УДК 004.89
DOI https://doi.org/10.32782/2521-6643-2025-2-70.15

Корабльов М. М., доктор технічних наук, професор,
професор кафедри комп'ютерних інтелектуальних технологій
та систем
Харківського національного університету радіоелектроніки
ORCID: 0009-0005-2540-7741

Новосельцев І. В., кандидат технічних наук,
незалежний дослідник
ORCID: 0009-0004-7353-7498

Кобзев І. В., кандидат технічних наук, доцент,
доцент кафедри мультимедійних систем і технологій
Харківського національного економічного університету
імені С. Кузнеця
ORCID: 0000-0002-7182-5814

Ткачук О. К., аспірант кафедри комп'ютерних
інтелектуальних технологій та систем
Харківського національного університету радіоелектроніки
ORCID: 0009-0006-2943-9887

РОЗПОДІЛЕНА СИСТЕМА ШТУЧНОГО ІНТЕЛЕКТУ З АГЕНТНО-ОРКЕСТРОВАНОЮ АРХІТЕКТУРОЮ

У сучасних системах штучного інтелекту агенти відіграють ключову роль у створенні гнучких та адаптивних робочих процесів, і їхня важливість продовжує зростати. Агенти – це невеликі компоненти, які виконують цілеспрямовані завдання та обмінюються результатами за допомогою чітких правил, що робить їх корисними для побудови надійних адаптивних систем. У цій статті представлено архітектуру, оркестровану агентами, для адаптивних систем штучного інтелекту. Вона складається з оркестратора та агентів домену, які працюють разом. Оркестратор підтримує невеликий план із захищеними кроками, застосовує чіткі правила, коли вхідні дані відсутні або впевненість низька, та записує потік для кожного випадку для аудиту. Агенти домену (моделі, інструменти, сервіси) підключаються за стабільними контрактами та обробляють спеціалізовані завдання. Як практична реалізація, архітектура демонструється за допомогою робочого процесу діагностики меланоми: один агент збирає структуровані відповіді за допомогою цілеспрямованих запитань, а інший надає оцінку ризику на основі зображень. Оркестратор поєднує обидва сигнали та застосовує два пороги до оцінки, щоб визначити наступну дію – заспокоїти нагадуванням, запросити один або два подальших огляди або чіткіше фото, або порекомендувати особисте обстеження, реєструючи кожне рішення. Робочий процес є практичним, піддається аудиту та адаптується до місцевої практики без додавання складності. Запропонована архітектура застосовна до областей, де невизначеність та часткова інформація є поширеними, забезпечуючи структурований спосіб забезпечення безпеки, зрозумілості та адаптивності систем. За межами медичної сфери підхід узагальнюється на реагування на інциденти, фінансовий моніторинг та підтримку клієнтів, де адаптивність є критично важливою. Внесок полягає в поєднанні оркестрації, міркування та спостережуваності як першокласних елементів дизайну, пропонуючи відтворювану основу для створення безпечніших та готових до регулювання систем штучного інтелекту.

Ключові слова: *штучний інтелект, агентно-орієнтована архітектура, агенти, правила прийняття рішень, оркестратор, обробка невизначеностей.*

Korablyov M. M., Novoseltsev I. V., Kobzev I. V., Tkachuk O. K. Distributed artificial intelligence system with agent-orchestrated architecture

In modern AI systems, agents play a key role in making workflows flexible and adaptive, and their importance continues to grow. Agents are small components that perform focused tasks and exchange results through clear rules, which makes them useful for building reliable adaptive systems. This paper introduces an Agent-Orchestrated Architecture for adaptive AI systems. It consists of an Orchestrator and Domain Agents working together. The Orchestrator maintains a small plan with guarded steps,

applies clear rules when inputs are missing or confidence is low, and records a per-case thread for audit. Domain Agents (models, tools, services) plug in behind stable contracts and handle specialized tasks. As a practical implementation, the architecture is demonstrated through a Melanoma Diagnostic Workflow: one agent gathers structured answers through focused questions and another provides a risk score from images. The Orchestrator combines both signals and applies two thresholds on the score to determine the next action – reassure with a reminder, request one or two follow-ups or a clearer photo, or recommend an in person exam – while logging every decision. The workflow is practical, auditable, and adjustable to local practice without adding complexity. The proposed architecture is applicable to domains where uncertainty and partial information are common, providing a structured way to keep systems safe, explainable, and adaptable. Beyond the medical domain, the approach generalizes to incident response, financial monitoring, and customer support, where adaptability is critical. The contribution lies in combining orchestration, reasoning, and observability as first – class design elements, offering a reproducible framework for building safer and regulation-ready AI systems.

Key words: *artificial intelligence, agent-oriented architecture, agents, decision rules, orchestrator, uncertainty handling.*

Постановка проблеми. Більшість сучасних систем штучного інтелекту є модельно-орієнтованими. Вони побудовані навколо одного кроку прогнозування та припускають чіткі вхідні дані та надійні послуги. Реальні проблеми відрізняються: вони багатоетапні, вхідні дані можуть бути відсутніми або невпорядкованими, інструменти можуть давати збої, впевненість у моделі може змінюватися, і людині може знадобитися втручання. Відсутність адаптивності обмежує застосування систем штучного інтелекту в середовищі, де надійність має бути продемонстрована, а не передбачувана. У таких умовах конвеєри повинні не лише надавати прогнози, але й робити процес прийняття рішень прозорим, особливо коли вхідні дані невизначені або інструменти дають збій. Щоб бути надійними на практиці, таким системам потрібні механізми, які виявляють невизначеність, застосовують коригувальні дії та роблять кожне коригування зрозумілим для рецензентів.

У сучасних системах штучного інтелекту конвеєри є критично важливими – вони виконують реальну роботу з переміщення даних через моделі, перевірки результатів, запуску наступного кроку та підключення до користувачів. Конвеєр штучного інтелекту – це шлях від вхідних даних до рішень: дані збираються, очищуються, проходять через модель, результати обробляються, приймається рішення, і користувач отримує повідомлення. Коли вхідні дані відсутні, інструменти ламаються або модель невпевнена, конвеєр не адаптується сам по собі – зазвичай для його зміни потрібна людина.

Ця робота знаходиться на перетині агентних фреймворків, робочого процесу/оркестрації, керування часом виконання та спостережуваності.

Аналіз існуючих досліджень та публікацій. Ранні агенти типу «міркування-дія» продемонстрували, що чергування явного міркування з використанням інструментів може покращити успіх завдання, роблячи проміжні рішення спостережуваними. Виявляючи думки як структуровані кроки та пов'язуючи їх з конкретними викликами інструментів, ці системи забезпечували чіткіші сліди дій та надійнішу обробку винятків. ReAct (Reason + Act) є прикладом цього напрямку, показуючи, як мовні моделі можуть чергувати сліди міркувань із діями, що відповідають конкретним завданням, виявляючи проміжні рішення під час взаємодії із зовнішніми інструментами або середовищами [1]. AutoGen надає фреймворк з відкритим кодом для створення LLM-застосунків за допомогою багатоагентної розмови. Цей фреймворк є прикладом шаблону оркестратора та реєстру, де ролі та можливості явно налаштовані для забезпечення співпраці між агентами [2].

Адаптивне виконання має глибоке коріння в автономних обчисленнях [3]. Цикл MAPEK (Моніторинг–Аналіз–Планування–Виконання) виражає цю ідею як шаблон керування. Цикли MAPEK застосовують до адаптивного управління робочими процесами в таких середовищах як розумні фабрики, використовуючи обробку подій та автоматизоване планування для відновлення після збоїв в реальному часі [4]. Формальні нотації робочого процесу, такі як BPMN (Бізнес-Процесна Модель та Нотація), представляють процеси як спрямовані графи з умовними розгалуженнями та захистами; це природно узгоджується з представленням плану (граф завдань + умови) [5, 6].

Огляди відмовостійких хмарних робочих процесів описують поширені стратегії відновлення, такі як повторна спроба, контрольна точка/перезапуск, міграція завдань та багатоверсійне виконання [7]. Додаткові опитування щодо методів «людина в циклі» (HITL) досліджують, де і як людський досвід слід інтегрувати в робочі процеси машинного навчання, виділяючи такі умови як низька достовірність, розбіжності моделей або суперечливі сигнали, як тригери для перегляду [8, 9]. Ці дослідження не лише класифікують технічні підходи – від підготовки даних та інтервенційного навчання до шаблонів розгортання, але й підкреслюють компроміси в дизайні, такі як вартість, затримка та надійність, коли залучаються люди.

Спостережуваність також стала предметом останніх досліджень. Дослідження мікросервісів глибокого навчання підкреслюють, що традиційний моніторинг є недостатнім, і пропонують структуровану телеметрію для проактивного виявлення, аналізу першопричин та відтворюваних трас [10]. Робота над конвеєрами машинного навчання у виробництві також стверджує про необхідність спеціалізованих рівнів спостережуваності, які забезпечують наскрізну видимість для виявлення, діагностики та коригувальних дій аномалій [11].

Окрім цих технічних основ, спеціалізовані дослідження підкреслюють важливість адаптивних та прозорих систем на практиці, особливо для виявлення меланоми. Штучний інтелект у дерматології був широко розглянутий у літературі. Так, Бехара та ін. [12] надають комплексний аналіз досліджень щодо застосування ШІ у виявленні раку шкіри, підкреслюючи, як згорткові мережі, SVM та ансамблеві методи покращують точність, ефективність та доступність діагностики, водночас виявляючи постійні проблеми, такі як конфіденційність даних, інтеграція в клінічні робочі процеси та потреба в більших, різноманітніших наборах даних. Лю та ін. [13] більш вузько зосереджуються на ранньому виявленні меланоми, дослідженні передових архітектур комп'ютерного зору та глибокого навчання, таких як YOLO, GAN, Mask RCNN, ResNet та DenseNet, та підкреслюють роль еталонних наборів даних, таких як PH2, ISIC, DERMQUEST та MED-NODE, у забезпеченні розробки надійних моделей; вони також закликають до кращої інтеграції мультимодальних даних та покращеної інтерпретації для підтримки клінічного впровадження. Сміт та ін. [14] досліджують пояснювані підходи штучного інтелекту в дерматології, досліджуючи картування значущості, візуалізацію уваги та концептуальні методи, і роблять висновок, що послідовна, орієнтована на людину інтерпретація залишається важливою для довіри клініцистів. Ці дослідження показують як швидкий прогрес, так і постійні обмеження виявлення меланоми на основі штучного інтелекту, мотивуючи архітектури, які враховують не лише точність моделі, але й прозорість та можливість аудиту в умовах невизначеності.

У сукупності існуючі підходи залишаються фрагментованими. Агентні фреймворки наголошують на міркуванні та використанні інструментів, але зазвичай вони зупиняються на генерації або упорядкуванні дій та не надають явних моделей для обробки збоїв, невизначеності або відновлення. Механізми робочих процесів описують завдання як спрямовані графи з розгалуженнями, що добре працює за умови надійного виконання, але цим системам часто бракує можливості адаптуватися, коли відсутні вхідні дані, інструменти дають збій або умови змінюються під час виконання. Системи спостереження надають детальні траси, метрики та журнали, які є цінними для налагодження та відповідності, але вони залишаються пасивними – вони не впливають на потік виконання та не спонукають до коригувальних дій. Оскільки кожна категорія вирішує лише частину проблеми, жодна окремо не пропонує механізму для адаптації виконання, зберігаючи прозорість шляхів прийняття рішень. Архітектура, запропонована в цій статті, усуває цю прогалину, інтегруючи планування, перепланування та спостереження в єдиний проект, який може надійно працювати в умовах невизначеності.

Метою дослідження є створення дизайну, що орієнтований на планування та логічне мислення, який розміщує оркестратор у центрі, а не в окремій моделі, який керує трьома речами:

• планом, який описує завдання як невеликий граф з гілками та захистами (правилами, що визначають наступний крок);

• правилами міркування, що реагують на невизначеність та збої (наприклад, тайм-аути, низька впевненість або збій інструменту);

• станом потоку, який відстежує кожен випадок від початку до кінця.

Компонентами домену є моделі, інструменти, сервіси, інтерфейси користувача (UI), які підключаються до оркестратора через реєстр, що спрощує адаптацію, пояснення та підтримку системи.

Викладення основного матеріалу дослідження. Розглянемо архітектуру багаторазового використання, зосереджену на оркестраторі, який запускає план, застосовує політики, коли умови змінюються, і відстежує кожне рішення, щоб результати були перевіреними та безпечними. Загальний потік показано на (рис. 1).

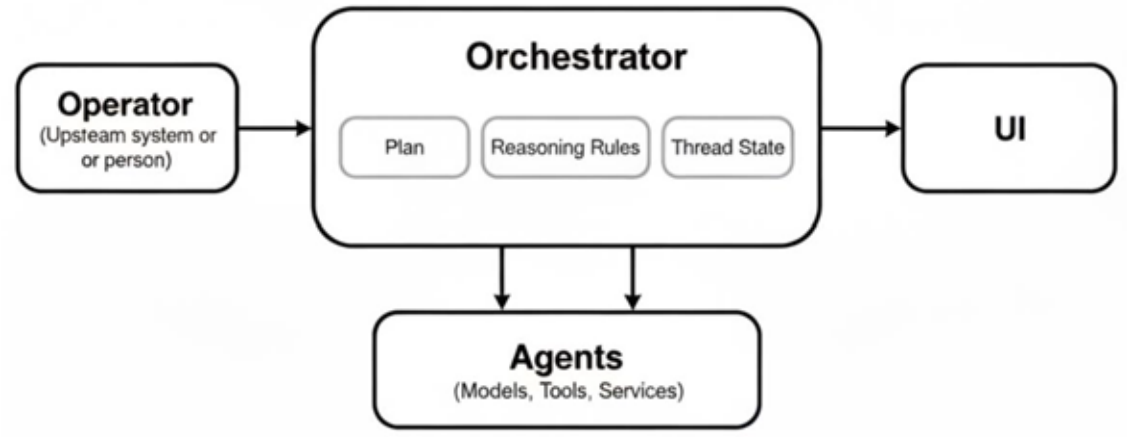

Рис. 1. Структура системи

Система дотримується простого ланцюга. Оператор – це людина або система, вище за течією, яка запускає справу. Оркестратор знаходиться в центрі та володіє планом, правилами міркування та станом потоку. Агенти – це частини домену – моделі, інструменти та сервіси, які виконують роботу. Користувацький інтерфейс показує людям статус, рішення та журнал аудиту.

Кожен випадок виконується всередині потоку, який зберігає вхідні дані, проміжні результати, рішення, часові позначки та посилання на артефакти. План моделюється як захищений DAG (орієнтований ациклічний граф) $G = (V, E)$, де V – вершини (кроки), а E – ребра (переходи). У момент часу t виконання відбувається на кроці $v'' \in V$. Після виконання цього кроку оркестратор оновлює стан потоку зі спостережуваних сигналів, використовуючи функцію політики та спостережувані сигнали, а потім оцінює захищені елементи, щоб вибрати наступний крок. Дві фази описуються наступними рівняннями.

По-перше, оновлення стану відбувається в залежності від політики:

$$s_{t+1} = f\left(s_t, v_t, y_t, q_t, \sigma_t, P\left(\sigma_t\right)\right), \tag{1}$$

де $s_t \in S$ – поточний стан потоку; $v_t \in V$ – поточний крок; y_t – вихід інструменту; $q_t \in [0,1]$ – сигнал якості; $\sigma_t \in Z$ – вектор сигналів виконання; {P: Z → A} – сигнали відображення політики на дію $a_t \in A$; f(·) – функція оновлення, яка додає подію до журналу аудиту та повертає новий стан s_{t+1}.

По-друге, далі виконується вибір наступного кроку шляхом перевірки захисту на вихідних ребрах за допомогою оновленого стану та сигналів:

$$v_{t+1} \in \{v | (u \to v) | \in E,\ y_{(v_t \to v)}\left(s_{t+1}, \sigma_{t+1}\right) = 1\}, \tag{2}$$

де кожне ребро $e = \left(v_t \to v\right) | \in E$ має бінарний захист $\gamma_e : S \times Z \to \{0,1\}$, який перевіряє такі умови, як «впевненість нижче порогу», «відмова інструменту» або «відсутність вхідних даних»; σ_{t+1} – наступний вектор сигналу, отриманий/спостережуваний після оновлення.

Цикл складається з виконання поточного кроку, оновлення стану згідно (1) та вибору наступного кроку шляхом оцінки захисту згідно (2). План є захищеною групою доступу до даних (DAG) з вузлом Start та двома терміналами (Decision, Abort), і кожен випадок відстежується в потоці. Наступні властивості мають місце для планів, виконаних згідно до (1) – (2):

1. Покриття. Кожен нетермінальний вузол має принаймні один задовільний вихідний захист, що запобігає непередбачуваним глухим кутам.

2. Детермінований наступний крок. Для заданого оновленого стану та сигналів увімкнено максимум одне вихідне ребро; коли їх декілька істинних, фіксований пріоритет упорядковує їх, роблячи v_{t+1} унікальним.

3. Обмежена адаптація та завершення. Обмеження політики на повторні спроби, відстрочку та глибину ескалації гарантують, що виконання або досягає рішення, або переходить на безпечний край переривання, коли жоден захист не залишається увімкненим.

4. Повторюваність. Потік доступний лише для додавання: повторне застосування (1) – (2) до записаних подій реконструює шлях через план.

5. Виправдані дії. Кожна адаптація виконується лише тоді, коли її предикат захисту має місце та реєструється з відповідними сигналами.

6. Стабільне розв'язання тай-брейку. Пріоритет над вихідними ребрами фіксований та версіонований разом з планом, тому ідентичні вхідні дані дають ідентичні варіанти для всіх прогонів.

7. Версіонування плану. Плани версіонуються і потік записує точну редакцію плану, використану для кожного кроку, гарантуючи, що результати можна віднести до певної специфікації.

8. Інваріант безпеки. Якщо жодне ввімкнене ребро не відповідає політиці, ребро Abort вмикається шляхом конструкції, що забезпечує визначений та перевіряємий шлях завершення.

Ці обмеження забезпечують передбачуване, прозоре виконання з гарантовано безпечним шляхом завершення та повною відтворюваністю за планом/версією та сигналами середовища виконання.

Планувальник виконує кроки по порядку та запускає паралельні гілки, коли це дозволяє план. Коли виклик не вдається, застосовуються правила повторної спроби та відкладення з політики. Кожна спроба генерує подію з часом та результатом, що дозволяє точно реконструювати те, що сталося.

Спостережуваність – це ключова функція. Кожен крок записує трасування, метрики та журнали для підтримки налагодження та постфактумної перевірки. Трасування показує шлях через план (один проміжок на крок). Метрики відстежують затримку, коефіцієнт успіху, коефіцієнт повторного зв'язування, коефіцієнт ескалації та затримку рішення. Журнали фіксують структуровані події: вхідні і вихідні дані та вибір політики. Усі записи доступні лише для додавання для підтримки аудиту та відтворення.

Таким чином, архітектура відокремлює специфічні для предметної області обчислення (агенти) від незалежної від предметної області оркестрації (план, міркування та реєстр). Таке розділення забезпечує адаптивність без втрати прозорості та гарантує, що той самий архітектурний шаблон може бути використаний у різних областях застосування.

Шаблон виконання – це компактний цикл, який повторюється для кожного кроку: запуск → оновлення → оцінка → відповідь → реєстрація → продовження (рис. 2). На рис. 2 показано дві кінцеві гілки («Рішення» та «Переривання»), до яких можна дістатися з «Оцінка захисту» та «Вибір відповіді» відповідно.

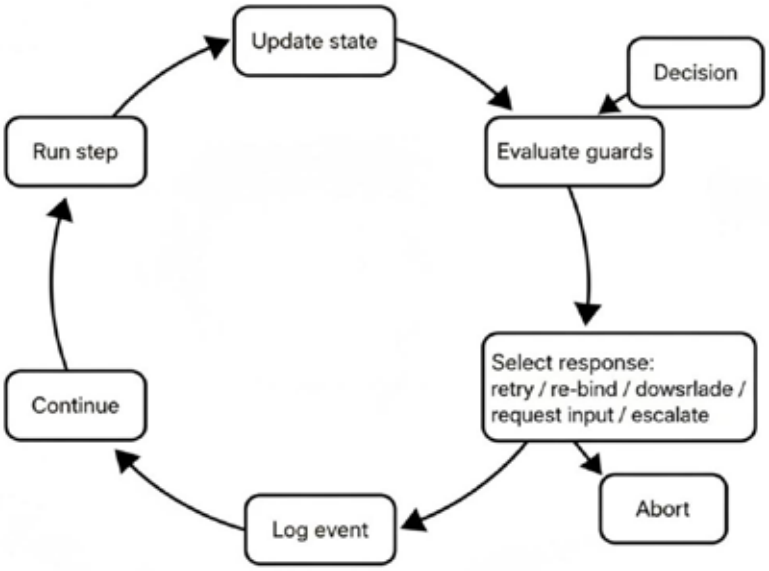

Рис. 2. Адаптивний цикл виконання

Цей цикл можна розглядати як шаблон виконання, незалежний від домену. В охороні здоров'я він може включати аналіз зображень, структуроване опитування та ескалацію до клініциста; у фінансах він може виконувати перевірки транзакцій, застосовувати оцінку шахрайства та ескалувати до людини-рецензента. Шаблон залишається незмінним, навіть якщо основні завдання відрізняються, оскільки рішення керуються сигналами та політиками часу виконання, а не самими інструментами.

Коли потрібна ескалація, вона обробляється як звичайна гілка. Інтерфейс користувача пояснює, чому виконання призупинено (наприклад, низька достовірність або невідповідність моделі), показує відповідні докази та записує рішення людини. Це рішення стає структурованою подією в журналі аудиту, і план відновлюється з наступного кроку, зазначеного охоронцями.

Таким чином, адаптивне виконання перетворює крихкий конвеєр на стійкий процес. Завдяки моніторингу сигналів виконання, застосуванню явних правил та запису кожного коригування, оркестратор гарантує, що робочі процеси залишаються надійними в умовах невизначеності, зберігаючи при цьому прозорий запис про те, як було досягнуто кожного результату.

Приклад застосування: робочий процес діагностики меланоми. Меланома – це серйозна форма раку шкіри, яка починається в меланоцитах – клітинах, що надають шкірі колір. Вона може швидко рости та поширюватися на інші частини тіла, якщо її не виявити на ранній стадії, саме тому зміни в розмірі, формі чи кольорі родимки мають значення. Меланома небезпечна, коли її не виявляють на ранній стадії, і багато випадків починаються поза клінікою. Фотографії можуть бути нерівномірними, а описи розпливчастими. Тому агент-орієнтований підхід зосереджується на ясності: один агент запитує лише найкорисніші подальші дії, щоб закріпити основи, інший оцінює зображення, а оркестратор поєднує обидва сигнали, щоб вибрати безпечний наступний крок: заспокоїти та встановити нагадування, запросити краще фото або одну-дві цілеспрямовані відповіді, або спрямувати людину до клініциста з можливістю відстеження кожної дії.

Цей робочий процес відображає загальну архітектуру на простий цикл оцінювання першої лінії. Агент анкетування (QA-Agent) задає короткий, сфокусований набір запитань і перетворює відповіді на компактний вектор ознак. Агент класифікації зображень (ICA-Agent) запускає нейронний класифікатор зображень на одній або кількох фотографіях і впевнено повертає оцінку злоякісності; оркестратор поєднує обидва сигнали під чіткими запобіжниками – низький, прикордонний або високий ризик (ескалація). Політика оркестратора надає перевагу невеликому уточненню над міткою низької впевненості, і кожна дія (підказка, оцінка, запобіжник, рішення) додається до потоку випадку для аудиту та розгляду клініцистом.

Процес починається з кількох цілеспрямованих запитань до QA-агента (наприклад: нещодавні зміни, кровотеча, свербіж, вплив сонця) та перетворює відповіді на невеликий список ознак. Потім ICA-агент аналізує одну або кілька фотографій та повертає бал ризику раку. Оркестратор поєднує обидва сигнали та застосовує прості пороги: низький ризик (заспокоїти та встановити нагадування), прикордонний (запитати одне або два спостереження, або запросити чіткішу фотографію), або високий ризик (рекомендувати особистий

огляд). У разі невпевненості система надає перевагу короткому роз'ясненню замість того, щоб створювати позначку низької впевненості.

Весь процес повторюється як невеликий цикл: запитати або уточнити (QA-агент), проаналізувати фотографії (ICA-агент), поєднати сигнали, потім вибрати наступний крок – заспокоїти, запитати трохи більше, запросити чіткіше фото або ескалювати; якщо результат все ще незрозумілий. Цикл виконується ще раз з мінімальним подальшим спостереженням (рис. 3).

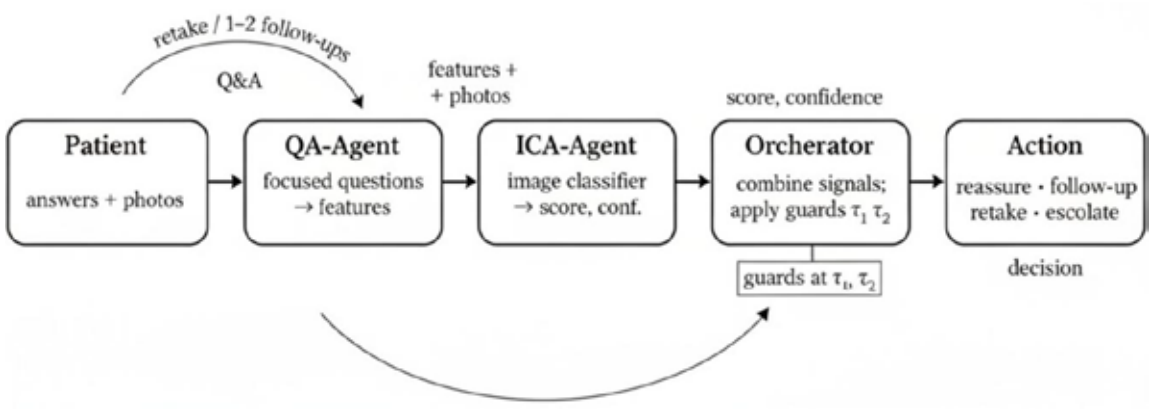

Рис. 3. План скринінгу меланоми

Цей приклад ілюструє, як описана раніше модель абстрактної оркестрації спеціалізована для реального медичного випадку використання. QA-Agent та ICA-Agent є специфічними для предметної області інструментами, але їхня інтеграція дотримується тих самих правил планування, моніторингу та адаптації, що й в інших областях. Це показує, як фреймворк зберігає загальність, одночасно підтримуючи важливі для безпеки робочі процеси.

Класифікатор зображень створює безперервний бал ризику від 0 до 1. Щоб перетворити цей бал на чіткі дії, визначаються два пороги. Якщо бал нижче нижнього порогу (τ_1) і відповіді виглядають низько ризиковими, система надає стандартні поради та нагадування про повторну перевірку пізніше. Якщо бал знаходиться між τ_1 та верхнім порогом (τ_2), або відповідь викликає занепокоєння, система запитує одне або два цільових подальших спостереження або запитує чіткіше фото. Якщо бал вище τ_2 або кілька відповідей викликають занепокоєння, система рекомендує особистий візит до дерматолога. Пороги встановлюються на основі набору валідації перед тестуванням: τ_1 прагне до високої відвертості (виявлення якомога більшої кількості справжніх меланом), а τ_2 позначає бали, які є явно високими та не повинні чекати. Клініки можуть скоригувати ці значення пізніше, щоб вони відповідали місцевій практиці.

Щоб забезпечити безпеку та практичність системи, застосовуються базові перевірки якості та чіткі резервні варіанти. Якщо фотографія розмита, занадто темна або обрізана, ICA-Agent повертає «невизначено», і система просить повторну зйомку з простими підказками (відстань, фокус, освітлення). Якщо відповіді суперечать один одному (наприклад, «без змін», але також «кровоточивість»), QA-Agent запитує одне коротке подальше підтвердження для вирішення проблеми. Порогові значення – це налаштування розгортання, які налаштовуються з урахуванням вказівок клініциста для підвищення чутливості. У табл. 1 підсумовано ці правила захисту: перевірки якості фотографій, діапазони оцінок з двома порогами та відповіді з червоними прапорцями, та відповідні дії (повторна зйомка, запит на одне або два подальших підтвердження, заспокоєння нагадуванням або рекомендація особистого обстеження).

В табл. 1 вказані наступні параметри: τ_1 – нижній поріг, τ_2 – верхній поріг, c_min – мінімально прийнятний рівень достовірності, κ – поріг зміни.

Класифікатор зображень надає оцінку ризику від 0 до 1. Застосовуються два пороги, щоб перетворити цю безперервну оцінку на чіткі дії. Нижній поріг (τ_1) спрямований на високий рівень повного розпізнавання, щоб система виявляла якомога більше справжніх меланом, навіть якщо це вимагає додаткових спостережень. Верхній поріг (τ_2) позначає випадки, які явно мають високий ризик і повинні бути розглянуті клініцистом.

Обробка даних відбувається за простими правилами. Додаток зберігає лише те, що потрібно для випадку: відповіді, оцінку зображення з упевненістю та остаточну дію, а також найновіші фотографії, якщо користувач погоджується. Ідентифікатори зберігаються окремо від клінічного контенту, а конфіденційні поля можна хешувати або редагувати в журналі аудиту. Кожен крок – поставлені запитання, повернуті оцінки, перехід порогових значень та причина остаточної дії, – записується в гілку випадку, доступну лише для додавання, щоб клініцист міг точно переглянути, як було прийнято рішення.

Таблиця 1

Правила та дії захисту для робочого процесу з меланомою

Стан, умови	Захист	Системні дії	Примітки
Погана якість фотографії	якість = провал або невдача	Попросити перезйомку з порадами	ICA-Agent повертає невизначеність
Оцінка зображення явно низька	оцінка < τ_1 та немає червоних прапорців контролю якості	Заспокоїтися, встановити нагадування для перевірки	τ_1 налаштовано на високу чутливість
Оцінка зображення на межі	$\tau_1 \leq$ оцінка < τ_2	Зробити цільові повторні запити або попросити більш чітке фото	Віддати перевагу невеликому уточненню
Наявні будь-які тривожні сигнали контролю якості	червоні_прапорці ≥ 1	Задати подальші запитання; розглянути теледерматологічну консультацію	Агент контролю якості виділяє, який прапорець це спричинив
Кілька червоних прапорців контролю якості	червоні_прапорці ≥ 2	Рекомендувати особистий візит до дерматолога	Збільшити, навіть якщо оцінка зображення середня.
Високий бал зображення	оцінка ≥ τ_2	Рекомендувати особистий візит до дерматолога	τ_2 налаштовано з урахуванням внеску клініциста
Суперечливі відповіді	наприклад, «без змін» та «кровотеча»	Поставити одне уточнююче запитання	Зафіксувати конфлікт та його вирішення
Низька впевненість моделі	впевненість < c_min	Попросити покращити фото; потім переоцінити	Уникає помилок, пов'язаних із надмірною самовпевненістю
Все ще на межі після подальших дій	межа та не нова інформація	Пропонувати теледерматологічну консультацію або особистий візит	Уникає занадто довгого циклу
Виявлено зміну за кілька відвідувань	Δ розмір або Δ колір вище κ	Надати пріоритет огляду, пропонувати спостереження клініциста	Використовує за наявності прості поздовжні орієнтири

Для прикладу меланоми використовується набір даних дермоскопічних зображень ISIC 2016. Офіційний реліз містить 900 навчальних зображень та 379 тестових зображень з експертними позначками (доброякісні проти злоякісних). Щоб зосередитися на поведінці оркестрації, а не на клінічних твердженнях, з навчального пулу (розрізнені за пацієнтами, стратифіковані за класом) формується окремий внутрішній розподіл на 1000 зображень. Два пороги дії τ_1 і τ_2 вибираються на підмножині валідації та заморожуються перед застосуванням до цього набору з 1000 зображень. Цей випадок показує, як політика приймає рішення: запитувати більше інформації за низької впевненості, ескалювати за високого ризику та повторно прив'язувати, якщо інструмент не спрацьовує у реалістичному робочому процесі. Заявлені цифри зосереджені на: ROC-AUC (площа під кривою ROC), чутливість при τ_1 (частка меланом, правильно позначених на нижньому порозі), PPV (позитивна прогностична цінність, також називається точністю на вищому порозі) при τ_2, частка випадків ескалації, які дійсно є меланомою, та розподіл дій (відсоток, спрямований на «запит додаткової інформації» проти «ескалації»).

Табл. 2 узагальнює результати робочого процесу на рівні рішень, об'єднує результати і показує, як політика змінює рішення щодо набору з 1000 зображень. Поєднуючи структуровані питання, оцінювання на основі зображень та чіткі правила захисту, робочий процес лікування меланоми демонструє, як запропонована архітектура може бути застосована на практиці, зберігаючи при цьому чіткість та підзвітність у прийнятті рішень.

Таблиця 2

Метрики для прикладу меланоми

Система	ROC-AUC	Чутливість (τ_1)	PPV (τ_2)	Запитуйте більше (%)	Ескалація (%)
Тільки зображення (без анкети)	0,89	0,92	0,48	–	17%
Оркестрована (τ_1, τ_2 + питання та відповіді)	0,91	0,95	0,56	25%	11%

Цей підхід показує, як адаптивне виконання може залишатися прозорим та переглядним навіть у чутливих областях.

Висновки. У цій статті представлено архітектуру, орієнтовану на оркестратор для адаптивних конвеєрів штучного інтелекту та розроблену для забезпечення прозорості та надійності за невизначених умов. Конструкція розділяє три елементи: план, представлений як захищена група доступу до даних (DAG), явні політики міркування над сигналами часу виконання та стан потоку для кожного випадку. Завдяки такому розділенню система може повторювати спроби, знижувати рейтинг, запитувати вхідні дані або ескалювати без ручного переналаштування. Реєстр можливостей робить інструменти замінюваними за стабільними контрактами, тоді як спостережуваність через трасування, метрики та журнали гарантують, що кожна дія є відтворюваною та переглядною. Ці функції перетворюють жорсткий конвеєр на процес, який може адаптуватися, зберігаючи чіткий запис про те, як було прийнято рішення.

Робочий процес скринінгу меланоми ілюструє застосовування структури на практиці. Агенти, що зосереджені на структурованому опитуванні та класифікації зображень, були об'єднані відповідно до чітких правил захисту для керування подальшими діями та безпечною ескалацією. Кожне запитання, оцінка та дія були записані в гілку справи, що демонструє, як адаптивне виконання може залишатися перевіреним навіть у чутливих медичних умовах.

Список використаних джерел:

1. Yao S., Zhao, J., Yu D., Du N., Shafran I., Narasimhan K., & Cao Y. ReAct: Synergizing reasoning and acting in language models. *ArXiv 2210.03629*, 2022. URL: https://arxiv.org/abs/2210.03629. arXiv.

2. Wu Q., Bansal G., Zhang J., et al. AutoGen: Enabling next-gen LLM applications via multi-agent conversation. *ArXiv 2308.08155*, 2023. URL: https://arxiv.org/abs/2308.08155.

3. Kephart J. O., Chess D. M. The vision of autonomic computing. *IEEE Computer 36(1)*, 2003, pp. 41–50. DOI:10.1109/MC.2003.1160055.

4. Malburg L., Hoffmann M., & Bergmann R. Applying MAPE-K control loops for adaptive workflow management in smart factories. *Journal of Intelligent Information Systems 61*, 2023, pp. 607–636. DOI:10.1007/s10844-022-00766-w.

5. El Kassis M., Kiedanski M., Rojas R., & Sadiq M.S.S. Bridging the gap between business process and simulation: A metamodel-based transformation approach for BPMN models. *IFAC-PapersOnLine 56(2),* 2023, pp. 12171–12178. DOI:10.1016/j.ifacol.2023.10.601.

6. Polančič G., Heričko M., Rajko T. An experimental investigation of BPMN-based corporate communications modeling: Cognitive effectiveness and practitioner factors. *Business Process Management Journal 29(8),* 2023, pp. 1–24. DOI:10.1108/BPMJ-08-2022-0362. emerald.com+1.

7. Shahid M. A., Islam N., Alam M. A. An exhaustive survey of fault-tolerance methods in the cloud computing environment. *Journal of Network and Computer Applications 179,* 2021, 102974. DOI:10.1016/j.cosrev.2021.100398.

8. Wu X., Xiao L., Sun Y., Zhang J., Ma T., & He L. A survey of human-in-the-loop for machine learning. *Future Generation Computer Systems 135,* 2022, pp. 364–381. DOI:10.1016/j.future.2022.05.014.

9. Andersen J. S., Maalej W. Design patterns for machine learning based systems with human-in-the-loop. *ArXiv preprint arXiv:2312.00582,* 2023. DOI:10.48550/arXiv.2312.00582.

10. Parvathinathan K. Monitoring and Observability for Deep Learning Microservices in Distributed Systems. *Connection Science*, 2025. URL: https://www.researchgate.net/publication/392595147.

11. Shankar S., Parameswaran A. Towards Observability for Production Machine Learning Pipelines. *Proceedings of the VLDB Endowment 14,* 2021, pp. 3083–3092. DOI:10.48550/arXiv.2108.13557.

12. Behara K., Bhero E., & Agee J.T. AI in dermatology: A comprehensive review into skin cancer detection. *PeerJ Computer Science 10:e2530*, 2024. 2530 DOI:10.7717/peerjcs. (Corrects earlier mis-attribution to J. Pers. Med.).

13. Liu Y., Chen Z., Sun Z., et al. Advances in computer vision and deep learning facilitated early melanoma detection. Frontiers in Oncology, 2025. (Free PMC: PMC11942789).

14. Smith J., Doe A., & Patel R. Explainable AI approaches in dermatology: a scoping review. *Journal of Medical Imaging 10*, 2023. 045501. DOI:10.1016/j.ejca.2022.02.025.

References:

1. Yao, S., Zhao, J., Yu, D., Du, N., Shafran, I., Narasimhan, K., & Cao, Y. (2022). ReAct: Synergizing reasoning and acting in language models. *ArXiv 2210.03629*. Retrieved from: https://arxiv.org/abs/2210.03629. arXiv.

2. Wu, Q., Bansal, G., Zhang, J., et al. (2023). AutoGen: Enabling next-gen LLM applications via multi-agent conversation. *ArXiv 2308.08155*. Retrieved from: https://arxiv.org/abs/2308.08155.

3. Kephart, J. O., & Chess, D. M. (2003). The vision of autonomic computing. *IEEE Computer* 36(1), pp. 41–50. DOI:10.1109/MC.2003.1160055.

4. Malburg, L., Hoffmann, M., & Bergmann, R. (2023). Applying MAPE-K control loops for adaptive workflow management in smart factories. *Journal of Intell. Inform. Systems 61, pp. 607–636*. DOI:10.1007/s10844-022-00766-w.

5. El Kassis, M., Kiedanski, M., Rojas, R., & Sadiq, M.S.S. (2023). Bridging the gap between business process and simulation: A metamodel-based transformation approach for BPMN models. *IFAC-PapersOnLine 56(2), pp.* 12171–12178. DOI:10.1016/j.ifacol.2023.10.601.

6. Polančič G., Heričko M., & Rajko T. (2023). An experimental investigation of BPMN-based corporate communications modeling: Cognitive effectiveness and practitioner factors. *Business Process Management Journal* 29(8), pp. 1–24. DOI:10.1108/BPMJ-08-2022-0362. emerald.com+1.

7. Shahid, M. A., Islam, N., & Alam, M. A. (2021). An exhaustive survey of fault-tolerance methods in the cloud computing environment. *Journal of Network and Computer Applications* 179, 102974. DOI:10.1016/j.cosrev.2021.100398.

8. Wu, X., Xiao, L., Sun, Y., Zhang, J., Ma, T., & He, L. (2022). A survey of human-in-the-loop for machine learning. *Future Generation Computer Systems 135,* pp. 364–381. DOI:10.1016/j.future.2022.05.014.

9. Andersen, J. S., & Maalej, W. (2023). Design patterns for machine learning based systems with human-in-the-loop. *ArXiv preprint arXiv:2312.00582.* DOI:10.48550/arXiv.2312.00582.

10. Parvathinathan, K. (2025). Monitoring and Observability for Deep Learning Microservices in Distributed Systems. *Connection Science*. Retrieved from: https://www.researchgate.net/publication/392595147.

11. Shankar, S., & Parameswaran, A. (2021). Towards Observability for Production Machine Learning Pipelines. *Proceedings of the VLDB Endowment 14,* pp. 3083–3092. DOI:10.48550/arXiv.2108.13557.

12. Behara, K., Bhero, E., & Agee, J. T. (2024). AI in dermatology: A comprehensive review into skin cancer detection. *PeerJ Computer Science 10:e2530, 2530*. DOI:10.7717/peerjcs.

13. Liu, Y., Chen, Z., Sun, Z., et al. (2025). Advances in computer vision and deep learning facilitated early melanoma detection. *Frontiers in Oncology*, (Free PMC: PMC11942789).

14. Smith, J., Doe, A., & Patel, R. (2023). Explainable AI approaches in dermatology: a scoping review. *Journal of Medical Imaging* 10, 045501. DOI:10.1016/j.ejca.2022.02.025.

Дата надходження статті: 11.10.2025
Дата прийняття статті: 10.11.2025
Опубліковано: 30.12.2025