Інформаційні технології, комп'ютерні науки

УДК 004.415.2:004.8

Киселевич Володимир Володимирович
ORCID: 0009-0006-6449-710X
Асистент кафедри комп'ютерних наук та інформаційних технологій
Житомирський державний університет імені Івана Франка

ПОРІВНЯЛЬНИЙ АНАЛІЗ АРХІТЕКТУРНИХ ПІДХОДІВ ДО ПОБУДОВИ АГЕНТНИХ СИСТЕМ ЗА КРИТЕРІЯМИ НАДІЙНОСТІ

Зіставлено шість базових архітектурних підходів до побудови програмних систем на основі агентів з великими мовними моделями за п'ятьма критеріями, а саме складністю розв'язуваних задач, вартістю, передбачуваністю, надійністю і простотою впровадження. Обґрунтовано методологічну неспроможність числової шкали від 1 до 5 для такого зіставлення та запропоновано порядкову якісну шкалу з обов'язковою атрибуцією кожної оцінки до конкретного джерела. Побудовано власну модель накопичення помилок за глибиною траєкторії, яка дає кількісне обґрунтування оцінок за критерієм надійності. Показано, що жоден із розглянутих підходів не має високих оцінок одночасно за всіма критеріями, тому побудоване зведення читається як профіль компромісів, а не як рейтинг. Ключові слова: агентні системи, великі мовні моделі, програмна архітектура, надійність, порядкова шкала, накопичення помилок, верифікаційний шар.

Постановка проблеми. Проєктувальник програмної системи на основі великих мовних моделей обирає архітектуру раніше, ніж отримує емпіричні підстави для вибору. Контрольоване зіставлення 260 конфігурацій на шести агентних бенчмарках і трьох сім'ях моделей показало, що перехід від одноагентної базової лінії до мультиагентної змінює продуктивність у широких межах, від приросту 80,8 відсотка на декомпонованому фінансовому міркуванні до падіння 70,0 відсотка на послідовному плануванні [1]. Отже, саме лише збільшення кількості агентів не гарантує якості, а знак ефекту визначається типом задачі. Аналіз понад 1600 анотованих трас із семи поширених мультиагентних фреймворків виявив 14 унікальних режимів відмов, згрупованих у три категорії, а саме хиби проєктування системи, неузгодженість між агентами і недостатню верифікацію завдання [2]. Порівняння архітектур додатково ускладнюється тим, що агентні оцінювання зосереджені переважно на точності й не враховують вартість, через що надмірно складні та дорогі конфігурації видаються кращими без відповідних підстав [3]. Наслідком є брак придатного інструменту, який давав би змогу зіставляти архітектурні підходи за кількома критеріями одночасно, не приписуючи вихідним даним тієї точності, якої вони не мають.

Аналіз останніх досліджень. Внутрішні обмеження одноагентних архітектур систематизовано в рецензованій праці з програмної архітектури мультиагентних систем, де названо відсутність налаштування на завдання, відсутність пам'яті та обмежений доступ до перевірної істини, а у відповідь на це запропоновано дворівневий поділ на операційний рівень і рівень знань [4]. Патерн з провідним агентом, який планує, відстежує прогрес і перепланує роботу після виявлення помилок, описано в системі Magentic-One, що досягає статистично конкурентних результатів на GAIA, AssistantBench і WebArena без зміни базових можливостей окремих агентів та способу їхньої взаємодії [5]. Значення програмної оболонки (*harness*) підтверджено в SWE-agent, де саме інтерфейс між агентом і комп'ютером, а не заміна моделі, забезпечує показники 12,5 відсотка pass@1 на SWE-bench і 87,7 відсотка на HumanEvalFix [6]. Той самий ефект зафіксовано на 75 змаганнях з інженерії машинного навчання, де конфігурація з оболонкою AIDE досягає рівня бронзової медалі щонайменше в 16,9 відсотка змагань [7]. Стійкість кооперативних структур до дефектних агентів виміряно на чотирьох завданнях і шести системах: ієрархічна структура втрачає 5,5 відсотка якості проти 10,5 і 23,7 відсотка для двох інших структур, а механізми Challenger та Inspector усувають до 96,4 відсотка помилок, допущених дефектними агентами [8]. Верифікаційно-доповнені схеми представлено оцінюванням агентних систем засобами інших агентів на бенчмарку DevAI із 55 реалістичних завдань і 365 ієрархічних вимог користувача [9], тоді як надійність самої суддівської моделі систематизовано в огляді відповідного напряму [10] і кількісно обмежено дослідженням позиційної упередженості на понад 150 000 екземплярах оцінювання, де ця упередженість істотно залежить від розриву в якості порівнюваних рішень [11]. Окремою перешкодою для зіставлення архітектур є відсутність єдиної кодової бази: інтеграція понад 20 усталених методів у спільному середовищі з покроковою перевіркою виходів показала, що дублювання реалізацій спричиняє несправедливі порівняння [12]. Комунікаційно-центрична класифікація мультиагентних систем дає підстави розрізняти оркестраційні патерни за протоколом

та змістом обміну повідомленнями [13]. Варіативність результатів має ще одне джерело, нижчого рівня, оскільки недетермінованість помітно впливає на ймовірності токенів у діапазоні 0,1-0,9 і слабко впливає тоді, коли ці ймовірності близькі до 0 або 1 [14]. Позначені в списку як препринти праці [1; 3; 5; 7; 8; 9; 10; 12; 13; 14] не пройшли рецензування, тому взяті з них числові оцінки трактуються тут як попередні.

Мета. Метою роботи є зіставлення шести базових архітектурних підходів до побудови агентних систем за п'ятьма критеріями (складність розв'язуваних задач, вартість, передбачуваність, надійність, простота впровадження) з обґрунтуванням того, якою шкалою таке зіставлення припустимо подавати. Додатковими завданнями є побудова власної моделі накопичення помилок за глибиною траєкторії для критерію надійності та формулювання умови, за якої зведена таблиця має інтерпретацію.

Виклад основного матеріалу. Для зіставлення виокремлено шість базових архітектурних підходів. Одноагентний підхід передбачає єдиний контур «запит, генерація, відповідь» без зовнішніх дій. Підхід, доповнений інструментами, додає виклики зовнішніх функцій і читання зовнішніх даних, залишаючи одного виконавця. Мультиагентний кооперативний підхід розподіляє роботу між рівноправними агентами, які обмінюються повідомленнями без виділеного керівного вузла. Підхід з оркестратором вводить провідного агента, що декомпонує задачу, стежить за прогресом і перепланує роботу [5]. Верифікаційно-доповнений підхід додає окремий шар перевірки результатів, реалізований суддівською моделлю або агентом-інспектором [8; 9]. Harness-центричний підхід переносить складність із взаємодії агентів у програмну оболонку, тобто в інтерфейс до середовища, набір дозволених операцій та формат зворотного зв'язку [6; 7]. Підставою для такого поділу є те, чим саме підхід розширює клас доступних задач, а саме доступу до зовнішніх дій, розподілу роботи між виконавцями, централізованого керування, окремої перевірки результату або впорядкування середовища виконання. Поділ узгоджується з п'ятьма канонічними архітектурами, виокремленими в контрольованому зіставленні конфігурацій [1], і доповнює це зіставлення розрізненням оболонки та верифікаційного шару.

Поширений спосіб подання такого зіставлення полягає у виставленні кожному підходу балів від 1 до 5 за кожним критерієм. Обґрунтовано, що для розглянутої предметної області ця шкала є методологічно неспроможною, і наведено чотири аргументи. По-перше, критерії вимірюються в неспівмірних одиницях: вартість обчислень під час висновування виражається числом викликів моделі та обсягом опрацьованих токенів, надійність вимірюється часткою успішних запусків, а простота впровадження не має природної метрики взагалі, тому спільна п'ятибальна шкала приховує різнорідність вимірюваних величин. По-друге, присвоєння числа створює ілюзію інтервальної шкали, на якій припустимі арифметичні операції, тоді як наявні дані дають щонайбільше порядкове відношення «нижча за» та «вища за»: з того, що один підхід оцінено в 4 бали, а інший у 2, не випливає подвійна перевага. По-третє, сумування балів за критеріями неявно запроваджує однакові вагові коефіцієнти, чого ніхто не обґрунтовував, хоч для промислового впровадження вартість і передбачуваність важать інакше, ніж для дослідницького прототипу. По-четверте, оцінки походять із різних експериментів на різних наборах задач і з різними базовими моделями, тому числа в різних рядках не є взаємно порівнюваними, а відсутність єдиної кодової бази робить частину зафіксованих приростів наслідком розбіжностей у реалізації [12]. Додатковою перешкодою є те, що навіть повторні запуски тієї самої конфігурації різняться через недетермінованість на рівні ймовірностей токенів [14], а відтворюваність агентних оцінювань залишається низькою [3].

Замість числової шкали запропоновано порядкову якісну шкалу з трьома градаціями (низька, помірна, висока) та з обов'язковою атрибуцією кожної оцінки конкретному джерелу, з якого вона виведена. Клітинка зведеної таблиці містить якісну оцінку і номер джерела у квадратних дужках, тому читач може перевірити походження оцінки та умови, за яких її отримано. Оцінки без джерела до зведення не вносяться. Результат подано у двох таблицях через обмежену ширину сторінки, а саме табл. 1 для трьох критеріїв та табл. 2 для двох критеріїв з додатковим стовпцем основного обмеження підходу.

Таблиця 1

Порядкові оцінки архітектурних підходів за складністю задач, вартістю і передбачуваністю

Підхід	Складність задач	Вартість	Передбачуваність
Одноагентний	низька [4]	низька [1]	помірна [14]
Доповнений інструментами	помірна [6]	помірна [3]	помірна [6]
Мультиагентний кооперативний	висока [1]	висока [3]	низька [2]
З оркестратором	висока [5]	висока [1]	помірна [5]
Верифікаційно-доповнений	висока [9]	висока [11]	помірна [8]
Harness-центричний	помірна [6]	помірна [7]	висока [6]

Джерело: розробка автора.

Оцінки за критерієм надійності обґрунтовано власною моделлю накопичення помилок за глибиною траєкторії. Позначимо через p імовірність коректності окремого кроку траєкторії, через k число послідовних кроків, кожен з яких мусить бути коректним, і через r частку хибних кроків, які виявляє верифікаційний шар за умови однієї повторної спроби. Тоді імовірність коректної траєкторії дорівнює p, піднесеному до степеня k, а за наявності верифікаційного шару вона дорівнює k-му степеню суми, у якій до p додано добуток трьох множників, а саме різниці між одиницею та p, частки r і самої величини p. Модель спирається на припущення про незалежність кроків, яке є спрощенням, бо реальні відмови корелюють через спільний контекст, а також на припущення про сталість p за глибиною і про єдину повторну спробу після виявлення хиби. Розрахунок дає такі значення: за p = 0,95 імовірність коректної траєкторії становить 77,4 відсотка на п’ятому кроці, 59,9 відсотка на десятому і 35,8 відсотка на двадцятому; за p = 0,90 вона становить 34,9 відсотка на десятому кроці та 12,2 відсотка на двадцятому; за p = 0,80 вона падає до 1,2 відсотка на двадцятому кроці. Верифікаційний шар з часткою виявлення r = 0,8 підвищує ці значення відповідно до 88,6 відсотка (p = 0,95, k = 10) та 56,7 відсотка (p = 0,90, k = 20). Сформульовано умову допустимої глибини: для p = 0,90 імовірність коректної траєкторії падає нижче половини вже після шостого кроку, а за наявності верифікаційного шару з r = 0,8 ця межа зсувається до двадцяти чотирьох кроків, тобто глибина кооперації є обмеженим ресурсом, який верифікація розширює приблизно вчетверо, але не робить необмеженим. Модель узгоджується з незалежними вимірюваннями, за якими архітектури без централізованої верифікації поширюють помилки сильніше [1], а інспектувальні механізми відновлюють до 96,4 відсотка помилок дефектних агентів [8].

Таблиця 2

Порядкові оцінки за надійністю і простотою впровадження та основне обмеження підходу

Підхід	Надійність	Простота впровадження	Основне обмеження
Одноагентний	помірна [4]	висока [4]	немає пам’яті та доступу до перевірної істини [4]
Доповнений інструментами	помірна [6]	висока [6]	залежність від якості інтерфейсу до середовища [6]
Мультиагентний кооперативний	низька [2; 8]	низька [12]	каскадне поширення помилок між агентами [8]
З оркестратором	помірна [5]	низька [5]	провідний агент є єдиною точкою відмови [2]
Верифікаційно-доповнений	висока [8]	низька [10]	упередженість суддівської моделі [11]
Harness-центричний	помірна [7]	помірна [7]	оболонка прив’язана до конкретного середовища [7]

Джерело: розробка автора.

Зіставлення табл. 1 і табл. 2 дає ключове твердження роботи: за жодним із шести підходів не досягнуто високих оцінок одночасно за всіма п’ятьма критеріями, причому в кожному рядку присутня щонайменше одна низька або помірна оцінка. Одноагентний та доповнений інструментами підходи мають високу простоту впровадження і низьку вартість, але поступаються складністю розв’язуваних задач [4; 6]. Мультиагентний кооперативний підхід розширює клас доступних задач, проте втрачає в передбачуваності й надійності через каскадне поширення помилок [2; 8]. Верифікаційно-доповнений підхід дає найвищу надійність, але платить за це додатковими викликами моделі та успадковує упередженість суддівського компонента [10; 11]. Harness-центричний підхід забезпечує високу передбачуваність за помірної вартості, проте його оболонка прив’язана до конкретного середовища і потребує повторної розробки для нового класу задач [6; 7]. Звідси випливає умова інтерпретації: побудоване зведення має сенс лише як профіль компромісів, тобто як опис того, чим кожен підхід платить за свої переваги, і не має сенсу як рейтинг, оскільки згортання рядка в один показник вимагало б вагових коефіцієнтів, які в наявних даних відсутні. Практичним наслідком є те, що вибір архітектури визначається профілем вимог проєкту, зокрема допустимою глибиною траєкторії та припустимою вартістю, а не позицією підходу в упорядкованому списку. Комунікаційно-центрична класифікація [13] придатна для подальшого уточнення рядків, що стосуються оркестрованого та кооперативного підходів, бо розрізняє протокол і зміст обміну повідомленнями. Окремо зазначено, що жодна клітинка зведення не є оцінкою підходу як такого, а є оцінкою підходу в умовах того експерименту, з якого її взято.

Обмеження. Робота не містить власного експерименту, а всі порядкові оцінки виведено з вимірювань інших авторів, тому вона не доводить самих цих вимірювань, а лише впорядковує їх. Оцінки в клітинках зведення залежать від наборів задач і базових моделей, використаних у джерелах, тому їх не слід переносити на інші предметні області без повторної перевірки. Побудована модель накопичення помилок спирається на припущення про незалежність кроків та сталість імовірності p за глибиною, тоді як реальні траєкторії містять корельовані відмови через спільний контекст, отже отримані значення є оптимістичною верхньою межею

справжньої надійності. Статистичної перевірки розбіжностей між підходами не виконано, оскільки вихідні праці подають агреговані показники без дисперсії за повторними запусками, а частина використаних джерел є препринтами. Вибірку опрацьованих джерел обмежено чотирнадцятьма працями, відібраними за критерієм наявності кількісних результатів, тому вона не є систематичним оглядом і не претендує на повноту покриття наявних архітектурних варіантів. Порядкова шкала з трьома градаціями також є спрощенням, бо не передає відстані між сусідніми градаціями.

Висновки. Зіставлено шість базових архітектурних підходів до побудови агентних систем за п’ятьма критеріями і показано, що числова шкала від 1 до 5 для такого зіставлення методологічно неспроможна через неспівмірність одиниць, ілюзію інтервальності, неявні однакові вагові коефіцієнти та несумісність вихідних експериментів. Запропоновано порядкову якісну шкалу з обов’язковою атрибуцією кожної оцінки до джерела, за якою побудовано дві зведені таблиці. Побудовано модель накопичення помилок за глибиною траєкторії, з якої сформульовано умову допустимої глибини: за $p = 0{,}90$ межа половинної успішності настає на шостому кроці без верифікації та на двадцять четвертому кроці за частки виявлення $r = 0{,}8$. Установлено, що жоден підхід не має високих оцінок одночасно за всіма критеріями, тому зведення читається як профіль компромісів. Подальші дослідження доцільно спрямувати на емпіричну перевірку припущення про незалежність кроків у межах єдиної кодової бази [12].

1. Kim Y., Gu K., Park C. [та ін.] Towards a Science of Scaling Agent Systems : препринт. arXiv:2512.08296. 2025. URL: https://arxiv.org/abs/2512.08296.
2. Cemri M., Pan M. Z., Yang S. [та ін.] Why Do Multi-Agent LLM Systems Fail? Advances in Neural Information Processing Systems 38. 2025. P. 135676–135729. DOI: 10.52202/085713-4082.
3. Kapoor S., Stroebl B., Siegel Z. S. [та ін.] AI Agents That Matter : препринт. arXiv:2407.01502. 2024. URL: https://arxiv.org/abs/2407.01502.
4. Becattini M., Verdecchia R., Vicario E. SALLMA: A Software Architecture for LLM-Based Multi-Agent Systems. 2025 IEEE/ACM International Workshop New Trends in Software Architecture (SATrends). 2025. P. 5-8. DOI: 10.1109/satrends66715.2025.00006.
5. Fourney A., Bansal G., Mozannar H. [та ін.] Magentic-One: A Generalist Multi-Agent System for Solving Complex Tasks : препринт. arXiv:2411.04468. 2024. URL: https://arxiv.org/abs/2411.04468.
6. Yang J., Jimenez C. E., Wettig A. [та ін.] SWE-agent: Agent-Computer Interfaces Enable Automated Software Engineering. Advances in Neural Information Processing Systems 37. 2024. P. 50528-50652. DOI: 10.52202/079017-1601.
7. Chan J. S., Chowdhury N., Jaffe O. [та ін.] MLE-bench: Evaluating Machine Learning Agents on Machine Learning Engineering : препринт. arXiv:2410.07095. 2024. URL: https://arxiv.org/abs/2410.07095.
8. Huang J., Zhou J., Jin T. [та ін.] On the Resilience of LLM-Based Multi-Agent Collaboration with Faulty Agents : препринт. arXiv:2408.00989. 2024. URL: https://arxiv.org/abs/2408.00989.
9. Zhuge M., Zhao C., Ashley D. [та ін.] Agent-as-a-Judge: Evaluate Agents with Agents : препринт. arXiv:2410.10934. 2024. URL: https://arxiv.org/abs/2410.10934.
10. Gu J., Jiang X., Shi Z. [та ін.] A Survey on LLM-as-a-Judge : препринт. arXiv:2411.15594. 2024. URL: https://arxiv.org/abs/2411.15594.
11. Shi L., Ma C., Liang W. [та ін.] Judging the Judges: A Systematic Study of Position Bias in LLM-as-a-Judge. Proceedings of the 14th International Joint Conference on Natural Language Processing and the 4th Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics. 2025. P. 292–314. DOI: 10.18653/v1/2025.ijcnlp-long.18.
12. Ye R., Huang K., Wu Q. [та ін.] MASLab: A Unified and Comprehensive Codebase for LLM-based Multi-Agent Systems : препринт. arXiv:2505.16988. 2025. URL: https://arxiv.org/abs/2505.16988.
13. Yan B., Zhou Z., Zhang L. [та ін.] Beyond Self-Talk: A Communication-Centric Survey of LLM-Based Multi-Agent Systems : препринт. arXiv:2502.14321. 2025. URL: https://arxiv.org/abs/2502.14321.
14. Fu T., Martínez G., Conde J. [та ін.] Beyond Reproducibility: Token Probabilities Expose Large Language Model Nondeterminism : препринт. arXiv:2601.06118. 2026. URL: https://arxiv.org/abs/2601.06118.