3. 계층적/재귀적 Agent 설계

계층과 재귀

- MAS의 성능을 개선하고 진화시키기 위한 고민
  - 단순한 Agent들의 집합을 넘어, 어떻게 거대한 문제를 해결하는 유기적인 팀을 구축할 것인가
  - 어떻게 Agent가 실수를 스스로 교정하고 결과물의 품질을 지속적으로 향상시키게 할 것인가
- Agent의 추론 능력과 협업 패턴을 결합한 두 가지 구조
  - 계층적 설계 (Hierarchical Design)
    - 복잡성을 관리하기 위한 분할과 정복(Divide and Conquer) 전략
  - 재귀적 설계 (Recursive Design)
    - 품질을 극대화하기 위한 반복적 개선(Iterative Refinement) 전략

계층 구조 – 1 Planner-Executor

- Planner-Executor - 지휘자와 연주자들
  - 복잡한 문제를 해결하기 위한 가장 고전적이고 효과적인 전략
  - Planner Agent (지휘자)
    - 고수준의 전략가
    - 전체 목표를 분석하여 다단계 계획을 수립하고, 각 단계를 적절한 실행자에게 할당
  - Executor Agent (연주자)
    - 특정 분야의 전문가
    - Planner로부터 할당받은 구체적인 단일 작업을 수행하는 데 집중

계층 구조 – 2 Delegation

- 하위 Agent 위임 (Delegation)
  - 상위 Agent가 특정 전문성이 필요한 복합적인 작업을 해당 능력을 갖춘 하위 Agent에게 통째로 위임하는 고도화된 협업 방식
  - Tool calling이 아닌 Agent calling
    - 복잡한 추론과 여러 단계의 Tool 사용을 포함하는 작업 전체를 위임
    - 상위 Agent는 하위 Agent의 복잡한 내부 동작을 알 필요 없이, 입력과 출력에만 집중 가능
    - 잘 만들어진 전문 Agent는 다른 여러 상위 Agent에 의해 재사용될 수 있음

재귀 구조 – Self Correction

- Self-Correction
  - Agent가 생성한 결과물을 스스로 평가하고 개선하는 재귀적(Recursive) 프로세스
  - 단 한 번에 완벽한 결과물을 만들려는 시도가 아닌, 인간 전문가처럼 초안을 만들고 반복적으로 퇴고하며 품질을 높여가는 방식
- 루프
  - 생성 -> 평가 -> 수정

재귀 구조 – Self Correction

- 루프
  - 생성(Generate)
    - Agent가 현재 지식을 바탕으로 초기 결과물(초안)을 생성
  - 평가(Critique)
    - 생성된 결과물이 목표 기준을 만족하는지 비판적으로 검토
    - 가장 중요한 과정
  - 수정(Refine)
    - 평가 단계에서 발견된 문제점과 개선 피드백을 바탕으로 결과물을 수정하고
    - 다시 평가 단계로 돌아가거나 루프를 종료

재귀 구조 – Self Correction

- 평가(Critique) 메커니즘
  - 평가 단계의 품질이 Self-Correction 루프 전체의 성패를 좌우
  - 무엇이 잘못되었고, 어떻게 개선해야 하는지에 대한 구체적인 피드백 제공
- 크게 두 가지 방법을 사용
  - 규칙 기반 체크리스트 (Rule-based Checklist)
  - LLM-as-Judge

재귀 구조 – Self Correction

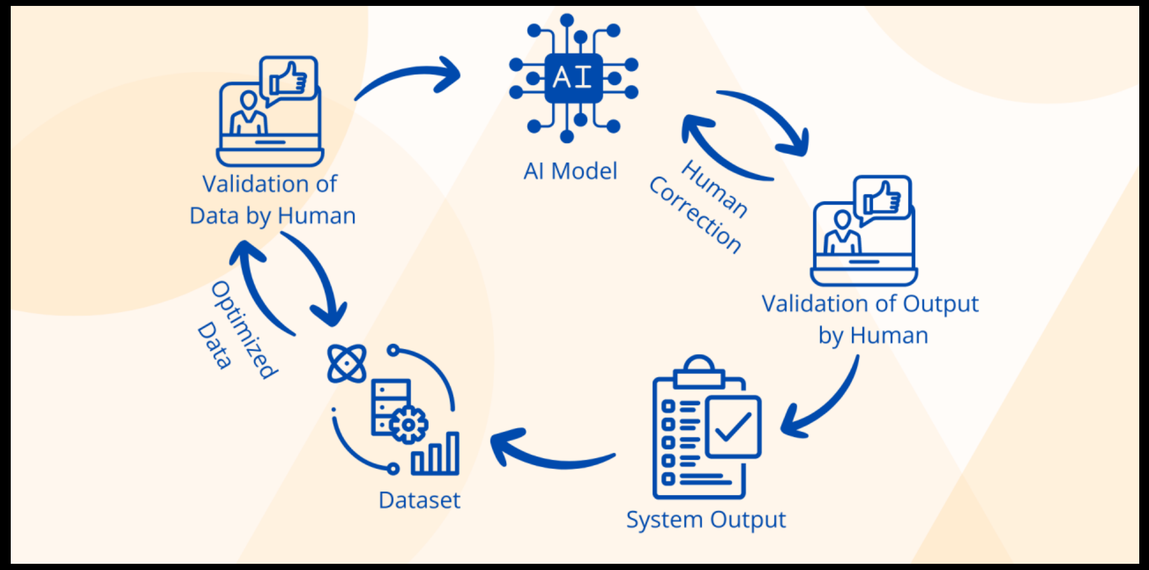
- 규칙 기반 체크리스트 (Rule-based Checklist)
  - 사전에 정의된 명시적인 규칙이나 체크리스트를 기반으로 결과물을 평가
  - 객관식 항목으로 구성하는 것이 좋음
    - 평가 항목의 수가 많을 수록
    - 평가 기준이 단순할 수록 좋음(정성 평가 < 척도 평가 < 여부 체크)
  - 예시
    - 글자 수는 1,000자 이상인가?(양 측정)
    - 주장에 대한 외부 출처 인용이 포함되었는가?(구조 평가)
    - 지정된 톤앤매너(예: 전문적, 유머러스)를 준수했는가?(스타일 평가)

재귀 구조 – Self Correction

- LLM-as-Judge
  - 결과물 평가 작업 자체를 다른 LLM에게 위임하는 방식
    - 평가 전문 LLM(Judge)이 미리 정의된 평가 기준(Rubric)에 따라 결과물의 품질을 채점
    - 구체적인 개선 피드백을 생성
  - G-Eval의 아이디어를 활용
    - 사실 상 규칙 기반 평가를 LLM이 수행하는 것 + 약간의 정성평가
    - 모든 규칙은 프롬프트로 주입
  - 주의 사항
    - 결과물을 생성한 Agent와 평가하는 Judge Agent는 서로 다른 LLM을 활용해야 자기 편향을 막을 수 있음

HITL

- HITL(Human-in-the-loop)
  - Agentic 자동화 파이프라인의 Cherry-on-top
  - 루프의 특정 지점에 사람의 개입을 허용
  - 시스템의 최종 결정 품질을 높이고 Agent가 사람의 의도를 더 잘 학습하도록 유도하는 기법
  - 가장 궁극적인 가치 판단/평가 기준에 인간의 선호도와 의도를 넣을 수 있음



https://www.linkedin.com/pulse/responsible-ai-use-human-in-the-loop-hitl-anderson-anthony