УДК 004.89

Juryev: визуальная среда оркестрации мультиагентных систем на основе графовой модели

Юрьев С.Р. (Гимназия 642)
Научный руководитель – аспирант Юрьев Р.Н. (Университет ИТМО)

Введение. В области искусственного интеллекта активно развиваются мультиагентные системы — программные комплексы, в которых несколько автономных агентов взаимодействуют для решения сложных задач. Существующие фреймворки оркестрации агентов (LangChain, CrewAI, AutoGen) предоставляют высокоуровневые абстракции, скрывающие внутреннюю логику маршрутизации данных и взаимодействия компонентов. Пользователь не контролирует, как именно агенты обмениваются информацией и какая логика маршрутизации применяется. В данной работе представлена система Juryev — визуальный low-level оркестратор агентов, обеспечивающий полный контроль над потоком данных и логикой исполнения через графовый интерфейс.

Основная часть. Система Juryev реализует графовый подход к построению конвейеров интеллектуальных агентов. Пользователю предоставляется визуальный канвас — бесконечное рабочее пространство, на котором из базовых блоков собираются произвольные конвейеры обработки данных. В системе используются два типа блоков: ноды (прямоугольники) — элементы, исполняющие произвольный Python-код для парсинга, маршрутизации, фильтрации и вызова API, и агенты (ромбы) — элементы, осуществляющие вызов языковых моделей (LLM) с одним входом и одним выходом.

Архитектура системы

Инициализация потока: Пользователь запускает стартовую ноду, которая генерирует начальные данные и передаёт их через функцию emit(data) к downstream-узлам графа.

Обработка данных: Ноды последовательно исполняют Python-код, трансформируя данные. Конвейер представляет собой направленный граф, узлами которого являются ноды и агенты, а рёбрами — связи передачи данных между ними.

Вызов языковой модели: Агент-ноды получают контекст от предшествующих нод и передают его языковой модели. Результат генерации определяется промптом агента и контекстом, накопленным в общем состоянии конвейера.

Оркестрация: Система предоставляет набор примитивов управления: emit(data) — передача данных и запуск следующих нод, state — персистентная память между запусками, trigger("name") — вызов произвольной ноды по имени, sleep(ms) и every(ms, fn) — управление временем исполнения.

Преимущества

Прозрачность: Пользователь видит весь граф исполнения, данные и статус каждой ноды в реальном времени, в отличие от «чёрных ящиков» существующих фреймворков.

Гибкость: Произвольный Python-код в нодах позволяет реализовать любую логику обработки без ограничений API фреймворка, а агент-ноды поддерживают подключение любой языковой модели.

Модульность: Каждый блок является независимым компонентом, который можно переиспользовать в различных конвейерах, что позволяет собирать RAG-системы, мультиагентные конфигурации и self-reflection циклы.

Недостатки

Порог входа требует знания языка Python для написания логики нод.

Выводы. Была разработана система визуальной оркестрации интеллектуальных агентов Juryev и проведён анализ её сильных и слабых сторон по сравнению с существующими фреймворками.

Список использованных источников:

1. Wang L., Ma C., Feng X. et al. A Survey on Large Language Model based Autonomous Agents // Frontiers of Computer Science. — 2024. — Vol. 18, No. 6. — Art. 186345.
2. LangGraph Documentation [Электронный ресурс]. URL: https://docs.langchain.com/oss/python/langgraph/overview (дата обращения: 18.02.2026).
3. Wu Q., Bansal G., Zhang J. et al. AutoGen: Enabling Next-Gen LLM Applications via Multi-Agent Conversation // arXiv preprint arXiv:2308.08155. — 2023.
4. Маслобоев А.В. Принципы разработки прикладных мультиагентных систем управления жизнеспособностью критических инфраструктур // Труды Института системного анализа РАН. — 2024. — Т. 74, № 2. — С. 62–81.

Автор: _____ / Юрьев Серафим Родионович /

Научный руководитель: _____ / Юрьев Родион Николаевич /