific texts using large language models. The relevance of the topic stems from the fact that large language models have already become a practical instrument for preparing and processing scientific texts, whereas methods for their controlled use are only now taking shape. The requirements for a scientific text are considerably broader than grammatical correctness: what matters is terminological precision, logical consistency, the correctness of temporal relations, and the possibility of establishing the source basis of statements. In one-step generation these requirements are difficult to control, since the model immediately produces the final text, leaving no intermediate states available for verification. It is shown that a more productive solution is to separate the process into source-related, terminological, conceptual, temporal, compositional, generative, and control levels, each of which leaves its own artefact. The article characterises the emergence of the multi-agent paradigm within distributed artificial intelligence research and the reasons for its renewed relevance after the advent of large language models. It traces the transition from the classical software agent, whose behaviour was defined by algorithms and logical models, to an agent whose role is determined by a natural-language system instruction, access to tools, and memory mechanisms. The principles of agent interaction and coordination are examined, substantiating why structured rather than free-form data exchange between roles matters for work with scientific texts. A comparison of the one-step and multi-agent approaches is carried out against the criteria of decomposition, traceability, specialisation, and built-in quality control. Contemporary LLM-based multi-agent systems are analysed as examples of different architectural solutions. The place of multi-agency as a model for organising philological work with scientific texts is outlined, whereby linguistic generation is separated from the terminological, conceptual, and compositional decisions that precede it.

Key words: large language models, multi-agent systems, ontology-guided generation, scientific text, traceability, decomposition, agent role.

Постановка проблеми. Поява великих мовних моделей істотно змінила практики роботи з текстом у науковій та освітній сферах. Ще кілька років тому такі системи сприймалися переважно як експериментальний засіб продовження або перефразування тексту. Тепер вони реферують джерела, виділяють терміни, класифікують фрагменти, пропонують структуру викладу й формують цілісні тексти в академічному стилі.

Для філологічних досліджень ця зміна особливо відчутна, адже значна частина дослідницьких процедур тут безпосередньо пов'язана з природною мовою: добором і тлумаченням термінів, аналізом семантичних зв'язків, зіставленням тек-

стів, формуванням дефініцій та узагальненням джерел. Проте швидке поширення LLM випередило формування усталених методологічних норм їх застосування.

Тому питання вже не зводиться до того, чи може модель створити граматично й стилістично прийнятний текст. Важливіше інше: як організувати генерування так, щоб результат залишався контрольованим, перевірюваним і придатним до наукового використання (Vaswani et al., 2017; Пасічник, Яромиш, 2025а, 2026).

Вимоги до наукового тексту значно ширші за граматичну правильність і зовнішню стилістичну гладкість. Важать точність термінології, відпо-

відність понять предметній галузі, логічна несуперечливість, коректність причинно-наслідкових і часових зв'язків, обґрунтованість узагальнень, структурна цілісність і можливість встановити джерельну основу тверджень. Для філолога окремою проблемою є сталість терміносистеми: той самий концепт не повинен без підстав отримувати різні назви в різних частинах тексту, так само як близькі за формою одиниці не можна автоматично ототожнювати. У режимі прямого генерування ці вимоги контролювати складно. Модель отримує великий запит, відразу формує фінальний текст, а користувач бачить переважно результат, але не послідовність рішень, що до нього привели (Celikyilmaz et al., 2020; van der Lee et al., 2021).

Через це стилістично переконливий фрагмент може приховувати неточність. Якщо її помічено вже наприкінці, не завжди зрозуміло, де вона виникла: під час добору джерел, виділення термінів, встановлення концептуального зв'язку чи безпосередньо під час формулювання речення. Вимога до промпта «не помилятися» тут мало що змінює. Методологічно продуктивніше організувати процес так, щоб помилку можна було локалізувати, зіставити з попереднім етапом і виправити до того, як вона стане частиною остаточного тексту (Gao et al., 2023; Ji et al., 2023; Lewis et al., 2020).

Одним зі способів такого впорядкування є онтологічно-кероване генерування. Його принцип полягає в тому, що текст створюється не безпосередньо з початкового запиту, а на основі попередньо сформованої структури знання: термінологічної бази, набору концептів, типізованих відношень між ними, темпоральних характеристик та композиційного плану. У цьому випадку мовна модель використовується не як єдиний універсальний генератор, а як компонент ширшої інформаційної технології. Формальна або напівформальна структура знання задає межі й опорні точки для генерування, тоді як LLM забезпечує мовну реалізацію, узагальнення й зв'язність викладу. Такий підхід особливо перспективний для філологічних і міждисциплінарних досліджень, де значення мовної одиниці часто залежить від контексту, а коректність терміна не може бути відокремлена від місця відповідного поняття в системі знань (Dessi et al., 2021; Gruber, 1993; Neuhaus, 2023; Noy, McGuinness, 2001; Пасічник, Яромич, 2025с).

Проте сама наявність онтології ще не розв'язує організаційної проблеми.

Побудова наукового тексту охоплює кілька різнотипних операцій, кожна з яких висуває власні вимоги до мовної моделі. Пошук і структурування

джерел потребують точності та стабільності формату; термінологічне опрацювання – чутливості до спеціалізованої лексики й уміння розрізняти близькі поняття; побудова онтології – здатності встановлювати й типізувати концептуальні зв'язки; темпоральне моделювання – обережної роботи з датами, періодами та еволюцією понять; планування – композиційної організації матеріалу; генерування – володіння академічним стилем; валідація – критичності та здатності зіставляти готовий текст із початковими артефактами. Виконання всіх цих функцій однією моделлю в межах одного запиту технічно можливе, однак методологічно воно приховує межі між окремими процедурами й ускладнює контроль їх якості (Пасічник, Яромич, 2025b, 2025d).

Актуальність дослідження, таким чином, визначається поєднанням трьох взаємопов'язаних обставин. По-перше, LLM уже стали реальним інструментом підготовки й опрацювання наукових текстів, тоді як методики їх контрольованого використання ще формуються. По-друге, для філологічного наукового тексту критично важливими є термінологічна, концептуальна й темпоральна узгодженість, які не забезпечуються самим фактом мовної зв'язності відповіді. По-третє, сучасна агентна парадигма створює можливість розкласти процес генерування на спеціалізовані, видимі й перевірювані етапи, поєднавши гнучкість великих мовних моделей зі структурованістю онтологічного подання знань.

Аналіз досліджень. Мультиагентний підхід сформувався не як наслідок появи великих мовних моделей, а значно раніше – у межах досліджень розподіленого штучного інтелекту, де центральною стала проблема спільної роботи кількох автономних обчислювальних сутностей. Причина переходу доволі проста: багато складних задач за своєю природою не є монолітними. Вони складаються з відносно самостійних підзадач, потребують різних типів знань, виконуються в мінливому середовищі й часто передбачають одночасну дію кількох виконавців.

У такому трактуванні агентність є насамперед способом організації складної системи. Агент не обов'язково має бути «інтелектуальним» у людському розумінні й не потребує повної картини всього процесу; йому достатньо мати власну мету, доступні дії та правила взаємодії з оточенням. У класичних роботах цю організацію описують через автономність, реактивність, проактивність і соціальність: автономність означає, що компонент виконує свою функцію без постійного зовнішнього керування; реактивність пов'язана зі здатністю змінювати поведінку у відповідь на нові

дані; проактивність – ініціювати дії заради мети; соціальність – обмінюватися повідомленнями та координуватися з іншими агентами (Jennings et al., 1998; Wooldridge, 2009). Ці властивості не втратили значення після появи LLM. Змінився передусім спосіб, у який вони реалізуються.

На цій основі й виникли LLM-орієнтовані мультиагентні системи. Кілька екземплярів однієї моделі або різні моделі можна наділити відмінними ролями, після чого організувати між ними обмін повідомленнями й проміжними артефактами. Тоді центральними стають уже не лише характеристики LLM, а профілювання агентів, розподіл задач, пам'ять, керування контекстом, способи комунікації та механізми узгодження розбіжностей (Guo et al., 2024; Han et al., 2024).

Окремі складники цієї проблематики опрацьовано і в українському науковому просторі – у дослідженнях, присвячених застосуванню великих мовних моделей у філологічних задачах, лексикографічному аналізу, концептуалізації предметної області та автоматизованому реферуванню (Пасічник, Яромич, 2025a; 2025b; 2025c; 2025d; 2026). Водночас питання про те, як організувати взаємодію кількох спеціалізованих агентів так, щоб кожен етап роботи з науковим текстом залишався контрольованим, досі не має усталеного теоретичного розв'язання.

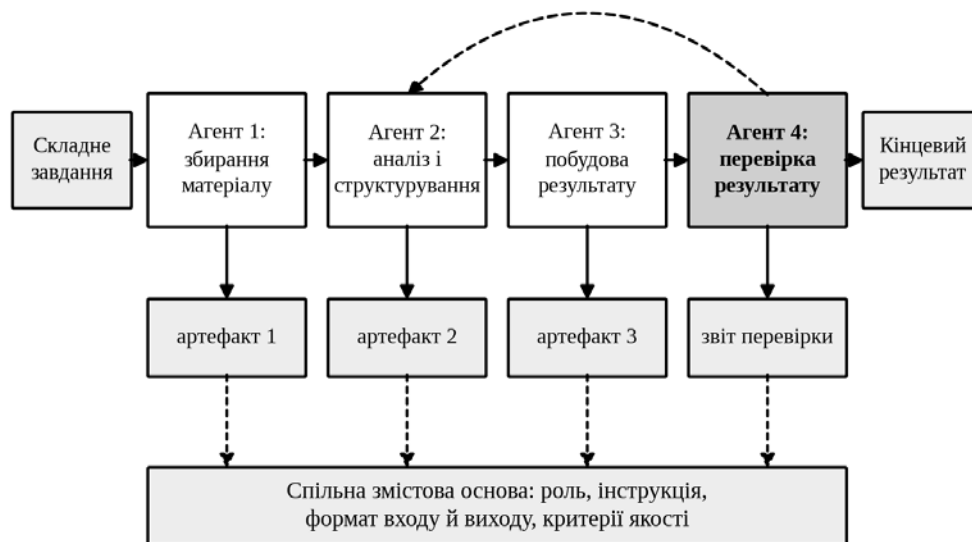
Мета статті – теоретичне обґрунтування мультиагентного підходу до онтологічно-керованого

генерування наукових текстів. Для досягнення цієї мети передбачається простежити становлення мультиагентної парадигми та причини її актуалізації в умовах розвитку великих мовних моделей; схарактеризувати перехід від класичного програмного агента до агента на основі LLM; описати принципи взаємодії та координації агентів і визначити, які з них придатні для роботи з науковим текстом; порівняти одноетапне генерування з мультиагентною організацією за практично значущими критеріями; проаналізувати наявні LLM-мультиагентні системи як приклади різних архітектурних рішень; окреслити місце мультиагентності як моделі організації філологічної роботи з науковим текстом. Експериментальну реалізацію описаної моделі в середовищі Langflow подано в окремій публікації, що продовжує цю статтю.

Виклад основного матеріалу. Розгляд доцільно почати із загальної схеми мультиагентного конвеєра, яка узагальнює описану логіку декомпозиції та слугує орієнтиром для подальшого викладу (рисунок 1).

Від класичного програмного агента до агента на основі великої мовної моделі

До появи сучасних LLM поведінку програмного агента переважно задавали алгоритмами, правилами, логічними моделями або спеціалізованими процедурами. Такий агент міг бути автономним, однак його компетенція зазвичай зали-



пунктиром позначено можливе повернення завдання на попередній етап за результатами перевірки

Рис. 1. Загальна схема мультиагентного конвеєра: спеціалізовані ролі, проміжні артефакти та можливе повернення на попередній етап

шалася вузькою: він умів виконувати лише ті дії, для яких було передбачено формальні правила або програмні механізми. У середовищах, де стани, дії й цілі можна чітко описати, це працювало добре. Природна мова створювала суттєву проблему. Щоб агент міг опрацьовувати довільні інструкції, інтерпретувати тексти, узагальнювати інформацію та формувати новий текст, необхідно було окремо програмувати велику кількість мовних процедур. Саме тут LLM принципово змінили архітектурні можливості агентних систем: мовна модель стала універсальним механізмом інтерпретації природномовної інструкції та породження відповіді, а роль агента почала задаватися не лише кодом, а й системним описом його функції (Park et al., 2023; Shinn et al., 2023; Yao et al., 2023).

Сучасний LLM-агент можна спрощено описати як поєднання мовної моделі, інструкції, контексту, пам'яті та доступних інструментів. Системна інструкція визначає роль і межі компетенції; контекст містить інформацію, з якою агент працює в конкретному запуску; пам'ять дає можливість зберігати релевантні результати попередніх кроків; інструменти забезпечують вихід за межі суто текстового генерування – наприклад, звернення до пошуку, бази даних, API, обчислювального модуля чи локального сховища. Значущим кроком у цьому напрямі став підхід ReAct, який поєднав формування міркувань із виконанням дій у зовнішньому середовищі: модель не лише продукує текст, а може чергувати інтерпретацію поточного стану з викликами інструментів і використовувати отримані результати для наступних кроків (Yao et al., 2023). Для подальшого розвитку агентних систем це було принципово, оскільки мовна модель перестала бути замкненим генератором відповіді й стала компонентом циклу «сприйняття – рішення – дія – новий стан».

Водночас варто зберігати термінологічну точність. Одноразовий запит до моделі ще не робить її агентом; один агент із кількома інструментами теж не утворює мультиагентної системи. Про мультиагентність ідеться тоді, коли функції справді розподілені між кількома окремими учасниками, для яких визначено власні входи, інструкції, виходи та місце в загальному потоці даних.

Принципи взаємодії та координації агентів

Спосіб поєднання агентів істотно впливає на властивості всієї системи. Найпростішою формою є послідовний конвеєр: результат одного агента стає входом наступного. Така структура особливо придатна для задач, у яких сам предмет опрацювання проходить кілька логічно впорядкованих перетворень – спочатку формується корпус, потім

із нього виділяють терміни, на основі термінів створюють концептуальну модель, після чого ця модель використовується для планування й генерування тексту. Перевагою послідовної схеми є прозорість: легко встановити, який агент створив певний проміжний артефакт і що саме він отримав на вході. Вразливість теж відома – помилка раннього етапу може передаватися далі. Тому для наукових задач важливо не лише організувати ланцюжок, а й передбачити контрольні точки, в яких проміжний результат може бути перевірений або повернений на уточнення.

Послідовний конвеєр – лише один із варіантів координації. У складніших системах агенти можуть працювати паралельно, підпорядковуватися окремому координатору або вступати в дискусію. Паралельна схема корисна, коли потрібно отримати кілька незалежних інтерпретацій і потім їх зіставити. Ієрархічна передає одному агенту функцію диспетчера: він декомпонує завдання, розподіляє підзадачі й вирішує, чи потрібен ще один цикл. Дискусійна модель, навпаки, робить ставку на взаємну критику перед формуванням спільного результату. У кожному випадку якість системи значною мірою визначається не кількістю агентів, а правилами їхньої комунікації. Надлишкова взаємодія теж має ціну: контекст розростається, виконання дорожчає, а помилки можуть циркулювати між компонентами (Guo et al., 2024; Han et al., 2024).

Для філологічних задач особливо корисним є структурований обмін. Вільний текст легко стирає межу між фактом, інтерпретацією та службовим коментарем; натомість список термінів із дефініціями, таблиця концептів і зв'язків, JSON-структура чи ієрархічний план значно краще піддаються перевірці. Саме тому агентна організація природно поєднується з онтологічним підходом: LLM забезпечує гнучкість інтерпретації, а структура виходу обмежує довільність.

Переваги мультиагентної організації для генерування наукового тексту

Порівняння одноетапного генерування та мультиагентної організації не слід зводити до питання, який підхід швидше створює текст. Одноразовий запит до сильної LLM майже завжди буде простішим і дешевшим за багатоетапний конвеєр. Перевага агентної схеми виникає в іншій площині – в організації дослідницького процесу. Замість одного великого перетворення «запит → текст» формується послідовність менших перетворень, кожне з яких має окремий зміст і може бути оцінене незалежно.

Для наукового письма це принципово: кінцевий текст повинен не лише звучати переконливо, а

й зберігати зв'язок із джерельною та концептуальною основою. Розгорнуте зіставлення обох підходів подано в таблиці 1.

Найочевидніша перевага мультиагентної організації – декомпозиція. Агенту не потрібно одночасно шукати джерела, визначати терміни, будувати онтологію й писати академічний текст. Чим точніше окреслена роль, тим простіше сформулювати інструкцію і, що не менш важливо, перевірити її виконання. Для філолога така деталізація робить методику описуваною через звичні дослідницькі операції: можна окремо зафіксувати термінологічне вилучення, концептуалізацію, композиційне планування й генерування, а не обмежуватися нечітким формулюванням «використано ІІІ для написання тексту».

З декомпозицією безпосередньо пов'язана простежуваність. Якщо проміжні артефакти зберігаються, сумнівний термін можна повернути до термінологічної бази, суперечливий зв'язок – до онтології, хронологічне узагальнення – до темпоральної моделі, а структурний дисбаланс – до плану. Математично строгого доказу походження кожного речення це не дає. Зате створює практичний ланцюг аудиту, якого бракує монолітному генеруванню.

Ще одна перевага – спеціалізація. Мовні моделі помітно відрізняються за роботою з довгим контекстом, структурованим виводом, інструментами, багатомовністю, академічним стилем і критичною перевіркою. Тому використання однієї LLM для всіх ролей зручне технічно, але не завжди виправдане методично. Гетерогенна конфігурація дозволяє добирати модель під конкретну операцію, а не під абстрактне уявлення про «найкращу модель взагалі».

Контроль також можна перенести всередину самої архітектури. У звичайному сценарії текст спочатку генерується, а вже потім користувач намагається знайти в ньому помилки. У мультиагентній схемі перевірка може бути окремою функцією: один компонент оцінює результат іншого, зіставляє його з онтологією або повертає завдання на повторне виконання.

Автоматично надійною систему це не робить. Додаткові агенти створюють і додаткові точки відмови; помилка раннього етапу може перейти далі як нібито достовірний вхід. Зростають також час, вартість і складність відтворення. Тому мультиагентність має сенс не там, де потрібно просто швидко отримати гладкий текст, а там, де цінність мають проміжні результати, контроль і можливість локалізувати збій.

Приклади сучасних LLM-мультиагентних систем

Показовим прикладом загального фреймворку є AutoGen, у якому програми будуються як взаємодія кількох налаштовуваних агентів, здатних вести між собою діалог, використовувати LLM, зовнішні інструменти та за потреби залучати людину. Загальну організацію такої системи подано на рисунку 2. Автори демонстрували застосування такого підходу до задач програмування, математики, запитань-відповідей, операційних досліджень та інтерактивного прийняття рішень (Wu et al., 2023). Важливою тут є не конкретна предметна область, а архітектурна ідея: поведінка складного застосунку визначається схемою взаємодії спеціалізованих агентів, а не одним великим промптом. Це робить AutoGen добрим прикладом переходу від «моделі як сервісу відповіді» до «моделі як учасника процесу».

Таблиця 1

Порівняння одноетапного та мультиагентного підходів до роботи з LLM

Критерій	Один промпт / одна модель	Мультиагентна організація
Декомпозиція задачі	Різні операції приховані всередині одного запиту	Кожна функція виділена в окремий етап або роль
Проміжні результати	Здебільшого недоступні як самостійні артефакти	Формуються та можуть зберігатися після кожного етапу
Локалізація помилки	Помилку часто видно лише у фінальному тексті	Можна встановити етап, на якому виникла неточність
Спеціалізація моделей	Одна модель виконує всі функції	Для різних ролей можна застосовувати різні моделі та інструменти
Контроль термінології	Підтримується переважно через загальну інструкцію	Термінологічна база може бути окремим входом для наступних агентів
Простежуваність	Походження окремого твердження часто непрозоре	Можна зіставляти фінальний текст із корпусом, онтологією та планом
Вартість і складність	Нижчі	Вищі через кілька запусків і необхідність оркестрації

Іншу логіку демонструє MetaGPT. У цій системі мультиагентну співпрацю організовано за аналогією з робочим процесом команди розроблення програмного забезпечення: окремим агентам надаються ролі, що відповідають різним професійним функціям, а результати одного етапу стають формалізованими входами для іншого. Автори описують цей принцип як перенесення стандартних операційних процедур людської команди в LLM-мультиагентну систему (Hong et al., 2024). Для дослідження наукового тексту аналогія особливо продуктивна, оскільки демонструє, що рольова модель може відображати не технічні модулі, а реальні типи інтелектуальної праці. Так само як у MetaGPT окремі агенти відповідають за аналіз вимог, проєктування та реалізацію, у науко-

вому конвеєрі ролі можуть відповідати за корпус, термінологію, онтологію, планування, генерування й валідацію. Відповідність між рольовою схемою MetaGPT і послідовністю операцій наукового конвеєра показано на рисунку 3.

Окремий напрям становлять системи, у яких кілька агентів генерують альтернативні міркування, дискутують або взаємно критикують відповіді. Їхня логіка виходить із припущення, що одна модель може пропустити власну помилку, тоді як незалежна оцінка іншого агента здатна виявити суперечність або запропонувати альтернативне тлумачення. Огляди Guo et al. (2024) та Han et al. (2024) відносять колективне розв’язання складних задач, рольову взаємодію й ітеративні дискусії до важливих напрямів сучасних LLM-



Рис. 2. Організація взаємодії агентів у фреймворку AutoGen (складено за: Wu et al., 2023)

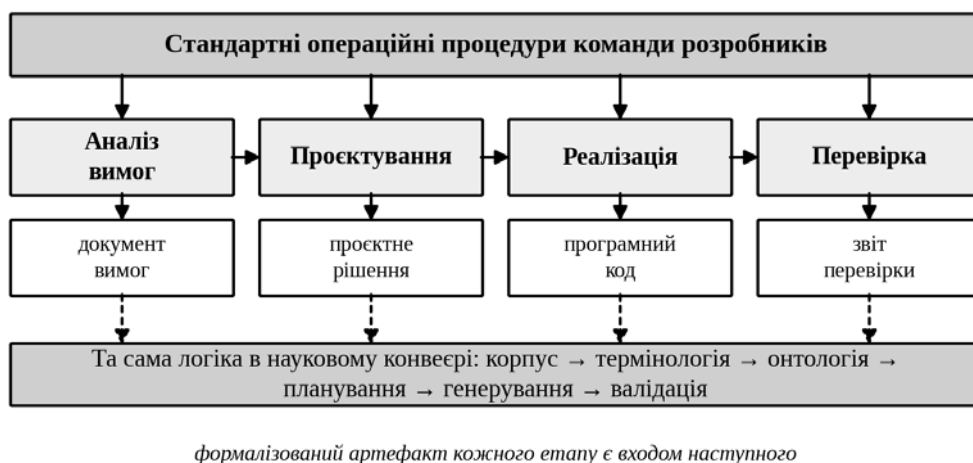


Рис. 3. Рольова організація та формалізовані артефакти в системі MetaGPT і паралель із науковим конвеєром (складено за: Hong et al., 2024)

мультиагентних досліджень. Для філології подібний механізм потенційно цікавий не лише як «голосування моделей», а як спосіб організувати альтернативні інтерпретації тексту: один агент може пропонувати класифікацію або тлумачення, інший – шукати контрприклад, третій – перевіряти термінологічну відповідність. Проте в експериментальній реалізації, поданій в окремій публікації, використано простішу послідовну схему, оскільки її легше контролювати та відтворювати. Схожа рольова комунікація реалізована в ChatDev, де спеціалізовані агенти взаємодіють під час проектування, програмування й тестування (Qian et al., 2024), тоді як Reflexion і Self-Refine демонструють цінність мовного зворотного зв'язку та ітеративного уточнення результату (Madaan et al., 2023; Shinn et al., 2023).

Наведені приклади показують, що мультиагентна парадигма не задає єдиного типу архітектури. Спільним є лише принцип розподіленої відповідальності: складна задача подається як система ролей і взаємодій.

Мультиагентність як модель організації філологічної роботи з науковим текстом

У філологічному контексті найсуттєвішою рисою мультиагентної системи є можливість відокремити мовне генерування від рішень, які йому передують. Зв'язний текст часто створює ілюзію, ніби термінологічний вибір, концептуальна структура й композиція виникли одночасно. Насправді це різні операції. Коли вони винесені в окремі етапи, дослідник отримує змогу перевіряти не лише фінальне формулювання, а й те, на якій термінологічній і концептуальній основі воно з'явилося. Онтологічно-кероване генерування додає до цієї організації ще один рівень дисципліни: між джерелом і текстом з'являється структуроване представлення знання. Воно не замінює інтерпретацію, проте задає межі, у яких LLM формує мовну реалізацію. Принциповим є ще одне обмеження: мультиагентний підхід не ототожнюється з повною автономією наукового дослідження. Навіть якщо технічно весь конвеєр може бути запущений без втручання людини, інтерпретація проміжних і фінальних результатів залишається відповідальністю дослідника. У запропонованій схемі агенти автоматизують операції з інформацією та текстом, але не скасовують потреби перевіряти джерела, оцінювати коректність дефініцій, приймати рішення щодо прийнятності концептуальних зв'язків і визначати, чи впливають висновки з опрацьованого матеріалу. Тому мультиагентність доцільно трактувати як інструмент керованої декомпозиції й

підтримки дослідницької роботи, а не як заміну суб'єкта наукового пізнання.

Викладені теоретичні положення потребують перевірки на конкретному матеріалі. Така перевірка має полягати не в демонстрації того, що кілька мовних моделей здатні обмінюватися повідомленнями, а в побудові потоку, у якому кожна роль відповідає реальній операції роботи з науковим матеріалом і залишає після себе інспектований результат. Опис відповідної реалізації – семиагентного конвеєра в середовищі Langflow, принципів добору моделей для окремих ролей, перебігу тестового запуску та отриманих проміжних артефактів – становить предмет окремої публікації.

Перспективи подальших досліджень

Теоретичний аналіз мультиагентного підходу окреслює кілька напрямів подальшого опрацювання. Першим із них є розроблення типології агентних ролей для філологічних задач. Наведений поділ на джерельну, термінологічну, концептуальну, темпоральну, композиційну, генеративну та контрольну функції є робочим, але не єдино можливим. Відкритим залишається питання про те, які операції справді доцільно розділяти, а які втрачають зміст поза цілісним контекстом. Для лексикографічних і термінознавчих задач склад ролей імовірно буде іншим, ніж для аналізу художнього тексту чи підготовки наукового огляду.

Другим напрямом є формалізація критеріїв добору моделі під конкретну роль. Твердження про те, що одна модель «краща» за іншу, не має операційного змісту доти, доки не визначено критеріїв та їхньої вагомості. Логіка методу аналізу ієрархій (Saaty, 1980) дає для цього придатний апарат, однак його застосування до мовних моделей потребує окремого опрацювання: вибору релевантних критеріїв для кожного типу мовно-аналітичної операції, способів отримання оцінок і процедур їх перегляду з появою нових версій моделей.

Третім напрямом є потреба в метриках, що оцінюють не лише фінальний текст, а й проміжні артефакти. Наявні підходи до оцінювання порожденного тексту зосереджені на кінцевому результаті (Celikyilmaz et al., 2020; van der Lee et al., 2021; Liu et al., 2023), тоді як специфіка мультиагентної архітектури полягає саме в наявності проміжних станів. Потрібні методики, які дають змогу оцінювати повноту корпусу, коректність термінологічної бази, обґрунтованість концептуальних зв'язків і відповідність плану заявленій структурі знань – окремо один від одного.

Четвертий напрям пов'язаний з українськомовним матеріалом. Термінологічний і концеп-

туальний рівні опрацювання особливо чутливі до стану терміносистеми: у багатьох галузях паралельно функціонують англійські терміни, кілька українських відповідників, кальки та ще не установлені новотвори. У такому середовищі роль, відповідальна за термінологію, має не лише вилучати одиниці, а й зіставляти варіанти, фіксувати джерела їх уживання та розрізняти нормативний і фактичний статус, а роль, що будує концептуальну структуру, – вирішувати, чи належать мовні варіанти одному концептові. Це перетворює мультиагентну схему із засобу генерування тексту на інструмент дослідження терміносистеми (Пасічник, Яромич, 2025b).

Нарешті, окремого опрацювання потребує розподіл відповідальності між дослідником і системою. Повна автономність не повинна бути самоціллю, особливо коли йдеться про науковий текст. Продуктивнішим є визначення контрольних точок, у яких рішення людини має найбільшу методологічну вагу: затвердження корпусу, нормалізація спірних термінів, прийняття або відхилення неоднозначних концептуальних зв'язків, погодження плану. За такої моделі агенти виконують масштабовані й повторювані операції, а дослідник зберігає відповідальність за інтерпретаційні рішення.

Висновки. Мультиагентний підхід у цій статті розглядався не як спосіб механічно збільшити кількість мовних моделей, задіяних у роботі, а як спосіб організувати складну процедуру. Замість прямого переходу від промпта до тексту він передбачає розділення джерельного, термінологічного, концептуального, темпорального, композиційного, генеративного та контрольного рівнів. Для філологічної роботи це принципово, оскільки рішення, які в гладко написаному тексті зливаються в один результат, стають окремими об'єктами перевірки.

Проведений аналіз показує, що мультиагентність не є наслідком появи великих мовних моделей. Вона сформувалася значно раніше – у межах досліджень розподіленого штучного інтелекту, де центральною була проблема спільної роботи кількох автономних компонентів. Спадкоємними залишилися базові властивості агента: автономність, реактивність, проактивність і соціальність. Принципово новим є те, що роль окремого агента тепер значною мірою задається природномовною інструкцією, а не програмним кодом, унаслідок чого вартість створення спеціалізованого виконавця різко знизилася.

Водночас підхід не слід трактувати як засіб автоматично усунути притаманні мовним моделям ризики. Кількість агентів сама по собі не гарантує фактологічної точності, а передавання результату від одного агента до іншого може, навпаки, поширювати ранню помилку на наступні етапи. Тому принциповою є наявність контрольних точок і можливість простежити походження кожного суттєвого елемента фінального тексту.

Для філологічних досліджень найсуттєвішим є те, що мультиагентна організація дозволяє відокремити мовне генерування від рішень, які йому передують. Зв'язний текст часто створює ілюзію, ніби термінологічний вибір, концептуальна структура й композиція виникли одночасно; насправді це різні операції. Онтологічне подання знань додає до цієї організації ще один рівень дисципліни: між джерелом і текстом з'являється структуроване представлення, яке не замінює інтерпретацію, проте задає межі, у яких мовна модель формує мовну реалізацію.

Викладені положення становлять теоретичну основу для експериментальної перевірки, опис якої подано в окремій публікації. Її предметом є семиагентний конвеєр онтологічно-керованого генерування, реалізований у середовищі Langflow, і аналіз того, наскільки описані тут переваги виявляються на конкретному матеріалі.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Пасічник В. В., Яромич М. В. Великі мовні моделі та онтології у філологічних дослідженнях: аналітичний огляд джерел. Актуальні питання гуманітарних наук. 2025. Вип. 83, т. 3. С. 236–250. DOI: <https://doi.org/10.24919/2308-4863/83-3-35>
2. Пасічник В. В., Яромич М. В. Інформаційна технологія автоматизованого реферування наукових текстів засобами великих мовних моделей. Науковий вісник НЛТУ України. 2026. Т. 36, № 1. С. 105–117. DOI: <https://doi.org/10.36930/40360112>
3. Пасічник В. В., Яромич М. В. Інформаційна технологія лексикографічного аналізу з використанням великих мовних моделей (на прикладі проблемної області «Штучний інтелект»). Актуальні питання гуманітарних наук. 2025. Вип. 92, т. 2. С. 239–255. DOI: <https://doi.org/10.24919/2308-4863/92-2-34>
4. Пасічник В. В., Яромич М. В. Концептуалізація предметної області «Штучний інтелект» з використанням інструментів великих мовних моделей. Вісник Житомирського державного університету імені Івана Франка. Філологічні науки. 2025. Вип. 2 (105). С. 180–200. DOI: [https://doi.org/10.35433/philology.2\(105\).2025.180-200](https://doi.org/10.35433/philology.2(105).2025.180-200)
5. Пасічник В. В., Яромич М. В. Порівняльний аналіз великих мовних моделей в контексті вирішення лінгвістичних задач. Актуальні питання гуманітарних наук. 2025. Вип. 89, т. 1. С. 288–305. DOI: <https://doi.org/10.24919/2308-4863/89-1-41>

6. Celikyilmaz A., Clark E., Gao J. Evaluation of text generation: a survey : preprint. arXiv, 2020. DOI: <https://doi.org/10.48550/arXiv.2006.14799>
7. Dessi D., Osborne F., Reforgiato Recupero D. et al. Generating knowledge graphs by employing Natural Language Processing and Machine Learning techniques within the scholarly domain. *Future Generation Computer Systems*. 2021. Vol. 116. P. 253–264. DOI: <https://doi.org/10.1016/j.future.2020.10.026>
8. Gao L., Dai Z., Pasupat P. et al. RARR: researching and revising what language models say, using language models. *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Toronto : Association for Computational Linguistics, 2023. P. 16477–16508. DOI: <https://doi.org/10.18653/v1/2023.acl-long.910>
9. Gruber T. R. A translation approach to portable ontology specifications. *Knowledge Acquisition*. 1993. Vol. 5, № 2. P. 199–220. DOI: <https://doi.org/10.1006/knac.1993.1008>
10. Guo T., Chen X., Wang Y. et al. Large language model based multi-agents: a survey of progress and challenges. *Proceedings of the Thirty-Third International Joint Conference on Artificial Intelligence*. Jeju : IJCAI Organization, 2024. P. 8048–8057. DOI: <https://doi.org/10.24963/ijcai.2024/890>
11. Han S., Zhang Q., Yao Y. et al. LLM multi-agent systems: challenges and open problems : preprint. arXiv, 2024. DOI: <https://doi.org/10.48550/arXiv.2402.03578>
12. Hong S., Zhuge M., Chen J. et al. MetaGPT: meta programming for a multi-agent collaborative framework. *The Twelfth International Conference on Learning Representations*. Vienna, 2024. URL: <https://openreview.net/forum?id=VtmBAGCN7o> (дата звернення: 14.08.2026).
13. Jennings N. R., Sycara K., Wooldridge M. A roadmap of agent research and development. *Autonomous Agents and Multi-Agent Systems*. 1998. Vol. 1. P. 7–38. DOI: <https://doi.org/10.1023/A:1010090405266>
14. Ji Z., Lee N., Frieske R. et al. Survey of hallucination in natural language generation. *ACM Computing Surveys*. 2023. Vol. 55, № 12. Article 248. DOI: <https://doi.org/10.1145/3571730>
15. Lewis P., Perez E., Piktus A. et al. Retrieval-augmented generation for knowledge-intensive NLP tasks. *Advances in Neural Information Processing Systems*. 2020. Vol. 33. P. 9459–9474. URL: <https://proceedings.neurips.cc/paper/2020/hash/6b493230205f780e1bc26945df7481e5-Abstract.html> (дата звернення: 14.08.2026).
16. Liu Y., Iter D., Xu Y. et al. G-Eval: NLG evaluation using GPT-4 with better human alignment. *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*. Singapore : Association for Computational Linguistics, 2023. P. 2511–2522. DOI: <https://doi.org/10.18653/v1/2023.emnlp-main.153>
17. Madaan A., Tandon N., Gupta P. et al. Self-Refine: iterative refinement with self-feedback. *Advances in Neural Information Processing Systems*. 2023. Vol. 36. DOI: <https://doi.org/10.52202/075280-2019>
18. Neuhaus F. Ontologies in the era of large language models – a perspective. *Applied Ontology*. 2023. Vol. 18, № 4. P. 399–407. DOI: <https://doi.org/10.3233/AO-230072>
19. Noy N. F., McGuinness D. L. *Ontology development 101: a guide to creating your first ontology : technical report KSL-01-05*. Stanford : Stanford Knowledge Systems Laboratory, 2001. URL: https://protege.stanford.edu/publications/ontology_development/ontology101-noy-mcguinness.html (дата звернення: 14.08.2026).
20. Park J. S., O'Brien J. C., Cai C. J. et al. Generative agents: interactive simulacra of human behavior. *Proceedings of the 36th Annual ACM Symposium on User Interface Software and Technology*. San Francisco : ACM, 2023. Article 2. P. 1–22. DOI: <https://doi.org/10.1145/3586183.3606763>
21. Qian C., Liu W., Liu H. et al. ChatDev: communicative agents for software development. *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Bangkok : Association for Computational Linguistics, 2024. P. 15174–15186. DOI: <https://doi.org/10.18653/v1/2024.acl-long.810>
22. Saaty T. L. *The analytic hierarchy process: planning, priority setting, resource allocation*. New York : McGraw-Hill, 1980. 287 p.
23. Shinn N., Cassano F., Gopinath A. et al. Reflexion: language agents with verbal reinforcement learning. *Advances in Neural Information Processing Systems*. 2023. Vol. 36. DOI: <https://doi.org/10.52202/075280-0377>
24. van der Lee C., Gatt A., van Miltenburg E., Krahmer E. Human evaluation of automatically generated text: current trends and best practice guidelines. *Computer Speech & Language*. 2021. Vol. 67. Article 101151. DOI: <https://doi.org/10.1016/j.csl.2020.101151>
25. Vaswani A., Shazeer N., Parmar N. et al. Attention is all you need. *Advances in Neural Information Processing Systems*. 2017. Vol. 30. P. 5998–6008.
26. Wooldridge M. *An introduction to multiagent systems*. 2nd ed. Chichester : Wiley, 2009. 484 p.
27. Wu Q., Bansal G., Zhang J. et al. AutoGen: enabling next-gen LLM applications via multi-agent conversation : preprint. arXiv, 2023. DOI: <https://doi.org/10.48550/arXiv.2308.08155>
28. Yao S., Zhao J., Yu D. et al. ReAct: synergizing reasoning and acting in language models. *The Eleventh International Conference on Learning Representations*. Kigali, 2023. URL: https://openreview.net/forum?id=WE_vluYUL-X (дата звернення: 14.08.2026).

REFERENCES

1. Pasichnyk V. V., Yaromych M. V. (2025) Velyki movni modeli ta ontolohii u filolohichnykh doslidzhenniakh: analitychnyi ohliad dzherel [Large language models and ontologies in philological research: an analytical review of sources]. *Aktualni pytannia humanitarnykh nauk*, 83 (3), 236–250. DOI: <https://doi.org/10.24919/2308-4863/83-3-35> [in Ukrainian].
2. Pasichnyk V. V., Yaromych M. V. (2026) Informatsiina tekhnolohiia avtomatyzovanoho referuvannia naukovykh tekstiv zasobamy velykykh movnykh modelei [Information technology for automated summarisation of scientific texts by means

- of large language models]. *Naukovyi visnyk NLTU Ukrainy*, 36 (1), 105–117. DOI: <https://doi.org/10.36930/40360112> [in Ukrainian].
3. Pasichnyk V. V., Yaromych M. V. (2025) Informatsiina tekhnolohiia leksykohefichnoho analizu z vykorystanniam velykykh movnykh modelei (na prykladi problemnoi oblasti “Shtuchnyi intelekt”) [Information technology of lexicographic analysis using large language models (on the example of the problem area “Artificial intelligence”)]. *Aktualni pytannia humanitarnykh nauk*, 92 (2), 239–255. DOI: <https://doi.org/10.24919/2308-4863/92-2-34> [in Ukrainian].
4. Pasichnyk V. V., Yaromych M. V. (2025) Kontseptualizatsiia predmetnoi oblasti “Shtuchnyi intelekt” z vykorystanniam instrumentiv velykykh movnykh modelei [Conceptualisation of the subject area “Artificial intelligence” using large language model tools]. *Visnyk Zhytomyrskoho derzhavnoho universytetu imeni Ivana Franka. Filolohichni nauky*, 2 (105), 180–200. DOI: [https://doi.org/10.35433/philology.2\(105\).2025.180-200](https://doi.org/10.35433/philology.2(105).2025.180-200) [in Ukrainian].
5. Pasichnyk V. V., Yaromych M. V. (2025) Porivnialnyi analiz velykykh movnykh modelei v konteksti vyrishennia linhvystychnykh zadach [Comparative analysis of large language models in the context of solving linguistic tasks]. *Aktualni pytannia humanitarnykh nauk*, 89 (1), 288–305. DOI: <https://doi.org/10.24919/2308-4863/89-1-41> [in Ukrainian].
6. Celikyilmaz A., Clark E., Gao J. (2020) Evaluation of text generation: a survey. Preprint. arXiv. DOI: <https://doi.org/10.48550/arXiv.2006.14799>
7. Dessi D., Osborne F., Reforgiato Recupero D., Buscaldi D., Motta E. (2021) Generating knowledge graphs by employing Natural Language Processing and Machine Learning techniques within the scholarly domain. *Future Generation Computer Systems*, 116, 253–264. DOI: <https://doi.org/10.1016/j.future.2020.10.026>
8. Gao L., Dai Z., Pasupat P., Chen A., Chaganty A. T., Fan Y., Zhao V., Lao N., Lee H., Juan D.-C., Guu K. (2023) RARR: researching and revising what language models say, using language models. *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics*, 16477–16508. DOI: <https://doi.org/10.18653/v1/2023.acl-long.910>
9. Gruber T. R. (1993) A translation approach to portable ontology specifications. *Knowledge Acquisition*, 5 (2), 199–220. DOI: <https://doi.org/10.1006/knac.1993.1008>
10. Guo T., Chen X., Wang Y., Chang R., Pei S., Chawla N. V., Wiest O., Zhang X. (2024) Large language model based multi-agents: a survey of progress and challenges. *Proceedings of the Thirty-Third International Joint Conference on Artificial Intelligence*, 8048–8057. DOI: <https://doi.org/10.24963/ijcai.2024/890>
11. Han S., Zhang Q., Yao Y., Jin W., Xu Z., He C. (2024) LLM multi-agent systems: challenges and open problems. Preprint. arXiv. DOI: <https://doi.org/10.48550/arXiv.2402.03578>
12. Hong S., Zhuge M., Chen J., Zheng X., Cheng Y., Zhang C., Wang J., Wang Z., Yau S. K. S., Lin Z., Zhou L., Ran C., Xiao L., Wu C., Schmidhuber J. (2024) MetaGPT: meta programming for a multi-agent collaborative framework. *The Twelfth International Conference on Learning Representations*. URL: <https://openreview.net/forum?id=VtmBAGCN7o>
13. Jennings N. R., Sycara K., Wooldridge M. (1998) A roadmap of agent research and development. *Autonomous Agents and Multi-Agent Systems*, 1, 7–38. DOI: <https://doi.org/10.1023/A:1010090405266>
14. Ji Z., Lee N., Frieske R., Yu T., Su D., Xu Y., Ishii E., Bang Y. J., Madotto A., Fung P. (2023) Survey of hallucination in natural language generation. *ACM Computing Surveys*, 55 (12), Article 248. DOI: <https://doi.org/10.1145/3571730>
15. Lewis P., Perez E., Piktus A., Petroni F., Karpukhin V., Goyal N., Küttler H., Lewis M., Yih W.-T., Rocktäschel T., Riedel S., Kiela D. (2020) Retrieval-augmented generation for knowledge-intensive NLP tasks. *Advances in Neural Information Processing Systems*, 33, 9459–9474. URL: <https://proceedings.neurips.cc/paper/2020/hash/6b493230205f780e1b-c26945df7481e5-Abstract.html>
16. Liu Y., Iter D., Xu Y., Wang S., Xu R., Zhu C. (2023) G-Eval: NLG evaluation using GPT-4 with better human alignment. *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, 2511–2522. DOI: <https://doi.org/10.18653/v1/2023.emnlp-main.153>
17. Madaan A., Tandon N., Gupta P., Hallinan S., Gao L., Wiegrefe S., Alon U., Dziri N., Prabhume S., Yang Y., Gupta S., Majumder B. P., Hermann K., Welleck S., Yazdanbakhsh A., Clark P. (2023) Self-Refine: iterative refinement with self-feedback. *Advances in Neural Information Processing Systems*, 36. DOI: <https://doi.org/10.52202/075280-2019>
18. Neuhaus F. (2023) Ontologies in the era of large language models – a perspective. *Applied Ontology*, 18 (4), 399–407. DOI: <https://doi.org/10.3233/AO-230072>
19. Noy N. F., McGuinness D. L. (2001) *Ontology development 101: a guide to creating your first ontology*. Technical report KSL-01-05. Stanford Knowledge Systems Laboratory. URL: https://protege.stanford.edu/publications/ontology_development/ontology101-noy-mcguinness.html
20. Park J. S., O’Brien J. C., Cai C. J., Morris M. R., Liang P., Bernstein M. S. (2023) Generative agents: interactive simulacra of human behavior. *Proceedings of the 36th Annual ACM Symposium on User Interface Software and Technology*, Article 2, 1–22. DOI: <https://doi.org/10.1145/3586183.3606763>
21. Qian C., Liu W., Liu H., Chen N., Dang Y., Li J., Yang C., Chen W., Su Y., Cong X., Xu J., Li D., Liu Z., Sun M. (2024) ChatDev: communicative agents for software development. *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics*, 15174–15186. DOI: <https://doi.org/10.18653/v1/2024.acl-long.810>
22. Saaty T. L. (1980) *The analytic hierarchy process: planning, priority setting, resource allocation*. New York: McGraw-Hill. 287 p.
23. Shinn N., Cassano F., Gopinath A., Narasimhan K., Yao S. (2023) Reflexion: language agents with verbal reinforcement learning. *Advances in Neural Information Processing Systems*, 36. DOI: <https://doi.org/10.52202/075280-0377>
24. van der Lee C., Gatt A., van Miltenburg E., Krahmer E. (2021) Human evaluation of automatically generated text: current trends and best practice guidelines. *Computer Speech & Language*, 67, Article 101151. DOI: <https://doi.org/10.1016/j.csl.2020.101151>

25. Vaswani A., Shazeer N., Parmar N., Uszkoreit J., Jones L., Gomez A. N., Kaiser Ł., Polosukhin I. (2017) Attention is all you need. *Advances in Neural Information Processing Systems*, 30, 5998–6008.
26. Wooldridge M. (2009) *An introduction to multiagent systems*. 2nd ed. Chichester: Wiley. 484 p.
27. Wu Q., Bansal G., Zhang J., Wu Y., Li B., Zhu E., Jiang L., Zhang X., Zhang S., Liu J., Awadallah A. H., White R. W., Burger D., Wang C. (2023) AutoGen: enabling next-gen LLM applications via multi-agent conversation. Preprint. arXiv. DOI: <https://doi.org/10.48550/arXiv.2308.08155>
28. Yao S., Zhao J., Yu D., Du N., Shafran I., Narasimhan K., Cao Y. (2023) ReAct: synergizing reasoning and acting in language models. *The Eleventh International Conference on Learning Representations*. URL: https://openreview.net/forum?id=WE_vluYUL-X

Дата першого надходження статті до видання: 01.09.2026

Дата прийняття статті до друку після рецензування: 25.09.2026

Дата публікації (оприлюднення) статті: 16.10.2026

Стаття поширюється на умовах
ліцензії відкритого доступу (CC BY 4.0)

