

СРАВНИТЕЛЬНЫЙ АНАЛИЗ ПЛАТФОРМЫ OSTIS И ПОПУЛЯРНЫХ ФРЕЙМВОРКОВ ДЛЯ РАЗРАБОТКИ ИИ-АГЕНТОВ

Сравнивается Платформа OSTIS с популярными фреймворками для разработки ИИ-агентов (LangGraph, AutoGen и др.). Анализируются архитектура, представление знаний, управление агентами, интеграция моделей, сильные/слабые стороны.

ВВЕДЕНИЕ

Разработка ИИ-агентов — программ, способных автономно воспринимать, рассуждать и действовать — ключевое направление ИИ. Существуют различные фреймворки для их разработки: LangGraph, AutoGen, CrewAI, LlamaIndex и др [1]. Платформа OSTIS предлагает альтернативный, знание-ориентированный подход на основе семантических сетей, решающий некоторые их ограничения [2].

I. СРАВНИТЕЛЬНЫЙ АНАЛИЗ

Ключевое отличие Платформы OSTIS заключается в ее фундаменте — Технологии OSTIS, которая использует единую семантическую основу на базе универсального языка — SC-кода для представления как декларативных (факты, онтологии), так и процедурных (алгоритмы, агенты) знаний в виде графовых структур в общей семантической памяти (sc-памяти). Обработка знаний основана на событийно- и агентно-ориентированных моделях.

Другие популярные фреймворки используют иные подходы, часто фокусируясь на интеграции больших языковых моделей (LLM) и оркестрации задач:

1. *LangGraph* — расширение экосистемы LangChain, использует граф-ориентированную архитектуру для определения и выполнения шагов агента как узлов в направленном графе, позволяя создавать *циклические потоки* выполнения и гибко управлять состоянием.
2. *AutoGen* основан на *мультиагентных асинхронных беседах*, где агенты взаимодействуют друг с другом для решения задач, часто требуя минимального вмешательства человека.
3. *CrewAI* применяет ролевой подход, где агенты с определенными ролями и задачами сотрудничают в "команде" для достижения общей цели (*ролевая совместная работа*), поддерживая последовательное и иерархическое выполнение задач.
4. *LlamaIndex* фокусируется на интеграции с данными (Retrieval-Augmented Generation — RAG), предоставляя инструменты для *индексации и извлечения* информации из больших объемов данных для использования агентами.

5. *Semantic Kernel* использует модульный подход на основе "навыков"(skills) и плагинов, позволяя комбинировать ИИ-функции и обычный код, ориентирован на *интеграцию в корпоративные среды*, особенно в экосистеме Microsoft (.NET).

Для систематизации и наглядного представления различий между Платформой OSTIS и рассмотренными фреймворками, их ключевые характеристики сведены в таблицу 1.

Проведенный сравнительный анализ позволяет выделить сильные стороны и ограничения каждого из подходов:

1. Платформа OSTIS эффективна для сложных интегрированных систем (глубокие знания, адаптивность). Другие фреймворки обеспечивают быструю разработку узких задач (автоматизация, RAG) и легкую интеграцию LLM (Python).
2. Платформа OSTIS имеет крутую кривую обучения и меньшее сообщество. Популярные фреймворки зависят от недостатков LLM (непрозрачность, галлюцинации), предлагают поверхностную интеграцию знаний и имеют проблемы масштабирования в комплексных задачах.

ЗАКЛЮЧЕНИЕ

Платформа OSTIS применяет знание-ориентированный подход, в отличие от LLM-центричных/оркестрационных фреймворков. Ее единая семантическая основа и глубокое представление знаний позволяют создавать сложные, семантически совместимые интеллектуальные системы для комплексных задач, требующих интеграции разнородных знаний и моделей.

1. Joshi, S. Review of autonomous systems and collaborative AI agent frameworks / S. Joshi // SSRN Electronic Journal : preprint. — 2025.
2. Zotov, N. Software platform for next-generation intelligent computer systems / N. Zotov // Open Semantic Technologies for Intelligent Systems (OSTIS-2022) : сборник научных статей / БГУИР ; редкол.: В. В. Голенков [и др.]. — Минск, 2022. — Вып. 6. — С. 297–326.

Зотов Никита Владимирович, магистрант, кафедра ИИТ БГУИР, n.zotov@bsuir.by

Научный руководитель: Голенков Владимир Васильевич, д. техн. наук, профессор, golen@bsuir.by

Таблица 1 – Сравнительный анализ Платформы OSTIS и популярных фреймворков ИИ-агентов

Критерий	Платформа OSTIS	Популярные фреймворки
Основная парадигма	Знание-ориентированная. Интеграция разнородных знаний и моделей на базе семантических сетей (SC-код). Цель – создание <i>семантически совместимых</i> интеллектуальных систем.	LLM-центричная / Оркестрационная. Оркестрация LLM, мульти-агентное взаимодействие (через чат, роли, графы), интеграция с данными (RAG), использование внешних инструментов.
Представление знаний	Глубокое, унифицированное. Семантические сети (SC-код) для декларативных и процедурных знаний. Спецификации агентов – часть единой базы знаний.	Различное, часто неформальное или специфичное для задачи. Состояние в графах (LangGraph), неявное в LLM/беседах (AutoGen), векторные базы/индексы (LlamaIndex), объекты состояния, структурированный вывод (JSON/Pydantic). Обычно нет <i>единой формальной семантической основы</i> .
Интеграция моделей решения задач	Глубокая, семантическая. Интеграция различных методов (декларативных, процедурных, логического вывода, машинного обучения) на общей семантической основе – SC-коде.	Поверхностная/Модульная. Фокус на одном подходе (workflow, chat, RAG) или комбинация через API/плагины/вызовы функций без глубокой семантической интеграции.
Управление агентами	Событийно/агентно-ориентированная. Агенты реагируют на события в <i>sc-памяти</i> (изменения в графе знаний) и выполняют действия согласно спецификации (статической/динамической/полудинамической).	Разнообразная. Явные графы потоков работ (LangGraph), асинхронные беседы/сообщения (AutoGen), роли и задачи в "команде"(CrewAI), плагины/навыки (Semantic Kernel), вызов функций/инструментов LLM (OpenAI SDK и др.).
Гибкость и масштабируемость	Высокая. Модульная архитектура, унифицированное представление знаний и нацеленность на параллелизм способствуют масштабированию и адаптации сложных систем.	Различается. LangGraph предлагает масштабируемую инфраструктуру для потоков работ. Другие могут иметь ограничения при интеграции или в сложных сценариях. Существуют общие проблемы масштабирования и управления контекстом для AI агентов на базе LLM.
Память	Встроенная семантическая. <i>sc-память</i> как единая графовая база знаний для всей системы, обеспечивающая долговременную и контекстуальную память.	Явная/Внешняя/Зависимая от LLM. Управление состоянием в графе (LangGraph), краткосрочная/долгосрочная память (CrewAI), интеграция с векторными БД для RAG (LlamaIndex), управление памятью в Semantic Kernel, контекстное окно LLM.
Простота использования / Порог входа	Высокий. Требует изучения Технологии OSTIS, SC-кода и специфической архитектуры. Более крутая кривая обучения.	Различается. Некоторые фреймворки, особенно с GUI-инструментами (например, AutoGen Studio) или ориентированные на роли (CrewAI), считаются более интуитивными. Другие (LangGraph, Semantic Kernel) могут быть сложнее. Часто используется знакомый язык Python и экосистема вокруг LLM.