ПРАКТИЧНІ ПРОБЛЕМИ ОРКЕСТРАЦІЇ МУЛЬТИАГЕНТНИХ СИСТЕМ У ЗАДАЧАХ АВТОМАТИЗАЦІЇ БІЗНЕС-ПРОЦЕСІВ КОРИСТУВАЧІВ В РАМКАХ SaaS-ПРОДУКТУ

Рєзник М.О., Чумаченко С.В.
ТОВ «Праймкор», Харків, Україна
Харківський національний університет радіоелектроніки, Харків, Україна

Автоматизація бізнес-процесів у сучасних SaaS-рішеннях на основі штучного інтелекту переходить від виконання окремих детермінованих операцій до підтримки багатокрокових сценаріїв, у межах яких система повинна аналізувати контекст, обирати послідовність дій, розподіляти ролі між спеціалізованими компонентами та адаптувати логіку виконання до змінних умов. У таких сценаріях мультиагентні системи є доцільними, оскільки дають змогу розподіляти функції планування, аналізу, валідації, маршрутизації та інтеграції між окремими агентами. Практична ефективність такого підходу визначається не стільки можливостями окремого агента, скільки якістю оркестрації всієї системи. Для множини агентів критичними стають сумісність ролей, повторюваність результатів, своєчасне оновлення логіки та керованість змін [1, 2].

У межах SaaS-продукту оркестрація має виконувати функцію окремого керівного шару, який задає послідовність дій, правила передавання контексту, умови переходів між станами, механізми блокування конфліктних операцій, ресурсні обмеження, порядок залучення людини до процесу та завершення сценарію. За відсутності такого шару мультиагентна архітектура породжує координаційні витрати: дублювання задач, конфлікти станів, втрату бізнес-контексту, непрозорість маршруту та неконтрольоване споживання ресурсів. Практичні приклади мультиагентних систем показують, що атомарне призначення задач, ієрархія цілей і контроль бюджету є не допоміжними опціями, а базовими умовами стабільної роботи [2, 3].

Однією з базових проблем є вибір моделі оркестрації. Динамічна оркестрація через великі мовні моделі забезпечує високу адаптивність, оскільки план формується безпосередньо під час виконання. Проте слабкість цього підходу полягає в тому, що якість усього маршруту залежить від формулювання запиту, а будь-яка зміна вхідних умов може призвести до появи нового плану або іншого формату результату. Така модель ускладнює аудит, тестування та пояснення того, чому було обрано саме цей сценарій [1, 3]. Окрему небезпеку становлять галюцинації та каскадне поширення семантичної помилки, коли хибний висновок одного агента стає основою для коректно виконаних, але змістовно неправильних наступних кроків [1, 2].

Детермінована оркестрація, навпаки, спирається на формально задані переходи, умови та контрольні точки. Вона краще підтримує стабільність, регресійне тестування, трасування та відповідність регламентованим бізнес-процесам. Її обмеженням є низька гнучкість: зміна маршруту, виняткової умови або інтеграції потребує перегляду процесної логіки [1, 3]. Тому повністю жорстка схема також не є універсальним рішенням для SaaS-продуктів, де бізнес-процеси

користувачів відрізняються за структурою та правилами. Найбільш доцільною є гібридна оркестрація, у якій BPMN або подібний процесний рушій задає формальний каркас процесу, а агенти виконують інтелектуальні функції всередині контрольованого сценарію [1–3]. Саме такий підхід дозволяє перевіряти рішення агентів у контрольних точках, підтримувати потоки повідомлень і реалізовувати участь людини в контурі керування, а також підтримку довготривалих процесів [2, 3].

Практичні проблеми оркестрації в SaaS-продуктах концентруються в кількох взаємопов'язаних вузлах.

1. Керованість міжагентної взаємодії. Зі збільшенням кількості агентів ускладнюється маршрутизація задач, зростає кількість службових повідомлень і підвищується ризик беззмістовного дроблення роботи. Якщо глобальна ціль сформульована нечітко, система починає генерувати надмірну кількість підзадач і імітувати активність без пропорційного зростання корисного результату. Отже, оркестрація повинна формалізувати дерево цілей, ролі агентів та межі їхньої самостійності [1, 3].

2. Дублювання та конкурентне виконання задач. У мультиагентному середовищі кілька агентів можуть одночасно взяти в роботу одну й ту саму сутність, що збільшує витрати на обчислення та спотворює стан процесу. Практичним рішенням є атомарне призначення задач, блокування конкурентного доступу та фіксація статусів виконання [2, 3].

3. Ізоляція агентів. Розміщення всіх агентів у межах одного сервісу спрощує обмін контекстом, але зменшує контроль і збільшує радіус ураження у разі ін'єкції вхідних інструкцій або локальної помилки. Саме тому доцільною є ізоляція агентів у відокремлених середовищах, зокрема в контейнерах, із розділенням контекстів, ресурсних квот, журналів і метрик [1, 3].

4. Трасованість рішень. Бізнес-процес вимагає можливості відновити, який агент ініціював дію, на основі яких даних вона була виконана, який маршрут проходження мав запит і скільки було повторних спроб. Практика показує, що для цього недостатньо довільного обміну повідомленнями; потрібен структурований протокол взаємодії з кореляційними ідентифікаторами, часовими обмеженнями, параметрами життєвого циклу контексту, ключами ідемпотентності та наскрізним моніторингом. У мікросервісному SaaS-середовищі часто доцільніше будувати власний протокол поверх наявної шини повідомлень, ніж покладатися на універсальні схеми, оскільки це дає змогу точніше налаштувати маршрутизацію, корисне навантаження та аудит [1, 3].

5. Ресурсний контроль. Мультиагентна система здатна швидко збільшувати кількість викликів моделей, повторних спроб і допоміжних кроків. Якщо агент потрапляє в цикл помилок або безрезультатних уточнень, бюджет може бути вичерпано за короткий час. Тому бюджетні обмеження, механізми автоматичної паузи, періодичного контролю стану та політики зупинки неефективних сценаріїв є частиною оркестраційної логіки, а не лише інфраструктурним засобом [2], [3].

6. Підтримка довготривалих процесів і людського втручання. Частина бізнес-процесів включає очікування відповіді користувача, зовнішніх інтеграцій,

погоджень, тайм-аутів і відкладеного відновлення. У таких випадках процесний рушій повинен переводити схему в режим очікування, зберігати контекст і коректно відновлювати сценарій після події чи підтвердження. Аналогічно, у критичних точках, за низької впевненості моделі або за юридично чи фінансово значущих дій, система має передавати керування людині без руйнування процесної логіки [2, 3].

Окрему прикладну проблему становить розрізненість навіть типових бізнес-процесів у різних компаніях. Однакові за назвою процеси - онбординг клієнта, погодження документа, обробка звернення або маршрутизація заявки - у різних організаціях відрізняються складом етапів, послідовністю переходів, правилами ескалації, ролями учасників, винятками та зовнішніми інтеграціями. Це унеможливлює використання однієї жорсткої універсальної схеми автоматизації в межах SaaS-продукту.

Відповідно, оркестраційний шар повинен підтримувати налаштування під специфічний процесний конвеєр конкретної компанії без повного перепроєктування системи.

При цьому така гнучкість має поєднуватися з формальними обмеженнями, валідацією конфігурацій і контрольованими межами кастомізації, інакше налаштовуваність призведе до втрати цілісності процесу. Це додатково підсилює доцільність гібридної моделі, у якій змінними є параметри та окремі елементи процесу, а не самі базові принципи його керування [1, 3].

Отже, практичні проблеми оркестрації мультиагентних систем у SaaS-продуктах зосереджені навколо таких вузлів: керованість міжагентної взаємодії, дублювання і конкурентне виконання задач, ізоляція агентів, трасованість рішень, ресурсний контроль, підтримка довготривалих процесів і варіативність бізнес-процесів між компаніями. Найбільш доцільною архітектурною відповіддю на ці виклики є гібридна оркестрація, у якій формалізований процесний каркас, контрольні точки та правила безпеки поєднуються з агентною адаптивністю на рівні окремих інтелектуальних задач [1–3].

Саме така модель створює підґрунтя для побудови SaaS-продуктів, здатних автоматизувати специфічні користувацькі процесні конвеєри без втрати керованості, відтворюваності та масштабованості [1, 3].

Список літератури

1. Li X., Wang S., Zeng S., Wu Y., Yang Y. A survey on LLM-based multi-agent systems: workflow, infrastructure, and challenges // Vicinagearth. 2024. Vol. 1, No. 1. DOI: 10.1007/s44336-024-00009-2.

2. 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. AutoGen: Enabling next-gen LLM applications via multi-agent conversation // arXiv preprint. 2023. arXiv:2308.08155. DOI: 10.48550/arXiv.2308.08155.

3. Tran K.-T., Dao D., Nguyen M.-D., Pham Q.-V., O'Sullivan B., Nguyen H. D. Multi-agent collaboration mechanisms: A survey of LLMs // CoRR. 2025. abs/2501.06322. DOI: 10.48550/arXiv.2501.06322.