Agent Study Week1

2025. 02. 10

이혜승

Contents

- ❖ The orchestration layer
- ❖ Agents vs. models
- ❖ Cognitive architectures: How agents operate

What is an agent?

❖ General agent architecture and components

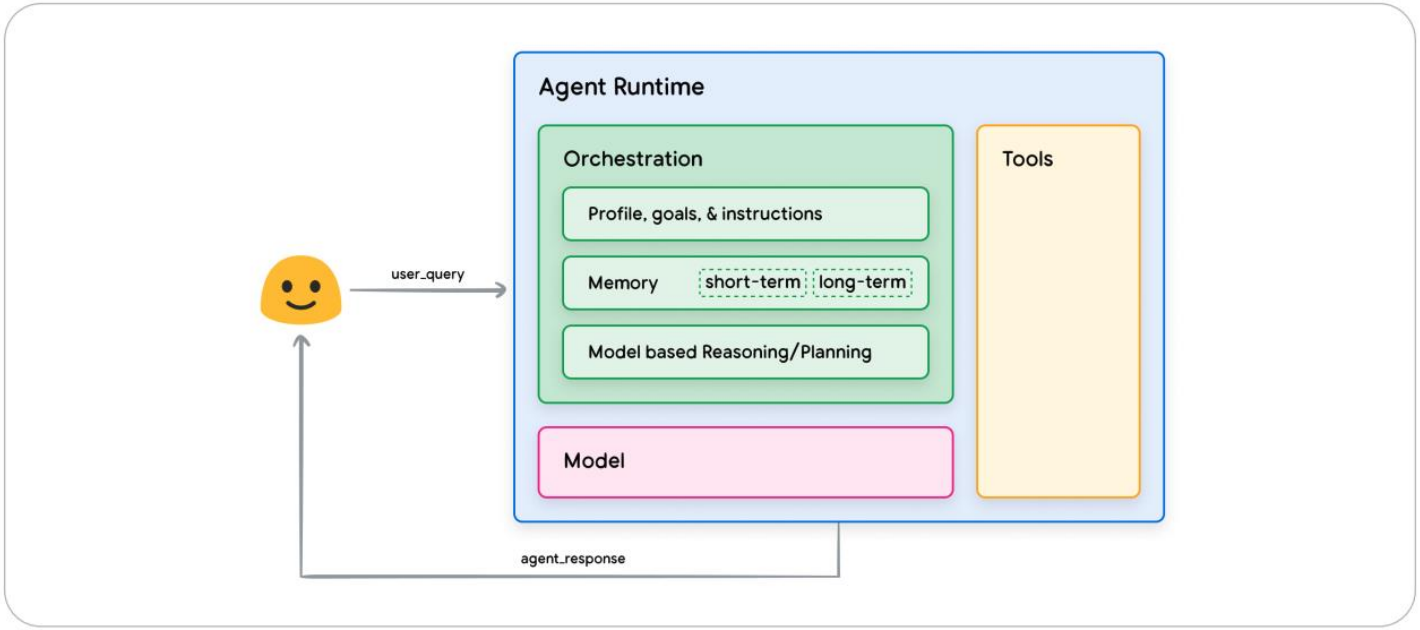
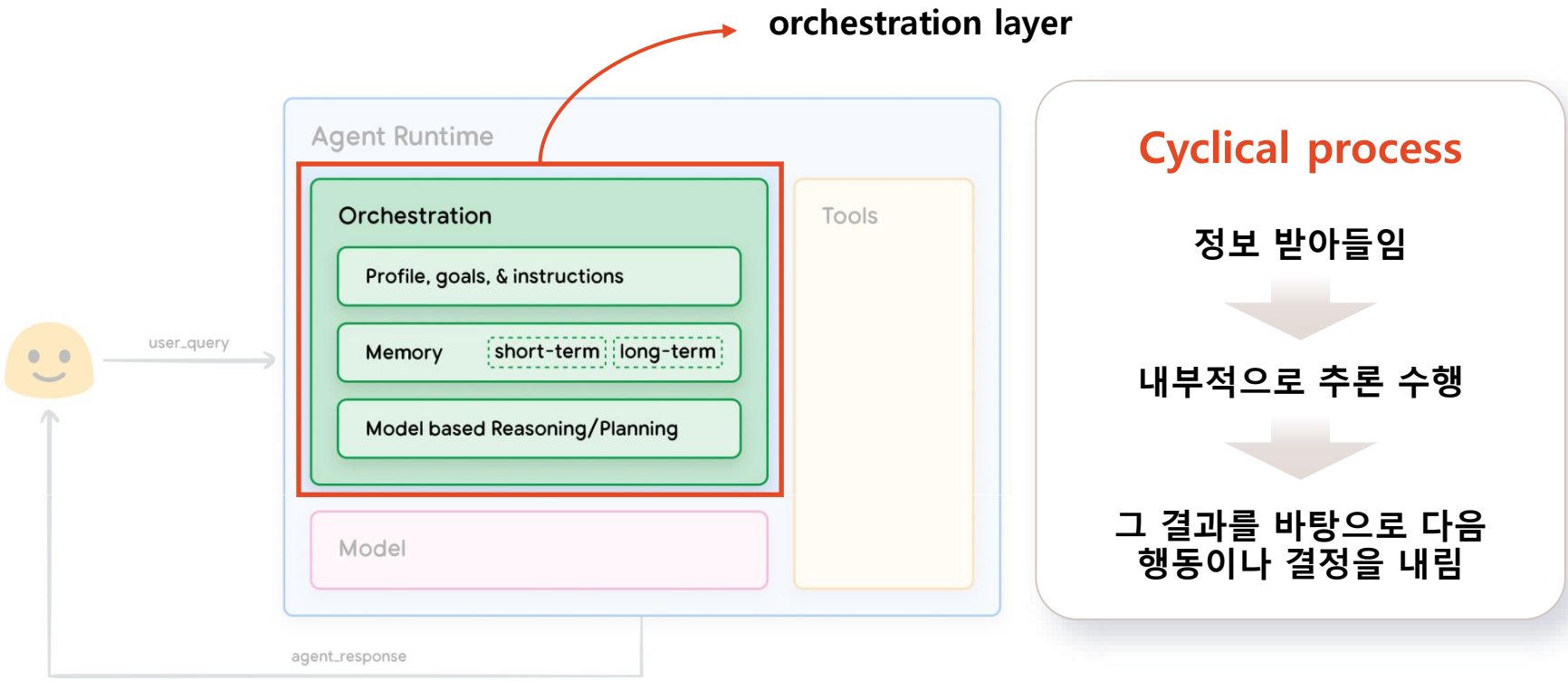


Figure 1. General agent architecture and components

The orchestration layer

❖ 오케스트레이션 계층

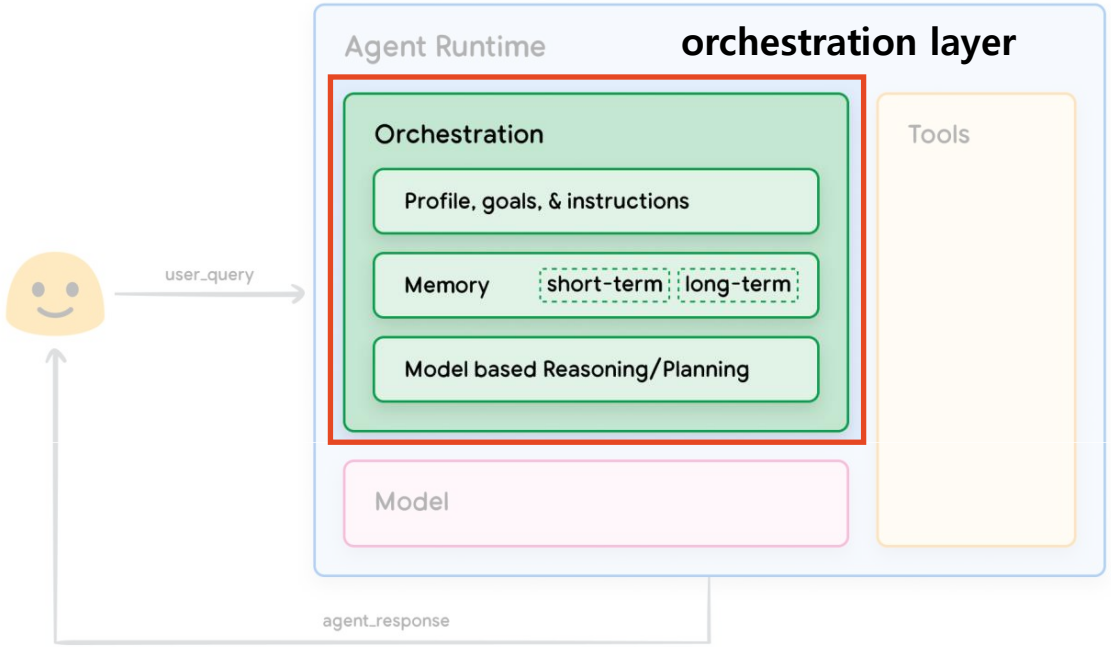
- Cyclical process를 에이전트가 **목표를 달성하거나 정해진 종료 지점**에 도달할 때까지 반복



The orchestration layer

❖ 오케스트레이션 계층의 복잡성

- 에이전트의 **유형 및 수행하는 작업**에 따라 크게 달라짐
- 단순한 경우: 의사 결정 규칙을 기반으로 간단한 계산 수행하는 루프
- 복잡한 경우: 연쇄적 논리, 추가적인 머신러닝 알고리즘 등을 포함하는 루프



Agents vs. models

❖ 에이전트와 모델의 차이

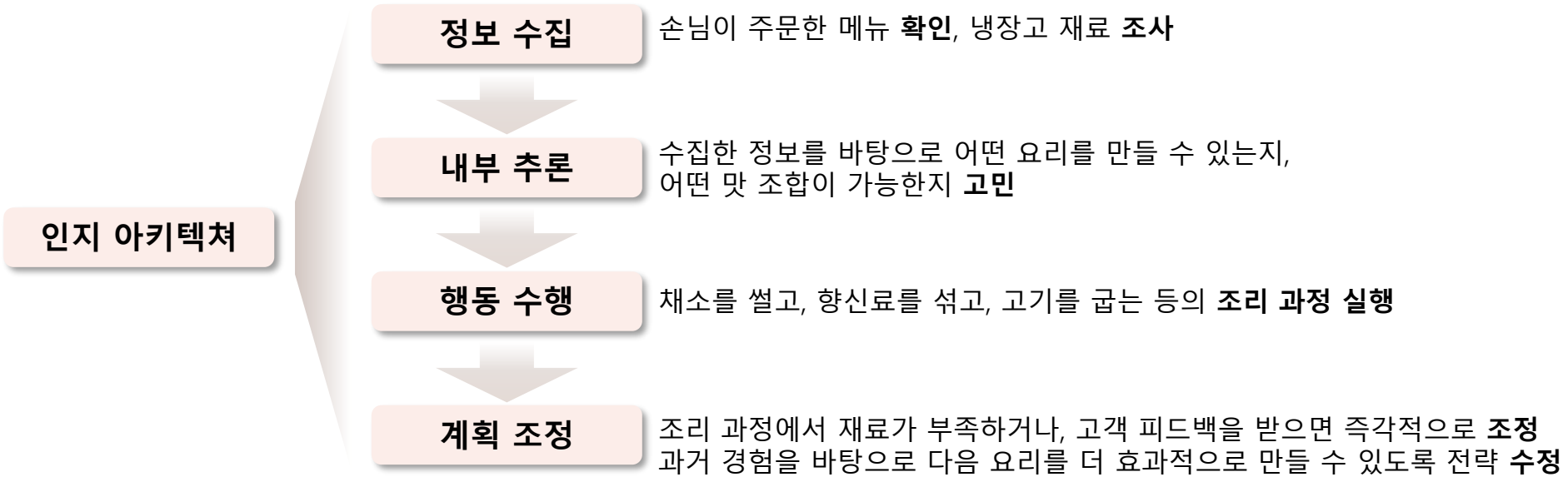
- 모델: 단순히 학습된 데이터에서 추론 수행
- 에이전트: 도구와 외부 시스템을 활용하여 더욱 확장된 기능 제공

	Models	Agents
지식 범위	학습 데이터에 포함된 지식만 활용 가능	외부 시스템과 도구(tools) 연결을 통해 지식을 확장
컨텍스트 및 세션 관리	단일 추론/예측 수행. 연속적 관리 X	세션 내역을 관리하여 여러 차례(turn-based) 추론/예측 가능. 오케스트레이션 계층을 통해 사용자 질의 및 의사 결정 기반으로 지속적 상호작용 수행
도구 활용	사용 X	사용 O
추론 및 논리 계층	논리 계층(logic layer) 구현 X	추론 프레임워크(CoT, ReAct 등)를 활용하는 네이티브 cognitive architecture를 포함, 에이전트 프레임워크와 연동 가능

Cognitive architectures: How agents operate

❖ 인지 아키텍처

- 예시) 바쁜 주방의 요리사
- 목표: 맛있는 요리 만들어 고객에게 제공, 이를 위해 계획 실행 조정의 반복적인 과정 수행



Cognitive architectures: How agents operate

❖ **Agent 인지 아키텍처**

- **요리사처럼, 에이전트도 인지 아키텍쳐를 활용하여 목표 달성**

Agent 인지 아키텍쳐의 역할

✅ **메모리 및 상태(state) 유지**

✅ 추론(reasoning) 및 계획(planning) 관리

✅ 프롬프트 엔지니어링(prompt engineering)과 관련 프레임워크 활용

📌 **예제:**

💬 **사용자:** "서울 날씨 어때?"

🤖 **Agent:** "서울의 현재 기온은 15도입니다."

💬 **사용자:** "내일은?"

🤖 **Agent:** "내일 서울의 예상 기온은 18도입니다." (이전 질문을 기억하여 자연스럽게 응답)

Cognitive architectures: How agents operate

❖ **Agent 인지 아키텍처**

- **요리사처럼, 에이전트도 인지 아키텍쳐를 활용하여 목표 달성**

Agent 인지 아키텍쳐의 역할

✅ 메모리 및 상태(state) 유지

✅ **추론(reasoning) 및 계획(planning) 관리**

✅ 프롬프트 엔지니어링(prompt engineering)과 관련 프레임워크 활용

📌 **예제:**

💬 **사용자:** "12명의 학생을 3개 조로 나누고 싶어. 한 조에 몇 명씩 들어가?"

🤖 **일반 LLM:** "4명."

🤖 **Agent (CoT 활용):**

- **Step 1:** 총 학생 수는 12명입니다.
- **Step 2:** 조의 개수는 3개입니다.
- **Step 3:** 12 ÷ 3 = 4
- **결론:** 따라서 한 조당 4명씩 배정할 수 있습니다.

Cognitive architectures: How agents operate

❖ **Agent 인지 아키텍처**

- **요리사처럼, 에이전트도 인지 아키텍쳐를 활용하여 목표 달성**

Agent 인지 아키텍쳐의 역할

✅ 메모리 및 상태(state) 유지

✅ 추론(reasoning) 및 계획(planning) 관리

✅ **프롬프트 엔지니어링(prompt engineering)과 관련 프레임워크 활용**

📌 **예제:**

💬 **사용자:** "비트코인 현재 가격은 얼마야?"

🤖 **일반 LLM:** "죄송합니다. 저는 실시간 데이터를 제공할 수 없습니다."

🤖 **Agent (RAG + API 활용):**

- **Step 1:** CoinMarketCap API를 호출하여 최신 가격 조회
- **Step 2:** 검색된 데이터를 기반으로 응답 생성
- **Step 3:** "현재 비트코인 가격은 45,000달러입니다."

Cognitive architectures: How agents operate

❖ Agent 인지 아키텍처

- 요리사처럼, 에이전트도 인지 아키텍쳐를 활용하여 목표 달성

Agent 인지 아키텍쳐의 역할

✅ 메모리 및 상태(state) 유지

✅ **추론(reasoning) 및 계획(planning) 관리**

✅ 프롬프트 엔지니어링(prompt engineering)과 관련 프레임워크 활용

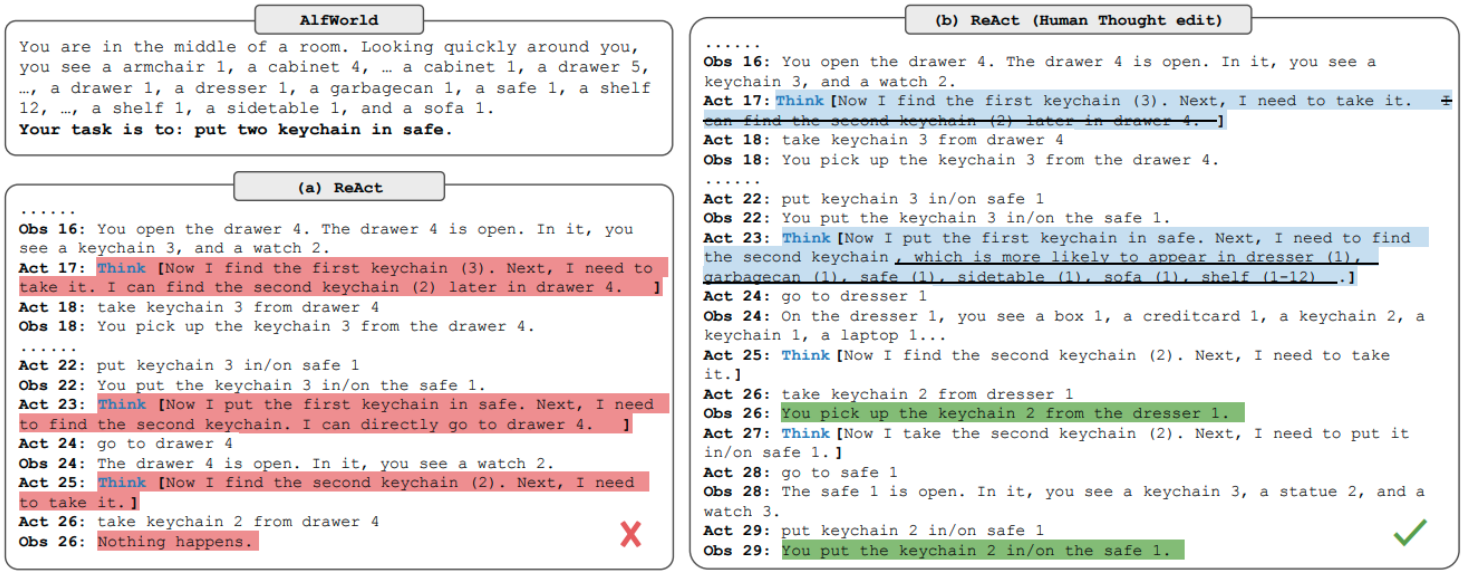
❖ 대표적인 Reasoning Techniques

- ReAct (Reasoning + Acting)
- Chain-of-Thought (CoT)
- Tree-of-Thoughts (ToT)

Reasoning Techniques

❖ ReAct (Reasoning + Acting)

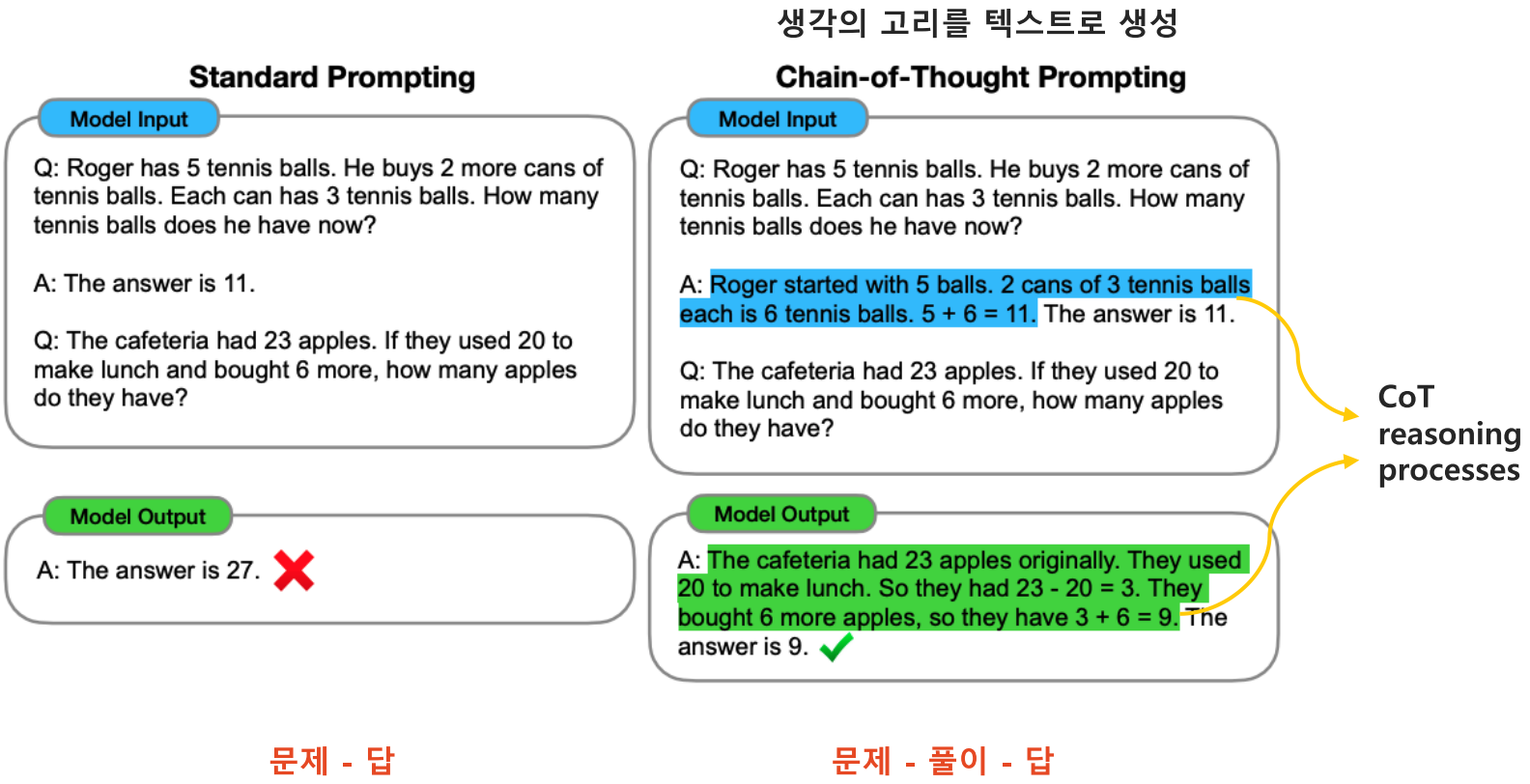
- 언어 모델이 사용자 질의에 대해 논리적으로 추론하고, 필요한 조치를 수행할 수 있도록 하는 프롬프트 엔지니어링 기법
- 맥락 내 예제(in-context examples) 유무에 관계없이 동작 가능
- 여러 최신 SOTA 모델보다 뛰어난 성능을 보이며, 인간 상호 운용성과 신뢰성 향상



Reasoning Techniques

❖ Chain-of-Thought(CoT)

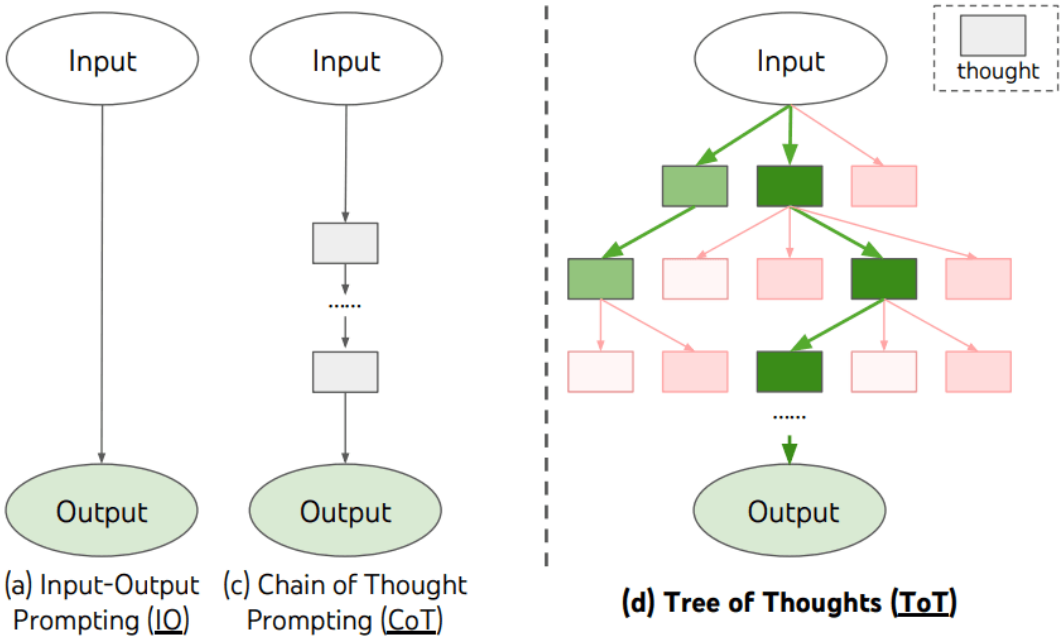
- 추론을 여러 중간 단계를 거쳐 수행할 수 있도록 prompting



Reasoning Techniques

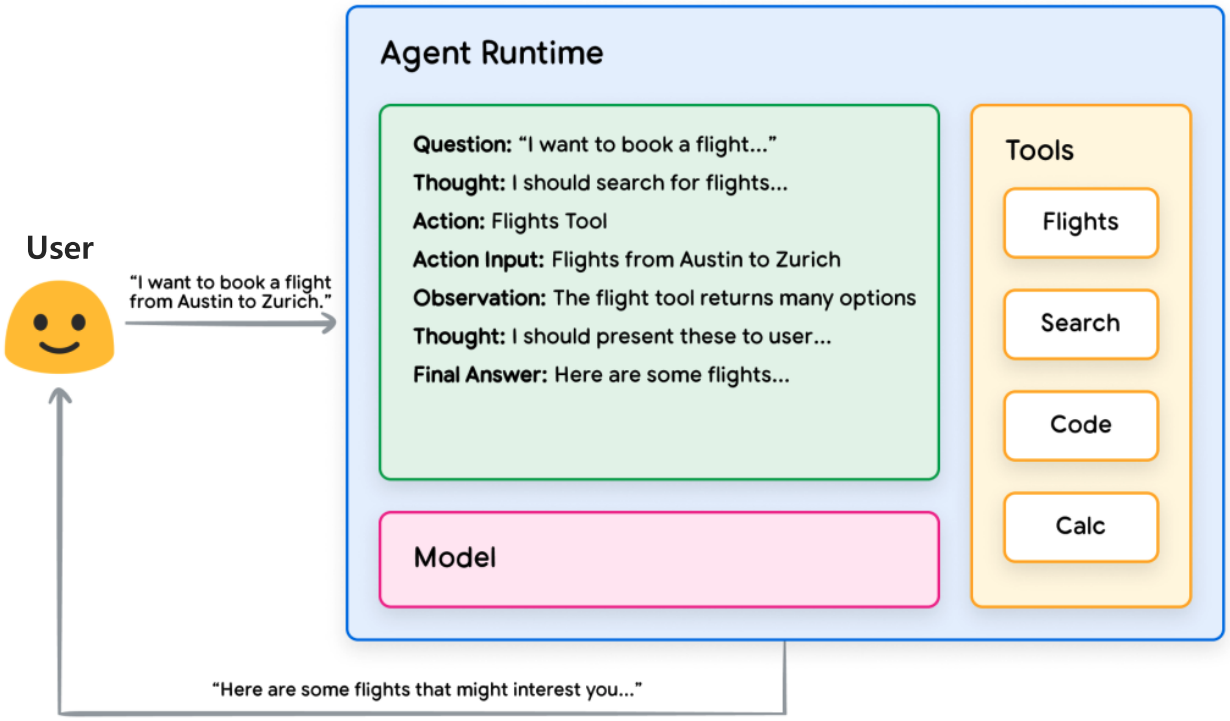
❖ Tree-of-Thoughts (ToT)

- 문제 해결 과정에서 탐색과 전략적 예측을 수행하는 데 적합한 프롬프트 엔지니어링 기법
- CoT를 확장한 개념
  - ➢ **여러 개의 사고 흐름을 탐색**
  - ➢ LLM이 다양한 문제 해결 방안을 고려하고 최적의 해결책을 찾을 수 있도록 지원



ReAct Framework Example

❖ Example agent with ReAct reasoning in the orchestration layer

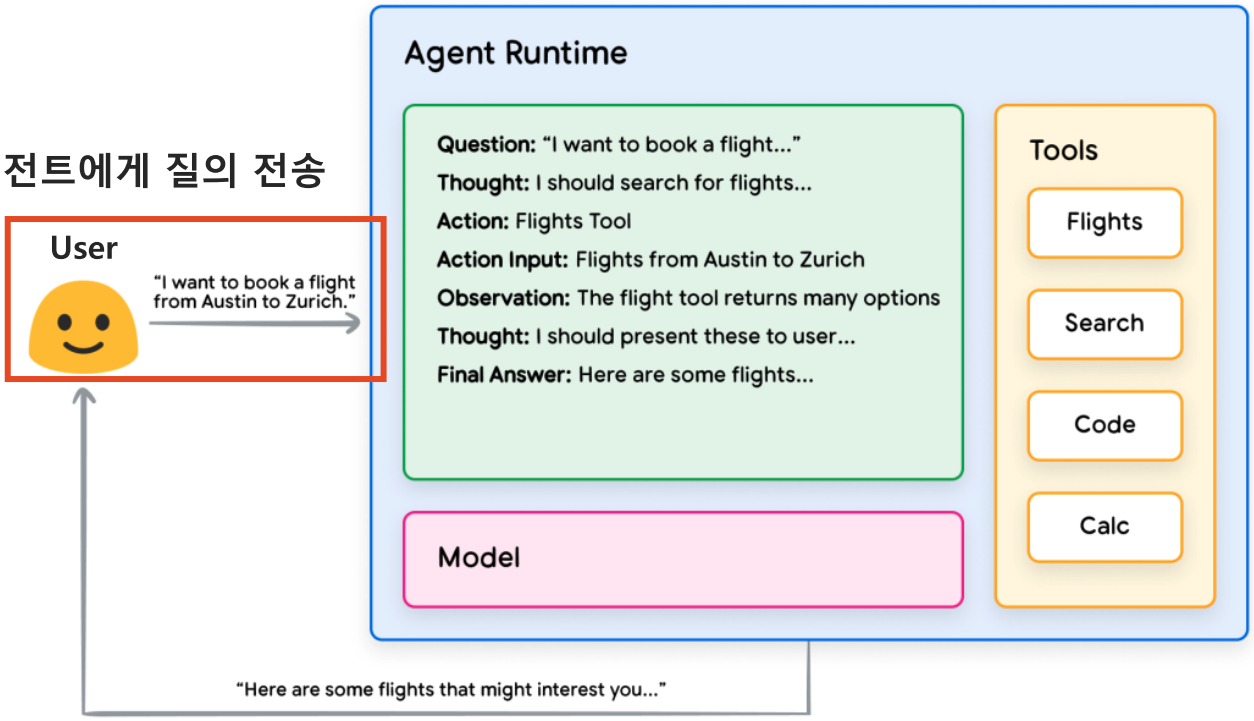


ReAct Framework Example

❖ Example agent with ReAct reasoning in the orchestration layer

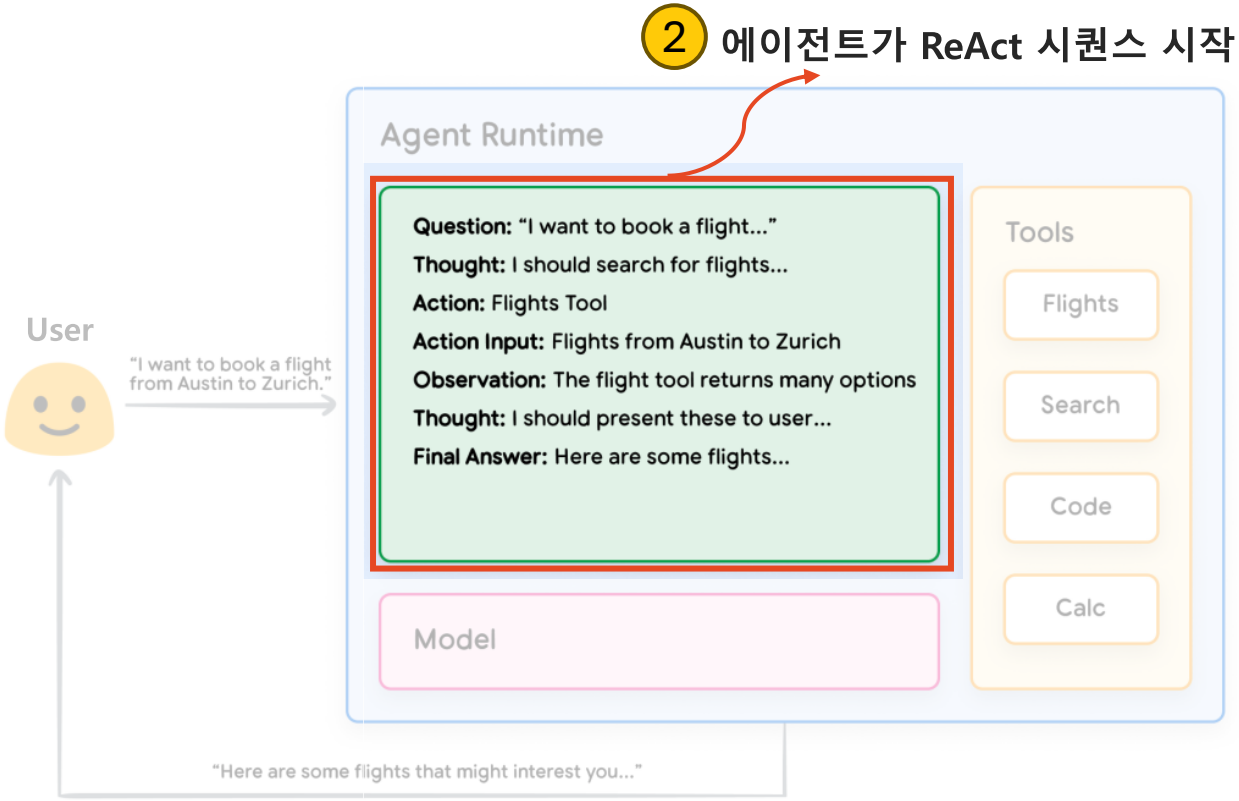
1

사용자가 에이전트에게 질의 전송



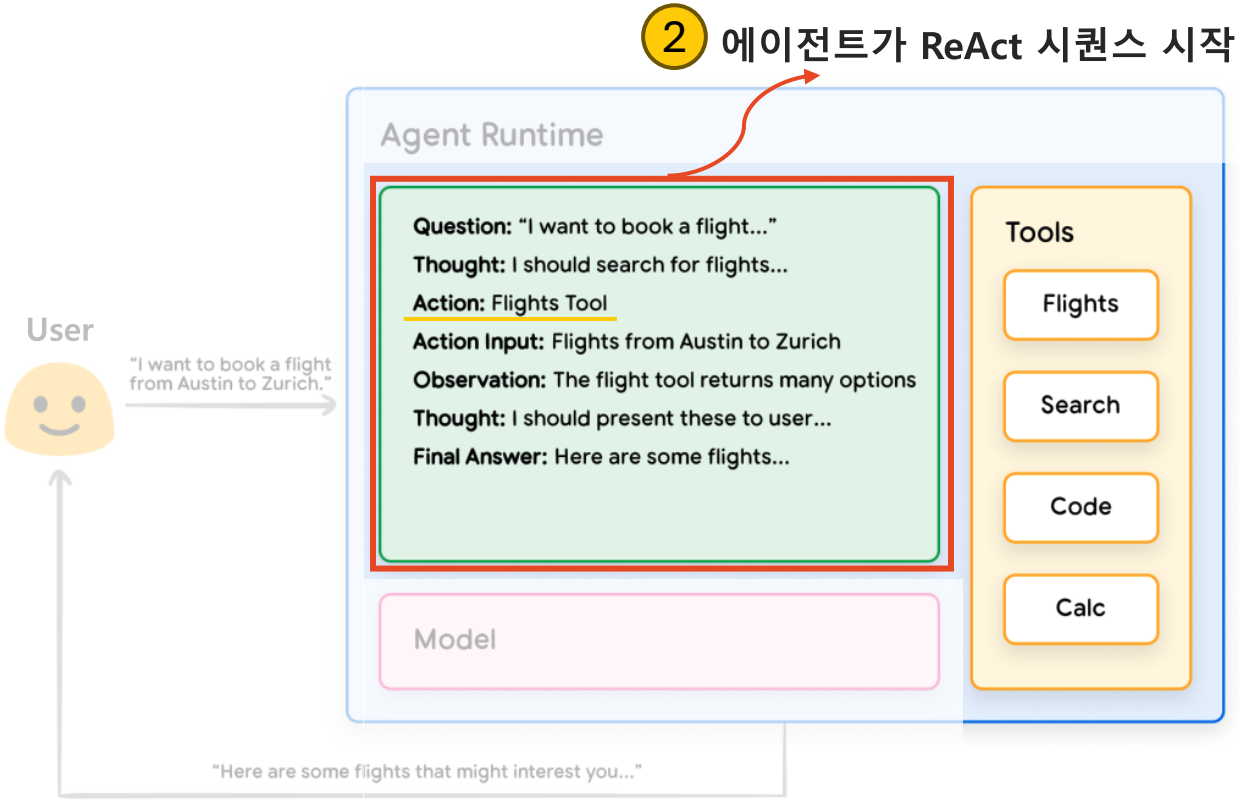
ReAct Framework Example

❖ Example agent with ReAct reasoning in the orchestration layer



ReAct Framework Example

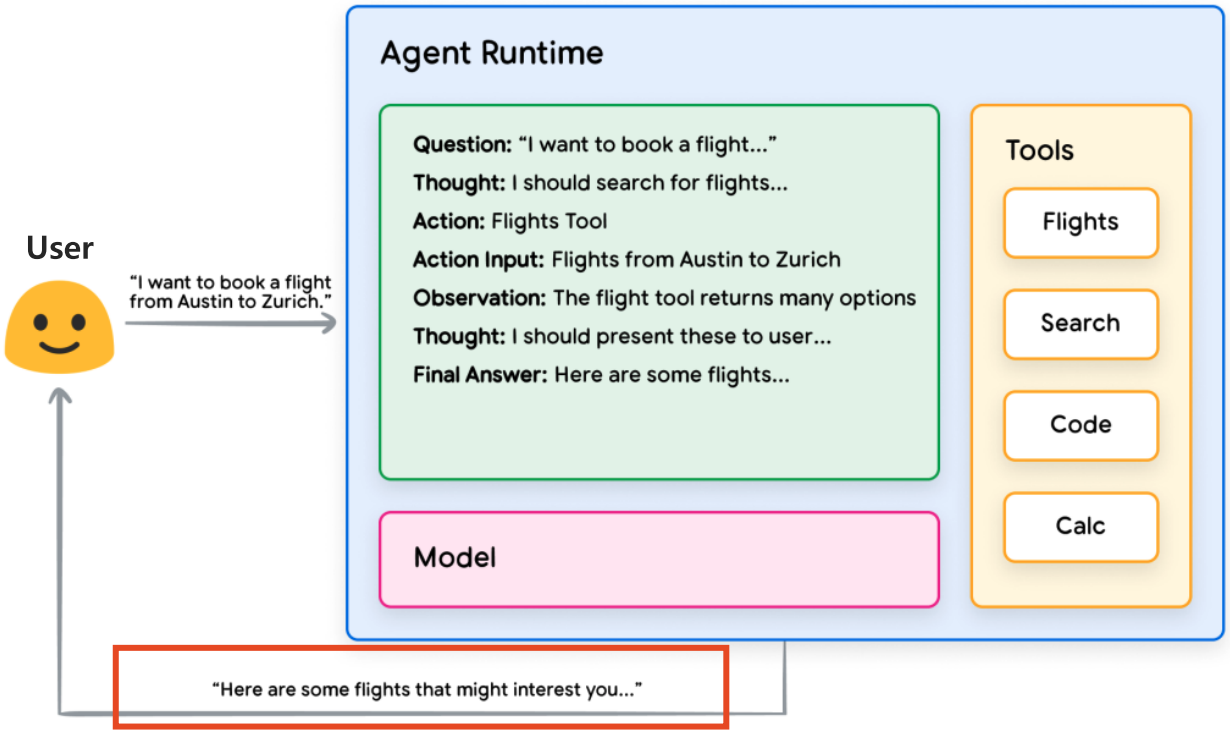
❖ Example agent with ReAct reasoning in the orchestration layer



ReAct Framework Example

❖ Example agent with ReAct reasoning in the orchestration layer

- Agent가 단순히 추측(hallucination)으로 응답하지 않고, 신뢰할 수 있는 정보를 기반으로 답변 생성



3 ReAct 루프가 종료되며 최종 답변 제공

Summary

❖ **Agent의 응답 품질은 다음 요소에 의해 결정됨**

- 모델의 추론 능력
- 적절한 도구 선택
- 도구의 정의 및 활용 방식

❖ **다음 시간**

- Agent가 최신 데이터를 활용하는 다양한 방법