УДК 004.896

ПРЕДМЕТНО-ОРІЄНТОВАНА МУЛЬТИАГЕНТНА СИСТЕМА ПІДТРИМКИ РОЗРОБКИ ІНФОРМАЦІЙНИХ СИСТЕМ

Заспа Г.О., к.т.н., доцент[1]; Білик В.В.[2]
[1,2] Черкаський державний технологічний університет
[1,2] Україна, Черкаси
[1] 0000-0001-8726-7100; [2] v.v.bilyk.asp23@chdtu.edu.ua

Анотація. *Запропоновано архітектуру предметно-орієнтованої мультиагентної системи на основі великих мовних моделей для підтримки розробки інформаційної системи університету. Система складається з трьох генеративних агентів, які автоматизують генерацію вимог, моделі даних і архітектури та програмного коду. Кожен агент використовує RAG над власним зрізом доменної бази.*
Ключові слова: *штучний інтелект; агентне програмування; мультиагентні системи; інженерія програмного забезпечення; інформаційні системи.*

Вступ. Використання штучного інтелекту стало в наш час нормою при промисловій розробці програмного забезпечення (ПЗ). Штучний інтелект використовується на всіх етапах життєвого циклу розробки ПЗ – постановки задачі, проєктування, конструювання, тестування тощо, зазвичай у вигляді агентних систем, що базуються на використанні великих мовних моделей (ВММ). При розробці таких систем постають проблеми побудови їх архітектур, мультиагентної оркестрації, ролі людини в процесі розробки, ефективного використання ВММ, використання знань, специфічних для предметних галузей – всі вони відображаються в публікаціях дослідників, які працюють в цій сфері, наприклад [1-4].

Дане дослідження присвячене проблемі побудови предметно-орієнтованої інтелектуальної мультиагентної системи підтримки розробки програмних систем на прикладі інформаційної системи підтримки освітньої діяльності університету. Існуючі агентні системи в галузі інженерії програмного забезпечення мають зазвичай загальний характер. Авторами висувається гіпотеза, що предметно-орієнтований комбінований підхід, що використовує Retrieval-Augmented Generation (RAG) з доменною онтологією + структуровані приклади + база знань бізнес-правил) забезпечує статистично значущу вищу точність і повноту згенерованих артефактів порівняно з «чистою» ВММ без адаптації до предметної галузі. Метою даної фази дослідження є побудова архітектури мультиагентної системи як конвеєра (pipeline) із трьох генеративних агентів, кожен з яких відповідає за окремий етап розробки ПЗ і звертається до власного зрізу спільної бази знань домену (векторна база для RAG).

Основна частина. Наразі дослідження в галузі використання штучного інтелекту на базі ВММ ведуться переважно в сфері покращення ефективності неспеціалізованих інструментів або на окремих етапах життєвого циклу розробки програмних систем, або на певній послідовності етапів. Наприклад, в [1] провадиться огляд застосувань мультиагентних систем на етапах вимог, генерації коду, забезпечення якості та підтримки. В [2] пропонується багатокроковий ітеративний підхід із залученням людських знань, на противагу однокроковій генерації; ВММ бере участь на кожному кроці, витягуючи елементи моделі поетапно. Проте, можна зустріти дослідження про доменно-специфічні адаптації, наприклад RAG-підходів [3] або в [4] вирішується задача наскрізної автоматизації складного інженерного процесу шляхом побудови мультиагентної системи з RAG, де перед генерацією

чи дією ВММ витягує релевантну інформацію із зовнішньої доменної бази знань, хоча автори приходять до висновку, що інтеграція доменних знань в агенти лишається складною.

В рамках даного дослідження пропонується розробити архітектуру та дослідити реалізацію частин предметно-орієнтованої інтелектуальної системи на базі ВММ, що автоматизує три послідовні етапи розробки інформаційної системи управління освітньою діяльністю університету:

- генерацію специфікації вимог на основі опису предметної області;
- генерацію моделі даних та архітектури ІС;
- генерацію програмного коду з подальшою оцінкою його якості.

Базовий методологічний підхід – retrieval-augmented generation у поєднанні з агентною (multi-agent) оркестрацією, де кожен етап реалізовано окремим агентом, а вихід одного агента є вхідними даними для наступного. Розроблена архітектура зображена на рисунку 1.

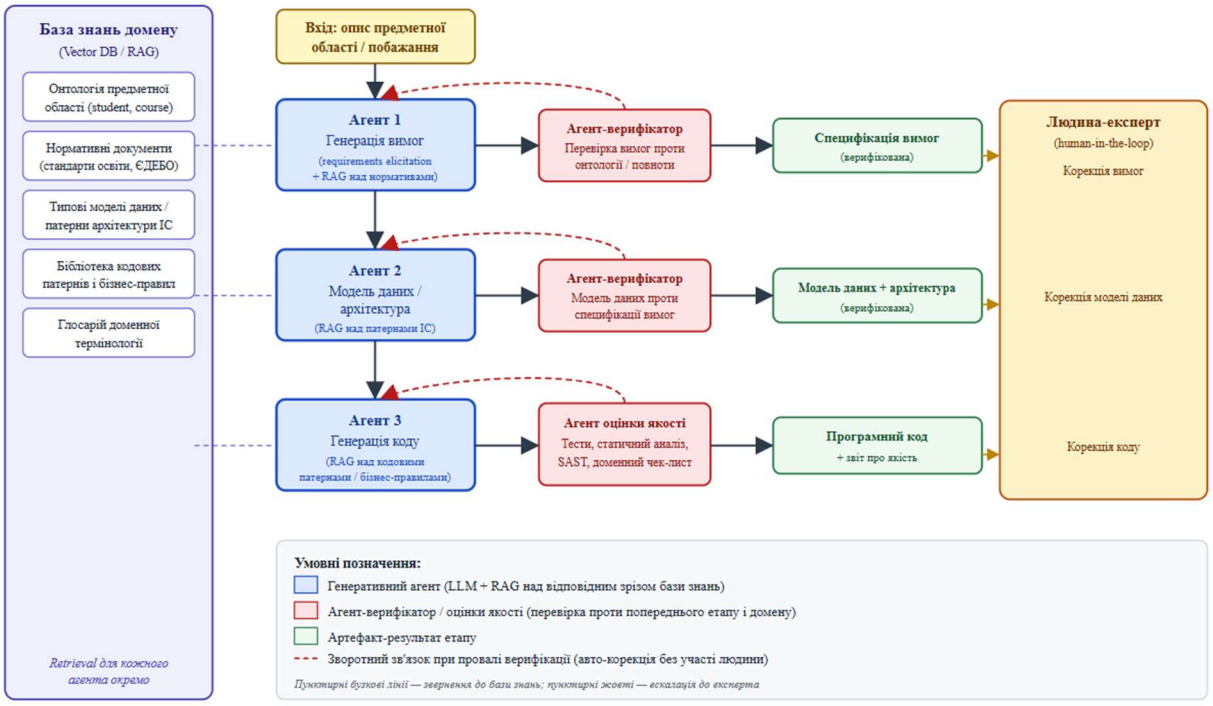


Рисунок 1. Архітектура агентної системи

Агент 1 (генерація вимог) виконує виявлення вимог на основі опису предметної області, використовуючи RAG над нормативними документами та онтологією домену. Агент 2 (модель даних / архітектура) генерує концептуальну модель даних та архітектурні рішення на основі верифікованої специфікації вимог, спираючись на бібліотеку типових патернів ІС університету. Агент 3 (генерація коду) генерує програмний код на основі верифікованої моделі даних/архітектури, використовуючи бібліотеку кодових патернів і бізнес-правил. Агенти-верифікатори – окремі виклики ВММ (або окремі спеціалізовані моделі/евристики), що перевіряють відповідність виходу кожного етапу вхідним даним і доменним обмеженням; при невдачі верифікації ініціюють автоматичну повторну генерацію (контур зворотного зв'язку на рисунку 1) без залучення людини. Агент оцінки якості коду – окремий компонент для етапу 3, що поєднує автоматичні перевірки (тести, статичний аналіз, тощо) із доменним чек-листом. Контур людини-експерта задіюється, коли автоматична верифікація не може однозначно підтвердити коректність артефакту.

В рамках розробленої архітектури були проведені експерименти з наповнення бази знань інформацією з документів університету, які стосуються освітньої діяльності. Було

визначено, що більшість нормативної інформації зберігається в документах формату docx та pdf. Досліджувалось використання бібліотек Docling, Unstructured та самостійно сформованих кастомних скриптів для отримання фрагментів інформації з документів (chunking). Для документів pdf найбільш якісний результат показало використання Unstructured в порівнянні з Docling та Docling+OCR. Для документів docx найбільш якісний результат показало використання кастомного скрипта в порівнянні з Docling та Unstructured, оскільки більшість документів не використовують стилів для заголовків, на які опираються обидві бібліотеки.

Висновки. В роботі запропоновано архітектуру предметно-орієнтованої мультиагентної системи на основі BMM для побудови інформаційних систем університетів. Система складається з трьох генеративних агентів, які автоматизують генерацію вимог, моделі даних і архітектури та програмного коду. Вона поєднує RAG над окремими зрізами доменної бази знань з автоматичною верифікацією та контуром людини-експерта, що обмежує поширення помилок між етапами розробки. Як перший крок в реалізації бази знань були проведені експерименти з розбиття інформації з документів на фрагменти, були сформовані інструменти, що дають найякісніший результат для файлів docx та pdf. Подальші дослідження спрямовані на реалізацію агентів та експериментальну перевірку переваги запропонованого підходу над BMM без доменної адаптації.

Використання інструментів ШІ. В рамках підготовки тез використовувались інструменти ШІ: Claude модель Opus 5.5 для генерування рисунка 1 та для форматування списку використаних джерел у формат APA, як помічник в пошуку літератури для дослідження, а також для генерування Python-скриптів в рамках формування бази знань.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

[1] He, J., Treude, C., & Lo, D. (2025). LLM-based multi-agent systems for software engineering: Literature review, vision, and the road ahead. *ACM Transactions on Software Engineering and Methodology*, 34(5), 1–30. https://doi.org/10.1145/3712003

[2] Yang, Y., Chen, B., Chen, K., Mussbacher, G., & Varró, D. (2024). Multi-step iterative automated domain modeling with large language models. In *Proceedings of the ACM/IEEE 27th International Conference on Model Driven Engineering Languages and Systems: Companion Proceedings* (pp. 587–595). Association for Computing Machinery. https://doi.org/10.1145/3652620.3687807

[3] Cheng, M., Luo, Y., Ouyang, J., Liu, Q., Liu, H., Li, L., Yu, S., Zhang, B., Cao, J., Ma, J., Wang, D., & Chen, E. (2026). A survey on knowledge-oriented retrieval-augmented generation. *ACM Transactions on Information Systems*. 44(8), Article 170, 1–57. https://doi.org/10.1145/3833415

[4] Fan, E., Hu, K., Wu, Z., Ge, J., Miao, J., Zhang, Y., Sun, H., Wang, W., & Zhang, T. (2025). *ChatCFD: An LLM-driven agent for end-to-end CFD automation with domain-specific structured reasoning* (arXiv:2506.02019). arXiv. https://doi.org/10.48550/arXiv.2506.02019

Zaspa H.O.[1], Bilyk V.V.[2]

Domain-specific multi-agent system for supporting information systems development

Abstract. *The architecture of a domain-specific multi-agent system based on large language models is proposed to support the development of a university information system. The system consists of three generative agents that automate the generation of requirements, the data model and architecture, and program code. Each agent uses RAG over its own slice of the domain knowledge base.*

Keywords: *artificial intelligence; agentic programming; multi-agent systems; software engineering; information systems.*