**УДК 004.912**

# ИНТЕЛЛЕКТУАЛЬНЫЙ АССИСТЕНТ ДЛЯ ПРЕОДОЛЕНИЯ ИНТЕГРАЦИОННОГО БАРЬЕРА В NO-CODE ПЛАТФОРМАХ НА БАЗЕ RAG И LLM

**Храмцов С.В.[1]**
**Научный руководитель – к.т.н, доцент, зам. директора ВШЦК Романов А. А.[1]**
**[1]Университет ИТМО**

## Введение

Сегменты малого и среднего бизнеса сталкиваются с существенными трудностями при внедрении и интеграции no-code платформ. Использование подобных решений нередко требует понимания механизмов авторизации, работы вебхуков и ряда других технических компонентов, что усложняет процесс для конечных пользователей [1]. Согласно недавним исследованиям, ключевым препятствием для end-user разработчиков остаётся интеграция с внешними API, даже несмотря на наличие инструментов искусственного интеллекта [3].

В данной работе представлена методика интеграции интеллектуального ассистента в no-code платформу Modula. Разрабатываемый ассистент формирует связи между бизнес-модулями в соответствии с предпочтениями пользователя и предоставляет инструкции по подключению внешних сервисов, опираясь на документацию Modula. Такой подход позволяет существенно снизить интеграционный барьер и повысить доступность no-code инструментов для пользователей с минимальной технической подготовкой.

## Основная часть

При внедрении интеллектуального ассистента в продукт-платформу возможно использование различных архитектурных и алгоритмических подходов. В рамках предлагаемого решения применяется микросервисная архитектура, обеспечивающая взаимодействие платформы с ИИ-ассистентом посредством API. Такой подход позволяет изолировать вычислительные компоненты, упростить масштабирование и обеспечить гибкость при обновлении моделей.

В качестве основной модели применяется qwen-coder 2.5 (openrouter-api версия) на первой итерации, так как нет мощностей для реализации квантованных моделей с качественной генерацией кода и большим контекстом, а затем будет использоваться ее квантованная версия на локальных мощностях [2].

Модель решает две основные задачи:

- Создание связных процессов, например, триггер - сообщение в телеграм боте, действие - создать задачу в bitrix, генерируя json процесса
- Генерация инструкций по связи своих сервисов с Modula для пользователя

В качестве фреймворка для постановки задачи модели используется celery, так как имеет в доступе удобный дашборд для мониторинга выполнения задач и отслеживания ошибок. При этом он не такой избыточный как airflow, runtime выполнения задач которого для работы с пользователем слишком велик.

Для решения задачи по созданию связных процессов с помощью нейросети используется подход RAG, основанный на эмбеддингах записей в базе данных модулей платформы. Сервис интеллектуального ассистента имеет свою собственную выделенную базу, которая синхронизируется с базой платформы с помощью специальных обработчиков.

Для решения задачи по генерации инструкций для пользователя используем аналогичный подход по синхронизации и векторизации документов. Каждая инструкция к новому модулю выкладывается сначала на сайт, а затем проходит через обработчика Ассистента, которая их векторизует.

Для MVP Ассистента для платформы Modula были установлены следующие критерии успеха:

- Генерация двух связных процессов: с точностью **не менее 85%**
- Время построения процесса по запросу пользователя: **менее 30 секунд**
- Время генерации инструкций для пользователя: **менее 30 секунд**

**Заключение**

Таким образом, реализованное решение демонстрирует, что использование LLM, развернутой в облачной среде, обеспечивает высокую точность ответов и качественное сохранение контекста. Учитывая необходимость строгого соблюдения форматов и порядка заполнения полей при генерации процессов, температура (T) модели была снижена до 0.1, что позволило минимизировать вариативность и повысить стабильность результатов. Удалось достигнуть следующих показателей:

- Точность генерации процессов Precision ~ 87%, Recall ~ 92%, Accuracy ~ 90%
- Медианное время генерации процессов (p95) t ~ 15.8 секунды
- Время генерации инструкций для пользователя (p95) t ~ 22 секунды

**Список литературы**


1. Gabriel Luis L. L., Ryan E., Danny C., Low-Code and No-Code Development in the Era of Artificial Intelligence: A Systematic Review // ResearchGate. – 2025. – ResearchGate:4:1218.
2. Bai J., et al. Qwen2.5-Coder Technical Report // arXiv. – 2024. – arXiv:2409.12186.
3. Weber I. Feasibility of AI-Assisted Programming for End-User Development // arXiv. –2025. – arXiv:2512.05666.