WCAA

🔧 Architecture Report

World-Centric Agent Architecture:

Manifesto 기반 결정론적 런타임과 계층화된 지능 오케스트레이션

(Draft v0.2)

초록 (Abstract)

최근 LLM 기반 에이전트 시스템은 추론 능력의 확장에도 불구하고, 장기 실행 안정성, 재현성, 그리고 실패 원인의 설명 가능성 측면에서 반복적인 한계를 드러내고 있다. 본 연구는 이러한 문제의 근본 원인이 모델의 지능 부족이 아니라, **세계(World)와 상태 변화가 암묵적으로 처리되는 기존 에이전트 아키텍처의 구조적 결함**에 있음을 주장한다.

우리는 이를 해결하기 위해 **World-Centric Agent Architecture**를 제안한다. 제안하는 구조는 Manifesto Core라는 결정론적 상태 엔진을 중심으로, 세계를 불변의 Snapshot과 명시적인 Patch/Apply 메커니즘으로 관리하며, 지능을 실행 주체가 아닌 **제안자(proposer)** 로 재배치한다. 또한 Student/Teacher/Orchestrator의 계층적 분리를 통해, 실행, 모델링, 그리고 세계 전이를 명확히 구분한다.

본 논문은 이 아키텍처가 소형 모델 및 단순한 행동 선택기에서도 안정적으로 작동함을 보이며, 이를 통해 **지능과 세계 모델링을 분리하는 새로운 에이전트 설계 관점**을 제시한다.

---

1. 서론 (Introduction)

LLM 기반 에이전트는 다양한 환경에서 인간과 유사한 문제 해결 능력을 보이지만, 동시에 다음과 같은 구조적 문제를 반복적으로 노출해왔다.

- 불가능한 행동의 반복 시도
- 상태 변경의 원인 불투명성
- 실행 중 구조 변경으로 인한 재현 불가
- 실패 후 복구 불가능성

기존 연구는 이러한 문제를 주로 모델 추론 능력의 부족으로 해석해왔다. 이에 따라 더 큰 모델, 더 긴 reasoning, 더 복잡한 planning loop가 제안되어 왔다.

그러나 본 연구는 다음 질문을 제기한다.

> 이 문제들은 정말로 지능의 문제인가,
> 아니면 세계와 상태 변화가 설계되지 않은 결과인가?

---

2. 기존 에이전트 아키텍처의 한계

2.1 암묵적 세계 모델

대부분의 에이전트 시스템은 세계의 규칙과 제약을 모델 내부 추론에 위임한다. 행동 가능성(action feasibility), 상태 전이의 정당성, 그리고 실패 원인은 명시적으로 표현되지 않는다.

이로 인해 다음과 같은 문제가 발생한다.

- 모델은 불가능한 행동을 추론으로 걸러야 한다
- 실패 시, 왜 실패했는지 구조적으로 설명할 수 없다
- 상태 변경이 언제, 왜 일어났는지 추적하기 어렵다

2.2 지능 중심 설계의 문제

이러한 구조에서는 지능이 다음 역할을 동시에 수행한다.

- 세계 이해
- 행동 선택
- 상태 변경 판단

이는 비결정적이고 오류를 내포한 LLM에게 과도한 책임을 부여하며, 시스템 안정성을 근본적으로 약화시킨다.

---

3. World-Centric Agent Architecture 개요

본 연구는 세계를 에이전트의 내부가 아닌, **1급 객체**로 취급하는 World-Centric 설계를 제안한다.

3.1 핵심 원칙

제안하는 아키텍처는 다음 원칙을 따른다.

1. 세계(World)는 명시적으로 모델링된다
2. 모든 상태는 불변 Snapshot으로 표현된다
3. 상태 변화는 Patch/Apply를 통해서만 발생한다
4. 지능은 상태를 변경하지 않고 제안만 수행한다
5. 실패는 삭제되지 않고 기록된다

---

4. Manifesto Core: 결정론적 세계 엔진

4.1 Snapshot과 Truth

Snapshot은 특정 시점의 세계를 나타내는 불변 객체다. 세계의 진실(Truth)은 오직 Snapshot으로만 표현되며, 직접 변경될 수 없다.

4.2 Patch / Apply 메커니즘

상태 변화는 Patch로 선언되고, Apply를 통해 새로운 Snapshot을 생성한다. 이로 인해 모든 변화는 다음 속성을 가진다.

- 결정론적
- 재현 가능
- 변경 경로 추적 가능

4.3 Action과 Availability

Action은 실행 가능한 선택지를 정의하며, availability는 세계의 계산된 사실(Computed Fact)을 참조한다. 불가능한 행동은 실행 단계에 도달하지 않는다.

---

5. 계층화된 지능: Student / Teacher / Orchestrator

5.1 Student: 세계 내부 실행 주체

Student는 현재 World 안에서 가능한 Action 중 하나를 선택한다. Student는 판단이나 추론을 요구받지 않으며, 단순한 선택기(random selector)로도 충분하다.

5.2 Teacher: 세계 외부 모델링 주체

Teacher는 실패 로그와 실행 기록을 관찰하고, 새로운 World 가설을 제안한다. Teacher는 상태를 변경하지 않으며, 실행 권한을 갖지 않는다.

5.3 Orchestrator: 세계 전이 관리

Orchestrator는 여러 World 후보를 fork 구조로 관리한다. 실패한 World는 보존되며, 성공한 World만 선택적으로 채택된다.

---

6. 지능의 재배치와 시스템 안정성

본 아키텍처의 핵심은 지능의 역할을 다음과 같이 재배치한 점이다.

- 지능은 실행하지 않는다
- 지능은 판단하지 않는다
- 지능은 세계를 제안할 뿐이다

이로 인해 지능의 크기와 무관하게 시스템의 안정성은 유지된다. 소형 모델, 대형 모델, 또는 인간 개입 모두 동일한 구조 안에서 작동한다.

---

7. 오라클 논란에 대한 짧은 논의

Teacher는 정답이나 숨겨진 상태에 접근하지 않으며, 결과를 보장하지 않는다. Teacher의 출력은 항상 가설이며, 모든 가설은 실행을 통해 독립적으로 검증된다.

따라서 Teacher는 oracle이 아니라, **틀릴 수 있는 모델링 제안자**로 정의된다.

---

8. 사례: LLM-BabyBench

본 아키텍처는 LLM-BabyBench와 같은 환경에서 효과적으로 작동한다. 특히, 행동 가능성을 세계의 사실로 외부화함으로써, 불가능한 행동 시도를 구조적으로 제거한다.

이는 모델 추론 능력을 향상시키지 않고도 안정적인 수행을 가능하게 한다.

---

9. 논의 (Discussion)

본 연구는 에이전트 성능 향상이 반드시 지능의 확장을 의미하지 않음을 보여준다. 많은 실패는 지능 부족이 아니라, 세계가 설계되지 않았기 때문이다.

World-Centric 설계는 에이전트 연구를 지능 중심에서 **시스템 중심**으로 전환할 필요성을 제시한다.

---

10. 결론 (Conclusion)

본 논문은 World-Centric Agent Architecture를 통해, 에이전트 시스템의 안정성, 재현성, 그리고 설명 가능성을 구조적으로 확보할 수 있음을 보였다. 이는 AGI 담론과 무관하게 장기적으로 운영 가능한 에이전트 시스템을 설계하는 하나의 방향을 제시한다.

핵심 기여 요약

1. 세계를 1급 객체로 다루는 World-Centric 설계 제안
2. 결정론적 Snapshot / Patch / Apply 기반 상태 관리
3. Student / Teacher / Orchestrator 분리를 통한 지능 재배치
4. 지능 크기와 무관한 시스템 안정성 확보