

## LangGraph 기반 CrewAI 다중 에이전트 통합 아키텍처

김재호<sup>1</sup> · 김장영<sup>2\*</sup>

### LangGraph-Based Integrated Architecture for CrewAI Multi-Agent Systems

Jae-Ho Kim<sup>1</sup> · Jang-Young Kim<sup>2\*</sup>

<sup>1</sup>Graduate Student, Department of Computer Science, The University of Suwon, Hwaseong, 18323 Korea

<sup>2</sup>\*Associate Professor, Department of Computer Science, The University of Suwon, Hwaseong, 18323 Korea

#### 요 약

최근 대형 언어 모델(LLM)의 급속한 발전과 함께, 다양한 도메인에서 복잡한 문제를 해결하기 위한 Multi-Agent System(MAS)이 새롭게 주목받고 있다. 본 연구에서는 역할 기반 협업 구조의 CrewAI와 그래프 기반 워크플로우 및 상태 오케스트레이션에 강점을 지닌 LangGraph를 통합한 하이브리드 아키텍처를 제안한다. CrewAI는 Agent, Task, Crew 단위의 모듈화로 개발 현장에서의 접근성을 높이지만, 내부 데이터 흐름과 협업 과정은 블랙박스화되어 복잡한 시나리오에서는 투명성과 디버깅에 한계가 있다. 반면, LangGraph는 상태 기반 그래프 모델을 통해 전체 워크플로우를 명확하게 시각화하여 확장성과 유지보수성을 크게 향상시킨다. 특히, 본 연구는 CrewAI의 엔티티를 LangGraph의 노드로 래핑하여, 모듈화된 워크플로우와 유연한 오케스트레이션, 그리고 동적 도구 통합을 실현함으로써, 고도화된 RAG 파이프라인과 도메인별 응용에서 우수한 성능과 확장성을 실증적으로 입증하였다.

#### ABSTRACT

With the rapid advancement of large language model (LLM), Multi-Agent Systems (MAS) have garnered renewed attention as a means of solving complex problems across diverse domains. This paper proposes a hybrid architecture that integrates CrewAI, which features a role-based collaboration structure, with LangGraph, known for its strengths in graph-based workflow and state orchestration. While CrewAI enhances accessibility in development environments through modularization at the Agent, Task, and Crew levels, its internal data flows and collaborative processes often become black boxes, limiting transparency and debugging in complex scenarios. In contrast, LangGraph significantly improves system scalability and maintainability by clearly visualizing the entire workflow with a state-based graph model. In this study, we demonstrate that, in particular, wrapping CrewAI entities as LangGraph nodes enables modular workflows, flexible orchestration, and dynamic tool integration, thereby empirically validating superior performance and scalability in advanced RAG pipelines and domain-specific applications.

**키워드 :** 임베딩, 대형 언어 모델, 검색 증강 생성(RAG), 멀티에이전트시스템

**Keywords :** Embedding, LLM(Large Language Model), RAG(Retrieval Augmented Generation), Multi-Agent System

---

Received 9 July 2025, Revised 21 July 2025, Accepted 28 July 2025

\* Corresponding Author Jang-Young Kim(E-mail:jykim77@suwon.ac.kr, Tel:+82-31-229-8345)

Associate Professor, Department of Computer Science , The University of Suwon, Hwasung, 18323 Korea

**Open Access** <http://doi.org/10.6109/jkiice.2025.29.8.987>

pISSN:2234-4772

©This is an Open Access article distributed under the terms of the Creative Commons Attribution Non-Commercial License(<http://creativecommons.org/licenses/by-nc/3.0/>) which permits unrestricted non-commercial use, distribution, and reproduction in any medium, provided the original work is properly cited.  
Copyright © The Korea Institute of Information and Communication Engineering.

## I. 서 론

대형 언어 모델의 혁신적인 발전은 단순한 질의응답을 넘어, 보다 복잡하고 맥락적인 문제를 해결할 수 있는 능력을 갖춘 AI 에이전트 시스템의 구현을 가능케 하였다. 그 결과로 단일 LLM 기반 시스템에서 벗어나, 복수의 에이전트가 각기 다른 역할과 책임을 맡아 상호작용하며 협업하는 MAS(Multi-Agent System) 패러다임이 자연스럽게 등장했다[1]. 기존의 단일 모델 기반 접근은 모든 정보를 한 모델에 집중시키는 방식이었으나, 점차 역할 분담과 전문화, 협업 구조가 강조되면서 다양한 에이전트 프레임워크들이 발전하기 시작했다. 대표적인 예시로는 AutoGPT, BabyAGI 등 단일 LLM의 자기지향적 플래닝 능력을 시연한 시스템이 있으며, 이후 ChatDev[2], AgentVerse, CAMEL 등에서는 역할 기반 에이전트 구성을 바탕으로 한 협업 및 대화 시나리오가 도입되었다. 그러나 이들 대부분은 에이전트 간의 데이터 흐름이나 시스템 전체의 구조적 투명성, 그리고 유지 보수 측면에서 한계를 드러냈다.

이러한 맥락에서 CrewAI는 역할, 태스크, 크루라는 개념적 단위를 통해, 각 에이전트가 특정 역할을 수행하고 이를 구조화하여 복잡한 협업 시나리오를 구현할 수 있게 하였다[3]. CrewAI의 가장 큰 장점은 추상화된 에이전트 설계와 직관적인 역할 분담, 그리고 모듈화된 태스크 배정이 가능하다는 점이다. 특히 복잡한 도메인에서 각 에이전트의 전문성을 명확하게 설정할 수 있고, 협업의 단위인 크루를 통해 전체 시스템을 계층적으로 관리할 수 있다. 하지만 내부적으로 데이터 이동 경로나 에이전트 간 상호작용이 블랙박스화되기 쉽고, 시스템 전체의 동작 흐름을 시각적으로 추적하거나, 조건 분기와 반복, 예외 처리 등 복잡한 워크플로우를 관리하기에는 다소 한계가 있었다.

이에 반해 LangGraph는 언어 모델 워크플로우를 그래프 기반 상태 기계(state machine)로 모델링할 수 있게 함으로써, 각 노드에서의 작업과 상태 전이, 데이터 흐름을 명확히 시각화할 수 있는 장점을 제공한다. LangGraph를 활용하면 복잡한 멀티에이전트 협업 구조를 그래프 형태로 직관적으로 설계할 수 있으며, 조건 분기, 반복 실행, 오류 처리 등도 노드와 엣지의 조합만으로 손쉽게 구현할 수 있다. 실제로 LangGraph는 오픈스택레이션 및 상태 지속성, 체크포인트 기능 등도 지원

하여, 장시간 실행되는 멀티에이전트 워크플로우의 복원성 및 신뢰성을 강화한다[4]. CrewAI와 LangGraph 각각의 장점에도 불구하고, 두 프레임워크가 분리된 채로는 실제 현장 적용에서 생산성이나 운영 효율성, 확장성 면에서 한계가 명확하다. 따라서 본 연구은 CrewAI의 역할 기반 협업 구조와 LangGraph의 그래프 기반 오픈스택레이션의 장점을 결합하여, 실제로 프로덕션 환경에서 사용할 수 있는 강력하고 유연한 통합 아키텍처를 제시하고자 한다.

## II. 배경 및 기존연구

단일 에이전트 시스템의 대표적인 AutoGPT는 ReAct(Reasoning+Acting) 패턴을 활용하여 단일 LLM이 일련의 작업을 순차적으로 수행하는 구조를 제시하였다. 이러한 방식은 복잡한 문제를 하나의 거대한 체인으로 해결한다는 점에서 혁신적이었으나, 협업이 필요한 복합 시나리오에서는 명확한 한계를 드러냈다[5]. 이어 등장한 AutoGen은 전문가 에이전트 간의 협업과 사용자 프록시 개념을 도입하여 대화형 코드 생성과 협력적 문제 해결에 대한 가능성을 확장했다. CrewAI는 에이전트와 태스크, 크루의 명확한 역할 구분을 통해 협업 효율성과 구조적 확장성을 동시에 달성하였고, 특히 엔티티 기반 장기 메모리와 시나리오 기반 팀워크 설계가 실제 업무에 적합함을 입증했다[3,6].

LangGraph는 상태 기반 워크플로우와 메시지 패싱 시스템을 통합하여, 멀티에이전트 간의 복잡한 협업 로직을 그래프로 시각화하고 통제할 수 있는 아키텍처를 제안하였다. 이를 통해 각 에이전트의 역할과 데이터 이동 경로, 오류 처리 과정, 휴면 인 더 루프 등 다양한 요구사항을 하나의 구조 안에서 직관적으로 표현할 수 있게 되었다. 최근의 연구들은 LangGraph가 다양한 언어 에이전트를 조합하여 번역, 질의응답, 멀티도메인 지원 등 복합 문제 해결에 효과적임을 실험적으로 보여주고 있다[4,7].

또한, Retrieval-Augmented Generation(RAG) 구조의 발전에 따라, 단순한 정보 검색을 넘어서 문서 요약, 필터링, 재질문 등 능동적이고 적응적인 정보 획득 전략(Agentic RAG, Adaptive RAG)이 멀티에이전트 시스템에서 실질적 성능 개선을 이끌어내고 있다[8]. STORM

과 같은 구조는 에이전트 간 역할 분담이 정보 수집 및 응답 품질에 중대한 영향을 미친다는 점을 실험적으로 입증하였다[9].

### III. 제안 방법 및 아키텍처

기존 CrewAI와 LangGraph 프레임워크는 각각 역할 기반 모듈화와 상태 기반 워크플로우 제어라는 장점을 지니고 있으나, 단독으로는 복잡한 조건 분기, 평가, 반복, 다양한 도구 통합 등 실제 대규모 멀티에이전트 시스템 구현에서 한계가 있었다[3,4].

본 연구에서 제안하는 통합 구조는 CrewAI의 구성요소(agent, task, crew)를 LangGraph의 노드로 캡슐화(wrapping)함으로써, 명확한 상태 기반 흐름 제어와 모듈화된 워크플로우 오케스트레이션을 동시에 실현한다. 래핑 전략은 시스템의 요구에 따라 Agent 단위, Task/Tool 단위, Crew 단위로 계층화할 수 있으며, 각 계층별로 제어권한과 유연성을 다르게 부여할 수 있다 [3,4].

Agent 단위 래핑은 각 에이전트를 LangGraph 내 독립 노드로 구현하여, 실행 순서, 데이터 흐름, 조건 분기, 반복 등을 세밀하게 제어할 수 있다. 예를 들어, 특정 에이전트의 출력이 미흡할 경우, 평가 노드를 통해 조건 분기 후 재실행하거나 다른 에이전트로 분기하는 복잡한 로직도 명확히 구현할 수 있다. Task/Tool 단위 래핑에서는 검색, 계산, API 호출 등 특정 기능 단위로 노드를 생성하여, 에이전트가 description 기반으로 다양한 도구를 상황에 맞게 해석·호출할 수 있다[3]. Crew 단위 래핑은 전체 크루(복수 에이전트와 태스크의 집합)를 단일 LangGraph 노드로 취급함으로써, 이미 정의된 고수준 협업 구조를 외부에서 블랙박스화하여 재사용성을 높인다. 이러한 다중적 래핑 구조는 실제 개발 시 요구에 따라 손쉽게 조합할 수 있다.

구현 단계에서는, 각 CrewAI 에이전트 및 태스크를 LangGraph 노드로 정의하고 입력 파라미터, 출력값, 상태 전이 조건을 명확히 하여 데이터 흐름과 협업 과정을 그래프 형태로 시각화할 수 있다. Tool 사용은 description 필드를 통한 자연어 인터페이스로 추상화되어, LLM 기반 에이전트가 각 Tool의 목적과 사용법을 자율적으로 해석, 상황에 맞게 활용하도록 설계하였다.

Retriever, 외부 API, 계산기 등 다양한 Tool을 하나의 구조 내에서 통합할 수 있어, 도메인 확장 및 재사용이 매우 용이하다[4].

이러한 통합 구조를 통해 다음과 같은 기술적·구조적 이점을 실현할 수 있었다.

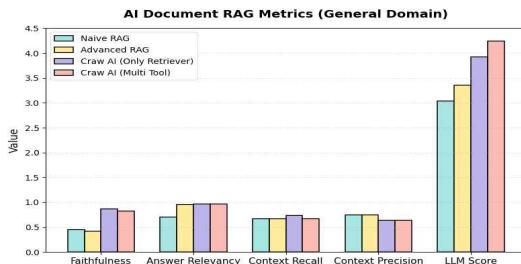
첫째, 데이터 흐름과 시스템 실행 경로가 각 노드 단위로 명확히 관리되어, 전체 워크플로우와 응답 처리를 외부에서 체계적으로 제어할 수 있다. 복잡한 조건 분기, 반복 실행, 실패 대응 등도 그래프 구조 내에서 일관성 있게 처리할 수 있다.

둘째, Tool 호출 및 통합은 description 기반 프롬프트와 일관된 인터페이스를 활용하여, 다양한 도구를 자연스럽게 조합할 수 있고, 새로운 도구의 추가·교체, 도메인 확장도 높은 유연성으로 지원한다.

셋째, LangGraph의 상태 기반 제어는 각 노드 실행 결과에 따른 동적 흐름 전환과 예외 상황 대응, 시스템 신뢰성 향상에 크게 기여한다. 평가 노드를 독립적으로 추가하여, BLEU, ROUGE, factual consistency 등 자동화된 품질 메트릭과 human in the loop 평가도 효과적으로 연계할 수 있다[10]. 이러한 평가지표와 평가 방식은, 본 논문의 실험뿐만 아니라 최근 RAG 및 NLP 분야에서 성능 비교와 품질 검증의 객관적·표준적 기준으로 널리 채택되고 있다.

### VI. 실험 및 평가

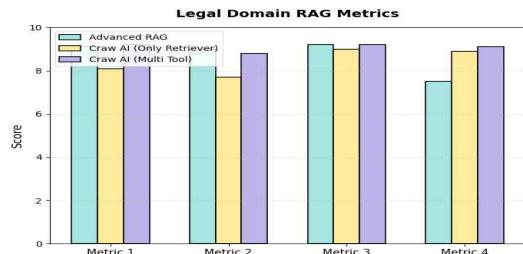
본 연구에서는 AI 기술 관련 문서, 법률·의료 도메인 문서, 연구 지원형 멀티에이전트 시스템 등 세 가지 유형의 실험군을 설정하여, 각 시스템의 성능을 정량적·정성적으로 평가하였다. 첫 번째 실험군은 일반 AI 문서를 대상으로 한 RAG 시스템으로, RAGAS Question Generator 및 Critic을 활용한 Faithfulness, Answer relevancy, Context Recall, Context Precision, LLM Score 등의 지표로 평가하였다. CrewAI+LangGraph(멀티툴 통합형)는 모든 평가항목에서 기존 RAG 대비 높은 점수를 기록하였다. 특히 멀티툴 통합형은 LLM Score 4.24, Faithfulness 0.87 등 가장 우수한 성과를 보였다.



**Fig. 1** AI Document RAG Metrics (General Domain). The chart compares four configurations across five evaluation metrics: Faithfulness, Answer Relevancy, Context Recall, Context Precision, and LLM Score.

위 그림1은, Crew AI(Only Retriever)는 Faithfulness(0.87)에서 가장 높은 점수를 기록하며, 원문 근거에 충실한 응답 생성에서 강점을 보였다. Answer Relevancy는 Multi Tool 구성에서 최고점(0.97)을 기록, 복수 도구 통합이 질문 의도 적합도에 크게 기여함을 알 수 있다. Context Recall은 전반적으로 고르게 분포되어 있으나 Only Retriever가 소폭 우위를 보인다. 반면 Context Precision은 Naive 및 Advanced RAG에서 더 높게 나타나, 단순한 검색 구조가 맥락의 정밀도에는 유리할 수 있음을 시사한다. LLM Score 역시 Multi Tool에서 가장 높게 나타나, LLM 기반 생성 품질(창의성, 응답 일관성 등) 향상에 멀티툴 전략이 효과적임을 확인하였다.

두 번째 실험은 법률 도메인에 특화된 QA 시스템을 대상으로, 정확성, 완결성, 명료성 등 실질적 응답 품질 기준을 적용하여 평가하였다. 법률 분야 특성상, 답변의 정답 일치도, 필수 정보 누락 여부, 용어·조문 인용의 명확성 등이 실용적 평가 기준이 되었다. CrewAI+LangGraph(멀티툴)는 모든 항목에서 평균 8.94점으로 가장 높은 성능을 나타냈으며, 단일 Tool이나 기존 RAG 대비 복잡한 질의에도 높은 적합성과 완성도를 보였다.



**Fig. 2** Legal Domain RAG Metrics. A comparative visualization of three RAG configurations—Advanced RAG, Craw AI (Only Retriever), and Craw AI (Multi Tool)—across four evaluation metrics: Accuracy, Completeness, Clarity, and Final Score.

그림 2는 법률 도메인에서 세 가지 RAG 구성 (Advanced RAG, Craw AI 단일 Retriever, Craw AI Multi Tool)의 네 가지 평가 지표(정확성, 완전성, 명료성, 종합점수)별 성능을 시각적으로 비교한 것이다. Advanced RAG는 대부분의 항목에서 고르게 높은 점수 (9점대)를 기록했으나, Metric 4에서는 7.4점으로 다소 부족함을 보였다. Craw AI (Only Retriever)는 전반적으로 낮은 점수(7점대)를 기록하여, 단일 검색만으로는 법률 문서의 정확성과 완전한 커버리지에 한계가 있음을 시사한다. 반면 Craw AI (Multi Tool)는 모든 지표에서 가장 높은 점수(69.13)를 보여, 툴 결합이 Retrieval-Generation 품질을 현저히 향상시킴을 실증적으로 보여준다.

STORM RAG와 CrewAI + LangGraph 시스템의 성능을 총 15개의 세부 기준으로 평가하였다. 각 기준은 주제 적합성, 구조 비교의 명확성, 개념적 정확성, 고급 검색 전략 활용, 생성 전략 다양성, 평가 매트릭 활용, 복잡 질의 대응력, 시스템 아키텍처 설명력, 실제 적용사례, 논리 구조 및 문장 가독성, 참고문헌 신뢰성 등 다양한 측면을 포괄한다. 두 시스템 모두 주제 적합성, 개념적 정확성에서 만점을 기록하는 등 기본적인 측면에서 강점을 보였다. 그러나 CrewAI + LangGraph는 모듈형 아키텍처의 명확성, 실제 배포와 운영 유연성, 통합 가능성, 도메인별 실제 적용사례, 설명의 명확성 및 출처 신뢰도 측면에서 STORM RAG 대비 더 높은 평가를 받았다.

총점 기준으로는 STORM RAG가 68.5점, CrewAI + LangGraph가 71.0점으로 집계되었다. CrewAI + LangGraph가 STORM RAG 대비 약 2.5점 높은 점수를

기록하며 전반적인 우위를 보였다. 이 결과는 명확성, 통합 유연성, 엔드투엔드 추론 능력 등에서 모듈형 RAG 구조의 설계적·운영적 강점을 뚜렷하게 보여준다. 특히 실제 현장에 요구되는 높은 적응성, 확장성이 필요한 상황에서 CrewAI + LangGraph의 효과적 우위가 실증적으로 확인되었다.

자동 평가(RAGAS) 도구의 경우, 도메인 특화 문서에서는 핵심 개념 누락, 문맥 오도, 케이스 커버리지 불완전, 중복 질문 등 한계가 드러났으며, 실제 현장에서는 전문가 수작업 검수 및 별도의 평가 기준(15항목, 5점 척도) 도입이 필수적임을 확인하였다. 다음은 각 metric에 대한 정확한 점수이다.

**Table. 1** Comparison of RAG Configurations in the Legal Domain Across Four Evaluation Metrics

System	Metric1	Metric2	Metric3	Metric4
Advanced RAG	9.0	9.0	9.0	7.4
CrewAI (Only Retriever)	8.17	7.60	8.13	7.97
CrewAI (Multi Tool)	9.13	8.60	9.1	8.94

그림 2의 Metric 1~4는 각각 정확성(Accuracy), 완전성(Completeness), 명료성(Clarity), 종합점수(Final Score)에 해당한다.

CrewAI와 LangGraph의 통합 구조는 기존 코드 수정 없이 노드 단위 wrapping만으로 다양한 워크플로우 및 조건 분기를 유연하게 설계할 수 있어, 통합 비용 절감, 코드 재사용성, 유지보수 효율성 측면에서 현저한 개선 효과를 확인하였다. 이러한 장점들은 실제 실험과 도메인별 적용을 통해 확인되었으며, 기존 연구에서 한계로 지적되었던 복잡한 분기, 반복, 평가, 동적 도구 통합, 유지보수 측면의 문제를 효과적으로 해결하였다.

## V. 심층 분석 및 비교

LangGraph 단독 사용 시, 그래프 기반의 고도로 사용자 정의 가능하고 제어 가능한 에이전트 워크플로우를 구축할 수 있다. 이는 복잡한 조건 분기, 상태 관리, 다양

한 외부 시스템과의 연동 등에서 강력한 제어력과 확장성을 제공한다는 점에서 명확한 강점이 있다. 그러나 CrewAI에 비해 에이전트 역할 정의와 협업 관리가 상대적으로 복잡하며, 모든 협업 논리를 사용자가 직접 설계하고 구현해야 한다는 점에서 개발자의 부담이 커질 수 있다. 학습 곡선 역시 상대적으로 높아, 대규모 시스템의 초기 설계 및 반복적인 디버깅에 더 많은 시간이 요구된다.

반면 CrewAI는 사용 편의성과 협업적 지능에 중점을 둔 구조로, 에이전트 간 역할 기반 협업이 프레임워크 차원에서 내장되어 있다. 이를 통해 복잡한 멀티에이전트 시스템도 빠르게 설계하고 운영할 수 있으며, 실제 프로토타입 제작이나 반복적인 시스템 확장에 특히 유리하다. 하지만 워크플로우의 세밀한 제어, 조건 분기, 상태 지속성, 외부 시스템과의 폭넓은 통합과 같은 부분에서는 한계를 드러낸다. CrewAI만으로는 복잡한 애플리케이션에서의 정교한 오케스트레이션이 어렵고, 개발자가 직접 세부 제어를 구현하기에는 구조적인 제약이 있다.

이에 반해 두 프레임워크를 결합한 통합 아키텍처는 각각의 강점을 상호 보완할 수 있다. 통합형 구조에서는 LangGraph가 제공하는 그래프 기반의 워크플로우 관리와 CrewAI의 역할 기반 에이전트 설계를 함께 활용함으로써, 복잡한 멀티에이전트 시스템에서도 제어와 사용 편의성 간의 균형을 달성할 수 있다. 구체적으로, CrewAI의 높은 수준의 협업 설계와 LangGraph의 세밀한 워크플로우 제어를 동시에 활용하여, 복잡성 관리와 아키텍처 설계의 명확성을 확보할 수 있다. 물론 통합형 구조는 두 프레임워크 모두에 대한 충분한 이해가 요구되며, 학습 곡선이 단일 프레임워크에 비해 다소 높을 수 있다. 그러나 명확한 아키텍처 및 인터페이스 설계를 통해 실제 현장 적용 시 개발 생산성, 운영 효율성, 품질·확장성 등 여러 측면에서 실질적인 이점을 얻을 수 있었다. 특히 본 연구의 통합 구조는 실제 배포 경험과 다양한 도메인 실험을 통해 높은 적응성과 실효성을 보였다. 복잡한 조건 분기, 외부 시스템 연동, 도메인별 맞춤형 협업 등 실무적 요구를 통합 구조가 효과적으로 충족시켰다. 이러한 분석은 향후 대규모 AI 오케스트레이션 환경에서 통합형 아키텍처가 더욱 중요한 역할을 하게 됨을 시사한다.

## VI. 결론

본 연구은 CrewAI의 직관적인 에이전트 설계와 LangGraph의 상태 기반 오케스트레이션 구조를 통합한 새로운 아키텍처를 제시하였다. Crew, Agent, Task/Tool 단위로 세분화된 래핑 전략과 명확한 상태 기반 워크플로우 설계를 통해, 실제로 다양한 도메인에서 높은 성능과 유연성을 보장하는 멀티에이전트 시스템을 구현할 수 있음을 실험적으로 입증하였다. 데이터 흐름의 시각화, 조건 분기 및 반복, Tool 통합, 평가 노드 삽입, Human-in-the-loop 감독 구조 등 통합 아키텍처만의 강점은 프로덕션 환경에서의 실질적 효율성과 성능 향상으로 이어졌다.

그러나 LangGraph와 CrewAI 통합에는 상태 관리 불일치, 데이터 공유 방식 차이, 통신 패턴 조정 등에서 구현상의 복잡성이 뒤따르며, 두 프레임워크 모두에 대한 깊은 이해가 필요하다. 이러한 점은 단일 프레임워크 대비 높은 학습 곡선과 설계 비용을 유발할 수 있다. 향후 연구에서는 상태 동기화 인터페이스, 도메인 특화 멀티에이전트 템플릿 개발, 실시간 피드백 기반 튜닝, 평가 워크플로우 자동화 도구 개발 등이 주요 과제로 남아 있다. CrewAI+LangGraph 기반 통합 구조는 단순 도구 결합을 넘어, 모듈성, 제어력, 협업성을 동시에 확보할 수 있는 새로운 기준점이 될 것이며, LLM 기반 AI 오케스트레이션 분야의 실질적 전환점을 제공할 것으로 기대한다. 본 연구의 통합 아키텍처는 다양한 도메인에서의 실현을 통해 뛰어난 성능과 유연성을 입증했으며, 향후에는 상태 동기화 자동화, 도메인별 특화 템플릿, 실시간 평가 및 피드백 기반 튜닝 등 실무적 연구가 활발히 이루어질 필요가 있다.

결론적으로, CrewAI+LangGraph 구조는 실제 서비스 및 다양한 도메인에서의 적용을 통해 우수한 성능과 효과를 실증적으로 확인하였으며, 향후 LLM 기반 멀티에

이전트 시스템의 발전에 크게 기여할 것으로 기대된다.

## REFERENCES

- [1] H. Du, S. Thudumu, R. Vasa, and K. Mouzakis, “A Survey on Context-Aware Multi-Agent Systems: Techniques, Challenges and Future Directions,” *arXiv preprint arXiv: 2402.01968*, Feb. 2024. DOI: 10.48550/arXiv.2402.01968.
- [2] C. Qian, W. Liu, H. Liu, N. Chen, Y. Dang, J. Li, C. Yang, W. Chen, Y. Su, X. Cong et al., “ChatDev: Communicative Agents for Software Development,” *arXiv preprint arXiv: 2307.07924*, Jul. 2023. DOI: 10.48550/arXiv.2307.07924.
- [3] GitHub. crewAIInc/crewAI [Internet]. Available: <https://github.com/crewAIInc/crewAI>.
- [4] GitHub. langchain-ai/langgraph [Internet]. Available: <https://github.com/langchain-ai/langgraph>.
- [5] GitHub. Significant-Gravitas/AutoGPT [Internet]. Available: <https://github.com/Torantulino/Auto-GPT>.
- [6] P. Venkadesh, S. V. Divya, and K. S. Kumar, “Unlocking AI Creativity: A Multi-Agent Approach with CrewAI,” *Journal of Trends in Computer Science and Smart Technology*, vol. 6, no. 4, pp. 338-356, Dec. 2024. DOI: 10.36548/jtcst.2024.4.002.
- [7] J. Wang and Z. Duan, “Agent AI with LangGraph: A Modular Framework for Enhancing Machine Translation Using Large Language Models,” *arXiv preprint arXiv: 2412.03801*, Dec. 2024. DOI: 10.48550/arXiv.2412.03801.
- [8] A. Singh, A. Ehtesham, S. Kumar, and T. Talaei Khoei, “Agentic Retrieval-Augmented Generation: A Survey on Agentic RAG,” *arXiv preprint arXiv:2501.09136*, Jan. 2025. DOI: 10.48550/arXiv.2501.09136.
- [9] Y. Shao, Y. Jiang, T. A. Kanell, P. Xu, O. Khatab, and M. S. Lam, “Assisting in Writing Wikipedia-like Articles From Scratch with Large Language Models,” *arXiv preprint arXiv:2402.14207*, Feb. 2024. DOI: 10.48550/arXiv.2402.14207.
- [10] H. Li, Q. Dong, J. Chen, H. Su, Y. Zhou, Q. Ai, Z. Ye, and Y. Liu, “LLMs-as-Judges: A Comprehensive Survey on LLM-based Evaluation Methods,” *arXiv preprint arXiv:2412.05579*, Dec. 2024. DOI: 10.48550/arXiv.2412.5579.



김재호(Jae-Ho Kim)

수원대학교 컴퓨터학부 학사  
수원대학교 컴퓨터학부 석사  
수원대학교 컴퓨터학부 박사과정  
Langchain Opentutorial Core Contributor  
※관심분야 : 인공지능, LLM, RAG, AI Agent



김장영(Jang-Young Kim)

연세대학교 컴퓨터과학 공학사  
Pennsylvania State Univ. 공학석사  
State University of New York 공학박사  
University of South Carolina 교수  
수원대학교 컴퓨터학부 교수  
※관심분야 : Big data, AI, Cloud computing, Networks