

АВТОМАТИЗАЦІЯ РОЗПОДІЛУ ЗАДАЧ МІЖ ВИКОНАВЦЯМИ ЗА ДОПОМОГОЮ ШІ ТА ВЕЛИКИХ МОВНИХ МОДЕЛЕЙ

¹ Вінницький національний технічний університет

Анотація

У роботі представлено систему автоматизованого розподілу задач між виконавцями з використанням рішень на основі штучного інтелекту (ШІ) та великих мовних моделей (ВММ), зокрема моделі GPT-4o mini. Розроблена система інтегрована в платформу Salesforce із застосуванням інструменту Agentforce, що забезпечує виконання ВММ-запитів (до OpenAI) під час виконання транзакції зі створення запису-задачі. Розроблене рішення враховує навички та завантаженість потенційних виконавців, а також зміст опису задачі при прийнятті рішення про призначення. Для підвищення продуктивності реалізовано патерн мемоізації у модулі вибору виконавця, що зменшує кількість повторних обчислень. Рішення протестовано на синтетичному наборі даних, що імітує різні сценарії розподілу задач, і продемонструвало коректність функціонування згідно з очікуваною логікою. Запропонована система є domain-agnostic, тобто не залежить від специфіки предметної області і може бути застосована в різних галузях.

Ключові слова: розподіл задач, штучний інтелект, великі мовні моделі, автоматизація.

Abstract

This paper presents a system for automated task allocation among performers using AI-based solutions and large language models (LLMs), specifically the GPT-4o mini model. The developed system is integrated into the Salesforce platform using the Agentforce tool, which enables LLM queries (to OpenAI) via Apex triggers. The assignment decision considers the skills and current workload of potential performers, as well as the task description content. A memoization pattern was implemented in the executor selection module to reduce repeated computations and improve efficiency. The solution was tested on a synthetic dataset simulating various task distribution scenarios and demonstrated correct functionality in line with expected logic. The proposed system is domain-agnostic, meaning it is not tied to a specific field and can be applied across different industries.

Keywords: task distribution, artificial intelligence, large language models, automation.

Вступ

Розподіл задач між виконавцями є одним із ключових процесів у сучасних організаціях, від якого залежать продуктивність і ефективність роботи команд [1]. Дослідження показують, що спосіб призначення завдань суттєво впливає на продуктивність працівників [2].

Як правило, кожна задача має певний контекст, який включає інформацію про постановника та опис самої задачі. Контекст може доповнюватися додатковими даними (як от аудіо- чи відео-записами, електронними листами, повідомленнями в месенджерах, тощо). У той самий час, потенційні виконавці володіють певним (та часто різним) набором навичок і компетенцій, завантаженням, тощо [3]. Таким чином, для якісного вирішення задачі важливо правильно співставити вимоги задачі з уміннями доступних фахівців.

Наразі проблема розподілу задач вирішується кількома підходами:

1. Експертною оцінкою [4];
2. Автоматичними класифікаторами [5];
3. Математичними алгоритмами [6, 7].

У контексті зростаючої складності та динамічного розвитку бізнес-процесів традиційні методи розподілу задач часто виявляються недостатньо гнучкими або дорогими у використанні. Це зумовлює актуальність пошуку нових підходів до автоматизації призначення задач.

Одним із перспективних напрямків є використання сучасних засобів ШІ, зокрема великих мовних моделей. ВММ здатні виконувати глибокий контекстуальний аналіз опису задачі та даних про вико-

навців, що потенційно дозволяє автоматично призначати задачі з точністю на рівні експертних оцінок [8]. Крім того, рішення на основі ВММ можна відносно легко адаптувати до змін у бізнес-процесах і вони потребують мінімального ручного налаштування під нові умови, що вигідно відрізняє їх від жорстко запрограмованих правил.

У цій доповіді розглянуто побудову такої системи автоматизації розподілу задач із використанням моделі GPT-4o mini та проаналізовано результати її тестування.

Результати дослідження

Розроблена система побудована на базі хмарної платформи Salesforce з використанням нещодавно представленого фреймворку Agentforce. Це рішення дозволяє легко інтегрувати можливості ШІ-агентів у бізнес-процеси Salesforce і викликати великі мовні моделі безпосередньо з Apex-коду. Зокрема, Agentforce надає доступ до LLM через єдиний API, і за замовчуванням усі запити агента обробляються моделлю OpenAI GPT-4 Omni [9].

Архітектурно рішення складається з Apex-тригера, що спрацьовує при створенні або оновленні запису задачі, та викликає дію Агента з вказаним шаблоном (див. рисунок 1).

```
9 public static CaseTriggerHandler getInstance() {
10     if (null == instance) {
11         instance = new CaseTriggerHandler();
12     }
13     return instance;
14 }
15
16 public void run(List<Case> newCases, Map<Id, Case> oldCasesMap) {
17     if (Trigger.isBefore && Trigger.isInsert) {
18         handleBeforeInsert(newCases);
19     }
20 }
21
22 @TestVisible
23 private void handleBeforeInsert(List<Case> newCases) {
24     performSkillBasedAssignment(newCases);
25 }
26
27 @TestVisible
28 private void performSkillBasedAssignment(List<Case> newCases) {
29     for (Case newCase : newCases) {
30         if (String.isBlank(newCase.Description)) {
31             continue;
32         }
33         try {
34             String promptResponse = AgentforceService.getInstance().invokeCaseSkillBasedAssignmentPromptTemplate(newCase.Description);
35             if (String.isBlank(promptResponse)) {
36                 continue;
37             }
38             Map<String, Object> promptResponseMap = (Map<String, Object>) JSON.deserializeUntyped(promptResponse);
39             if (!promptResponseMap.containsKey(PRM_PROP_ASSIGN_TO_ID)) {
40                 continue;
41             }
42             String userId = (String) promptResponseMap.get(PRM_PROP_ASSIGN_TO_ID);
43             if (!String.isBlank(userId)) {
44                 newCase.OwnerId = userId;
45             }
46         } catch (Exception e) {
47             newCase.addError(e.getMessage());
48         }
49     }
50 }
51 }
```

Рисунок 1 – Тригер-хендлер, що викликає дію Агента з вказаним шаблоном

Агент формує запит до ВММ, який містить опис задачі та релевантну інформацію про виконавців (їхні навички, роль, поточну зайнятість тощо). Модель GPT-4o mini генерує рекомендацію щодо оптимального виконавця, яка потім використовується системою для автоматичного призначення задачі відповідній особі.

При реалізації логіки розподілу особливу увагу приділено продуктивності та масштабованості рішення. Використання великої мовної моделі потребує значних обчислювальних ресурсів і може супроводжуватися затримками при великому обсязі запитів.

Щоб знизити навантаження на систему і мінімізувати кількість звернень до зовнішнього API, було застосовано патерн мемоізації у модулі вибору виконавця [10]. Такий підхід суттєво підвищує швидкодію розподілу задач та зменшує використання транзакційних лімітів (див. рисунок 2).

```
14     @TestVisible
15     private List<User> availableCSMembersMemo;
16
17     @TestVisible
18     private static UserSelector instance;
19
20 > public static UserSelector getInstance() { ...
25     }
26
27     public List<User> getAvailableCSMembers() {
28         if (null != availableCSMembersMemo) {
29             return availableCSMembersMemo;
30         }
31         List<User> allCSMembers = [
32             SELECT Id, Name, Email
33             FROM User
34             WHERE IsActive = TRUE
35             AND UserRole.Name LIKE :USER_ROLE_NAME_MATCHER_CUSTOMER_SUPPORT
36         ];
```

Рисунок 2 – Реалізація патерну мемоізації у селекторі

Функціональність системи перевірено шляхом експериментального тестування на синтетичному датасеті. Було згенеровано множину тестових задач із різними описами (різні категорії проблем, рівні складності тощо) та групу умовних виконавців із різними наборами навичок та рівнями завантаженості. Для кожного тестового прикладу визначався очікуваний «еталонний» виконавець на основі заздалегідь заданих правил відповідності навичок. Після тестування системи, результати автоматичного розподілу порівнювалися з еталонними значеннями. Результати співставлення підтвердили, що модель успішно визначає ключові вимоги із тексту опису задачі і знаходить виконавця, у що має необхідні знання та навички і є доступним.

Висновки

Розроблене рішення є підтвердженням того, що використання ШІ та BMM для вирішення прикладної задачі з розподілу завдання між виконавцями є можливим та доцільним.

Інтеграція BMM у бізнес-процес призначення виконавців дозволяє суттєво підвищити гнучкість та точність розподілу задач у порівнянні з традиційними методами. За рахунок глибокого аналізу контексту завдання та навичок співробітників, система забезпечує об'єктивне і обґрунтоване призначення, мінімізуючи вплив людського фактору та ймовірність помилкового вибору. Навіть у разі зміни вимог або появи нових типів задач немає потреби адаптувати алгоритми – досить надати моделі оновлений опис, і вона здатна адаптивно підібрати рішення.

Водночас впровадження такого AI-агента потребує відповідних початкових витрат та врахування вартості LLM-запитів, однак очікуване підвищення ефективності та автоматизація процесу можуть швидко окупити ці витрати..

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. It does not matter how hard you work: The importance of task allocation for worker productivity [Електронний ресурс] – Режим доступу: <https://www.sciencedirect.com/science/article/pii/S0165176523001404>
2. Sales Call Recordings: The Secret of High-Performing Sales Teams [Електронний ресурс] – Режим доступу: <https://www.salesforce.com/blog/sales-call-recording-strategies/>
3. Every Person Has Unique Talent, Ability, Gifts, and Skills [Електронний ресурс] – Режим доступу: <https://medium.com/koinonia/every-person-has-unique-talent-ability-gifts-and-skills-c0cff5de0731>
4. How to use expert judgment in project management [Електронний ресурс] – Режим доступу: <https://asana.com/resources/expert-judgment>

5. Automate Case Triage with Einstein Case Classification for Flows (Generally Available) [Електронний ресурс] – Режим доступу: https://help.salesforce.com/s/articleView?id=release-notes.rn_einstein_case_classification_flows.htm&release=238&type=5

6. Алгебри алгоритмів для моделювання розподілу ресурсів в ІТ проектах [Електронний ресурс] – Режим доступу: <https://science.lpnu.ua/uk/sisn/vsi-vypusky/vypusk-13-2023/algebry-algorytmiv-dlya-modelyuvannya-rozpodilu-resursiv-v-it>

7. Mathematical Model and Algorithm for the Task Allocation Problem of Robots in the Smart Warehouse [Електронний ресурс] – Режим доступу: <https://www.scirp.org/journal/paperinformation?paperid=60955>

8. Automating Task Assignments Using AI for Maximum Efficiency [Електронний ресурс] – Режим доступу: <https://medium.com/@jesse.henson/automating-task-assignments-using-ai-for-maximum-efficiency-9ed76b0660e1>

9. Supported Models – Agentforce Developer Guide [Електронний ресурс] – Режим доступу: <https://developer.salesforce.com/docs/einstein/genai/guide/supported-models.html>

10. Functional Programming: Understanding Memoization. [Електронний ресурс] – Режим доступу: <https://medium.com/swlh/functional-programming-memoization-c2af275a0b1d>

Слободян Роман Віталійович — аспірант кафедри Автоматизації та інтелектуальних інформаційних технологій, Вінницький Національний Технічний Університет, Вінниця, e-mail: romich.prof@gmail.com;

Богач Ілона Віталіївна — кандидат технічних наук, доцент кафедри Автоматизації та інтелектуальних інформаційних технологій, Вінницький Національний Технічний Університет, Вінниця, e-mail: bogach.i.v@vntu.edu.ua.

Slobodian Roman V. — Postgraduate at the Department of Automation and Intelligent Information Technologies, Vinnytsia National Technical University, Vinnytsia, email: romich.prof@gmail.com;

Bhoach Iona V. — Ph.D., Associate Professor at the Department of Automation and Intelligent Information Technologies, Vinnytsia National Technical University, Vinnytsia, e-mail: bogach.i.v@vntu.edu.ua.