УДК 004.415.53:004.8

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

СПОСТЕРЕЖУВАНІСТЬ LLM-АГЕНТІВ: ЧОМУ КЛАСИЧНІ ЗАСОБИ МОНІТОРИНГУ НЕПРИДАТНІ

Показано, що класичні засоби моніторингу програмних систем (журнали, метрики, розподілене трасування) спираються на припущення про детермінований потік керування, якого агентна система на основі великої мовної моделі не задовольняє. Виокремлено чотири припущення класичного моніторингу, що руйнуються в агентному середовищі. Запропоновано перелік десяти вимог до підсистеми спостережуваності, одиницею телеметрії якої є траєкторія рішення з фіксацією точок розгалуження, викликів інструментів і підстав вибору, та визначено, які з них не покриваються наявними засобами. Ключові слова: спостережуваність, LLM-агент, траєкторія рішення, розподілене трасування, атрибуція відмови, телеметрія, недетермінованість.

Постановка проблеми. Промислова експлуатація програмних систем на основі великих мовних моделей спирається на засоби спостережуваності, розроблені для детермінованих застосунків: журнали подій, числові метрики та розподілене трасування. Ці засоби припускають, що потік керування визначається структурою програмного коду, тому той самий вхід веде до того самого шляху виконання, а відхилення від очікуваного шляху свідчить про дефект. Агентна система порушує цю передумову на рівні механізму: траєкторія, утворена циклами міркування, дії та спостереження, формується під час висновування, а не задається кодом, тому два запуски з однаковим входом можуть відрізнятися послідовністю викликів інструментів, глибиною декомпозиції завдання та кінцевим результатом. Наслідком є розрив між тим, що фіксує підсистема моніторингу (тривалості, коди відповідей, частки успішних запусків), і тим, що потрібно інженерові для встановлення причини відмови, тобто підставою, на якій агент обрав саме цей крок, і переліком відкинутих ним альтернатив. Дослідження видимості агентних систем виокремлює три групи механізмів, а саме ідентифікатори агентів, моніторинг у реальному часі та журнали дій [1], проте одиниця запису та обов'язковий склад полів такого журналу залишаються неунормованими. Масштаб проблеми підтверджує емпіричний аналіз відмов мультиагентних систем: із трас семи поширених фреймворків виведено таксономію з 14 режимів відмови, згрупованих у три категорії (дефекти проєктування системи, неузгодженість між агентами, недостатня верифікація завдання), причому таксономію побудовано на 150 трасах з узгодженістю анотаторів kappa = 0,88 за загальної бази понад 1600 анотованих трас [2].

Аналіз останніх досліджень. Джерело недетермінованості описано на рівні розподілу ймовірностей токенів: варіативність між запусками є суттєвою для токенів з імовірностями в діапазоні 0,1–0,9 і значно меншою, коли ймовірність близька до нуля або одиниці, причому подібні варіації виявлено в усіх перевірених моделях [3]. Обмеженість класичного моніторингу застосунків зафіксовано у фреймворку когнітивної спостережуваності, який відновлює приховані міркування агента і перевірений на агентах AutoCodeRover та OpenHands з набором SWE-bench-lite [4]. Окремий напрям стосується атрибуції причини: наявні засоби відтворюють траєкторії виконання, але майже не допомагають визначити першопричину, а найкраща строга

точність атрибуції на еталоні Who and When становить 28,8 відсотка точних зазначень агента й кроку проти 21,7 відсотка для найсильнішого однопрохідного базового методу [5]. Спробу спертися на наявну інфраструктуру телеметрії втілено в багаторівневій платформі керування системами штучного інтелекту на основі OpenTelemetry, автори якої окремо розглядають недоліки наявних платформ моніторингу продуктивності застосунків [6]. Дослідження моніторингу ненадійних агентних систем показують, що обсяг спостереження визначає тип виявленої відмови: монітори в межах одного запуску виявляли детерміновані дефекти етапів, міжзапускові монітори виявляли стохастичні наслідки інтеграції, а детерміноване сортування спрямовувало 97 відсотків знахідок в автоматичне опрацювання [7]. Проблему звітності сформульовано як перенесення одиниці відтворюваності зі звітованих оцінок на записи запусків [8]. Числові оцінки за препринтами [3; 5; 6; 7; 8; 9] є попередніми.

Мета. Мета роботи полягає в тому, щоб виявити, які конкретні припущення класичних засобів моніторингу руйнуються в агентній системі, і сформулювати на цій основі перелік вимог до підсистеми спостережуваності, одиницею телеметрії якої є не окремий запит, а траєкторія рішення з фіксацією точок розгалуження, викликів інструментів і підстав вибору, а також визначити, які з цих вимог не покриваються наявними засобами моніторингу застосунків і з якої причини.

Виклад основного матеріалу. Класичний моніторинг спирається на чотири припущення, і кожне з них в агентній системі не виконується. Перше припущення стосується стабільності множини шляхів виконання: кількість різних шляхів виконання обмежена структурою коду, тому скінченний набір панелей показників покриває поведінку системи. В агентній системі множина шляхів формується під час висновування, отже перелік панелей не може бути замкненим наперед, а поява раніше не спостереженої послідовності кроків не є ознакою дефекту. Друге припущення стосується відтворюваності відмови за тим самим входом, з чого випливає стратегія налагодження повторним запуском. Оскільки варіативність виникає вже на рівні ймовірностей токенів і за ненульової температури впливає на згенерований текст [3], повторний запуск дає іншу траєкторію, тому відсутність відмови під час повторення не є доказом її усунення. Третє припущення стосується локальності причини збою: у детермінованому застосунку симптом виникає близько до дефекту в стеку викликів. В агентній системі причина може лежати за десять кроків до симптому, бо помилка на ранньому кроці потрапляє в контекст і підвищує ймовірність подальших помилок, тобто діє самозумовлення, яке не зникає зі збільшенням розміру моделі [9]. Четверте припущення стосується інформативності агрегованих метрик: середня частка успіху описує систему, але не пояснює конкретний запуск. Показник у 30 відсотків автономно виконаних завдань на симуляції робочого середовища компанії [10] фіксує розрив із реальними професійними задачами, проте не вказує, на якому кроці й через яку підставу вибору провалився окремий запуск, а різні за обсягом монітори взагалі виявляють різні класи відмов [7].

Запропоновано перелік вимог до підсистеми спостережуваності, у якому одиницею телеметрії є траєкторія рішення. Вимоги згруповано у чотири категорії. Перша категорія стосується одиниці та повноти запису. Вимога 1: траєкторія записується як цілісна одиниця з явним відношенням батьківства між кроками, а не як набір незалежних записів про запити. Вимога 2: кожен виклик інструмента фіксується разом з аргументами, результатом і станом контексту, у якому ухвалено рішення про виклик. Вимога 3: записуються відкинуті альтернативи разом із підставою вибору, а не лише обраний крок. Друга категорія стосується атрибуції причини. Вимога 4: між кроками будується граф інформаційних залежностей, а не тільки часова послідовність, бо крок може залежати від змісту кроку, віддаленого в часі. Вимога 5: симптом відмови пов’язується з кроком-джерелом через окреме відношення атрибуції, яке зазначає і крок, і відповідального агента, що відповідає прийнятій мірі суворої точності атрибуції [5]. Вимога 6: на кожному кроці фіксуються надані повноваження та фактично використані права доступу до зовнішніх інструментів, оскільки емпіричне обстеження серверів протоколу Model Context Protocol виявило загальні уразливості у 7,2 відсотка серверів і специфічне отруєння інструментів у 5,5 відсотка [11]. Запозиченими у цьому переліку є лише окремі емпіричні підстави вимог (міра суворої точності атрибуції [5], частки серверів інструментів, які є уразливими [11]), тоді як склад вимог, їхнє групування та формулювання належать авторові.

Третя категорія стосується порівнюваності між запусками. Вимога 7: разом із траєкторією зберігаються версія моделі, температура та інші параметри висновування, без яких порівняння двох запусків позбавлене

підстав. Вимога 8: підсистема дає змогу згрупувати запуски з однаковим входом і показати розбіжність траєкторій у вигляді дерева спільних префіксів із позначенням точки першого розходження. Вимога 9: правила обчислення похідних показників (частка успішності, облік токенів, час) зберігаються машиночитним описом із версією схеми, бо самі лише правила звітування можуть змінити результат за незмінних свідчень [8]. Четверта категорія стосується керування обсягом даних і придатності записів для аудиту. Вимога 10: обсяг телеметрії регулюється вибірковим записом за політикою, яка обов'язково зберігає повну траєкторію для запусків із відмовою та для контрольної частки успішних запусків, оскільки повний запис усіх міркувань є непропорційно дорогим, тоді як вимоги до свідчень відповідності систем керування штучним інтелектом за ISO/IEC 42001 роблять вибірку об'єктом окремого обґрунтування [12]. Розподіл вимог за категоріями та оцінку покриття наявними засобами подано в табл. 1. Сформульовано умову порівнюваності двох запусків, а саме збіг версії моделі, параметрів висновування та версії схеми обчислення показників, за невиконання якої розбіжність траєкторій не може бути віднесена ні до входу, ні до поведінки агента.

Окремого пояснення потребує те, чому частина вимог не покривається наявними засобами. Розподілене трасування будує дерево викликів за відношенням вкладеності, тому воно фіксує, що крок стався всередині іншого, але не фіксує, що один крок спирається на результат іншого; саме цей другий тип відношення потрібен для атрибуції причини, і його брак пояснює низьку точність автоматичного визначення винного кроку [5]. Числові метрики агрегують за розмірностями зі скінченною множиною значень, тоді як текст запиту такої множини не утворює, тому групування запусків з однаковим входом засобами метрик недосяжне. Нарешті, вибіркове збереження записів у наявних засобах налаштовують за часткою, а не за наслідком запуску, через що саме відмови потрапляють у вибірку з тією самою ймовірністю, що й успішні запуски.

Таблиця 1

Вимоги до підсистеми спостережуваності агентної системи та їхнє покриття класичними засобами

Вимога	**Категорія**	**Покриття класичними засобами**
1. Траєкторія як цілісна одиниця запису	Одиниця й повнота запису	Частково: трасування дає дерево викликів
2. Виклик інструмента з аргументами, результатом і станом контексту	Одиниця й повнота запису	Частково: стан контексту не фіксується
3. Запис відкинутих альтернатив і підстави вибору	Одиниця й повнота запису	Ні: span описує лише виконане
4. Граф інформаційних залежностей між кроками	Атрибуція причини	Ні: батьківство означає вкладеність викликів
5. Відношення атрибуції симптому до кроку й агента	Атрибуція причини	Ні: потрібні окремі засоби атрибуції
6. Повноваження та використані права на кожному кроці	Атрибуція причини	Частково: журнали доступу поза траєкторією
7. Версія моделі й параметри висновування	Порівнюваність між запусками	Так: звичайні атрибути ресурсу
8. Групування запусків з однаковим входом	Порівнюваність між запусками	Ні: кардинальність міток не витримує тексту запиту
9. Версіоновані машиночитні правила обчислення показників	Порівнюваність між запусками	Ні: правила звітування залишаються неявними
10. Вибірковий запис з обов'язковим збереженням відмов	Керованість обсягом і аудит	Частково: вибірка не враховує наслідку запуску

Джерело: розробка автора

Обмеження. Робота не містить власного експерименту, тому запропонований перелік вимог перевірено лише на узгодженість з опублікованими результатами, а не впровадженням у робочу систему. Оцінка покриття вимог наявними засобами спирається на документацію та опубліковані описи, а не на вимірювання в конкретному розгортанні. Частина джерел має статус препринта [3; 5; 6; 7; 8; 9], тому наведені з них числа є попередніми. Перелік із десяти вимог є аналітичною конструкцією: його повноту не перевірено ні опитуванням інженерів, ні аналізом інцидентів, тому не виключено, що для окремих класів агентних систем потрібні додаткові вимоги. Питання вартості зберігання траєкторій і впливу телеметрії на затримку відповіді в роботі не розглядається.

Висновки. Непридатність класичних засобів моніторингу для агентних систем має конкретну причину, а не загальний характер: руйнуються чотири припущення, а саме стабільність множини шляхів виконання, відтворюваність відмови за тим самим входом, локальність причини збою та інформативність агрегованих метрик. Запропоновано перелік з десяти вимог до підсистеми спостережуваності, згрупованих у чотири категорії (одиниця й повнота запису, атрибуція причини, порівнюваність між запусками, керованість обсягом і аудит), у якому одиницею телеметрії є траєкторія рішення з фіксацією точок розгалуження, викликів інструментів і підстав вибору. Установлено, що п’ять із десяти вимог не покриваються наявними засобами, а ще чотири покриваються лише частково, причому причиною є властивості моделі даних, а саме відсутність у span поняття відкинутої альтернативи, обмежена кардинальність міток у метриках і те, що батьківство у трасі відображає вкладеність викликів, а не інформаційну залежність. Подальше дослідження доцільно спрямувати на кількісну оцінку витрат на зберігання траєкторій за різних політик вибіркового запису та на експериментальну перевірку того, наскільки фіксація відкинутих альтернатив підвищує точність атрибуції першопричини.

1. Chan A., Ezell C., Kaufmann M.R. [та ін.] Visibility into AI Agents. The 2024 ACM Conference on Fairness, Accountability, and Transparency. 2024. P. 958–973. DOI: 10.1145/3630106.3658948.
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.
4. 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.
5. Rombaut B., Masoumzadeh S., Vasilevski K. [та ін.] Watson: A Cognitive Observability Framework for the Reasoning of LLM-Powered Agents. 2025 40th IEEE/ACM International Conference on Automated Software Engineering (ASE). 2025. P. 739–751. DOI: 10.1109/ase63991.2025.00067.
6. Zhu K., Ye X., Han Z. [та ін.] AgentDebugX: An Open-Source Toolkit for Failure Observability, Attribution, and Recovery in LLM Agents : препринт. arXiv:2607.18754. 2026. URL: https://arxiv.org/abs/2607.18754.
7. Naik N.K., Saroj A.K., Poudel V.P. [та ін.] Traccia: An OpenTelemetry-Based Governance Platform for AI Systems : препринт. arXiv:2607.14309. 2026. URL: https://arxiv.org/abs/2607.14309.
8. Boston M.F., Hanson G., Georgala E. [та ін.] Monitoring Agentic Systems Before They’re Reliable : препринт. arXiv:2606.02494. 2026. URL: https://arxiv.org/abs/2606.02494.
9. Masters C., Liu Z., Albrecht S.V. Rollout Cards: A Reproducibility Standard for Agent Research : препринт. arXiv:2605.12131. 2026. URL: https://arxiv.org/abs/2605.12131.
10. Sinha A., Arun A., Goel S. [та ін.] The Illusion of Diminishing Returns: Measuring Long Horizon Execution in LLMs : препринт. arXiv:2509.09677. 2025. URL: https://arxiv.org/abs/2509.09677.
11. Xu F. F., Song Y., Li B. [та ін.] TheAgentCompany: Benchmarking LLM Agents on Consequential Real World Tasks. Advances in Neural Information Processing Systems 38. 2025. P. 10291–10321. DOI: 10.52202/085713-0315.
12. Hasan M.M., Li H., Fallahzadeh E. [та ін.] Model Context Protocol (MCP) at First Glance: Studying the Security and Maintainability of MCP Servers. ACM Transactions on Software Engineering and Methodology. 2026. DOI: 10.1145/3814959.
13. Saleh K., Abdulsalam H.M. Operationalizing ISO/IEC 42001: Requirements and Conformance Evidence for AI Management Systems. 2025 International Conference on Artificial Intelligence for Sustainable Innovation (AI-SI). 2025. P. 1–4. DOI: 10.1109/ai-si66213.2025.11341700.