Lecture 8: AI Agents Orchestration

Mia Mohammad Imran

So far we talked about

- Prompt engineering
- RAG-based grounding
- Agentic AI design patterns
- CLI-based autonomous agents
- How to build a personalized agent

So far we talked about

- Prompt engineering
- RAG-based grounding
- Agentic AI design patterns
- CLI-based autonomous agents
- How to build a personalized agent

Now we move to a higher level abstraction:

> *How do we coordinate multiple agents, tools, memory systems, and control loops into a coherent system?*

This coordination layer is called AI Agent Orchestration.

In traditional systems engineering: what does orchestration refer to?

Assume **Uber Eats**

When a user clicks **“Place Order”**, how many things happen behind the scenes?

In traditional systems engineering: what does orchestration refer to?

Assume **Uber Eats:** When a user clicks **“Place Order”**, how many things happen behind the scenes?

1. Validate cart contents
2. Check restaurant availability
3. Reserve items in inventory
4. Process payment
5. Create order record in database
6. Notify restaurant
7. Send confirmation to user
8. If payment fails:
    a. Cancel reservation
    b. Do not create order
    c. Return error to user

What Is Orchestration Here?

- Decides execution order
- Coordinates multiple services
- Handles conditional logic
- Performs rollback if needed
- Ensures consistency

Each service does one job:

- Payment service → processes payment
- Inventory service → checks stock
- Notification service → sends alerts

The controller coordinates them (this coordination is software orchestration)

What Is AI Agent Orchestration?

AI Agent Orchestration is the structured coordination of:

- Multiple agents
- Tools and APIs
- Memory and knowledge sources
- Planning and execution loops
- Verification and reflection stages to accomplish complex tasks reliably

It answers:

- Who does what?
- In what order?
- With what information?
- Under what constraints?
- When should we stop or retry?

Orchestration transforms isolated agents into **systems**

https://www.ibm.com/think/topics/ai-agent-orchestration

Mapping Traditional → AI (conceptual, not direct mapping)

Traditional Software Engineering	AI Agent System
Controller	Orchestrator
Microservices	Specialized Agents
Database	Memory / RAG
Payment API	Tool Call
Health check	Test run
Rollback	Retry / Reflection

From Single Agent to Orchestrated System

Level 1: Prompting

- Single request ⟶ single response

Level 2: RAG-Augmented Prompt

- Retrieve ⟶ inject context ⟶ generate

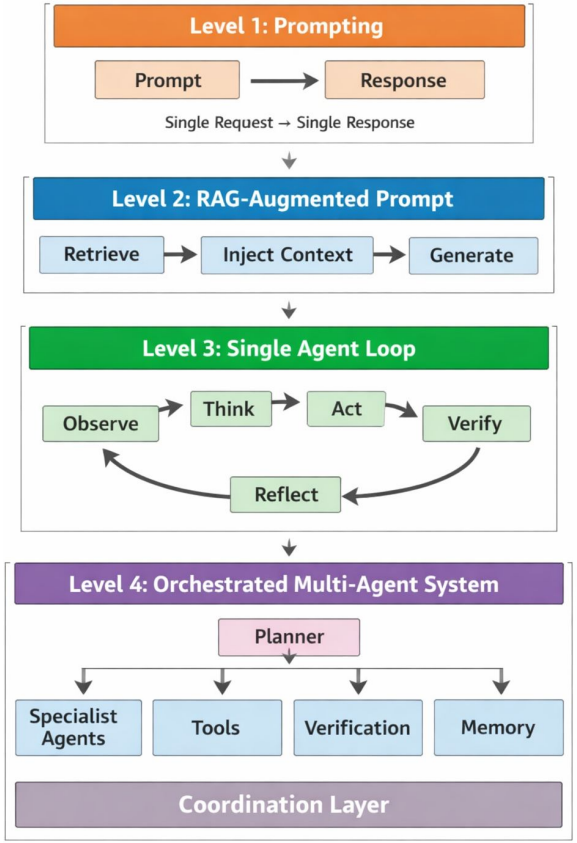
Level 3: Single Agent Loop

- Observe ⟶ Think ⟶ Act ⟶ Verify ⟶ Reflect

Level 4: Orchestrated Multi-Agent System

- Planner ⟶ Specialist Agents ⟶ Tools ⟶ Verification ⟶ Memory ⟶ Coordination Layer

https://www.ibm.com/think/tutorials/llm-agent-orchestration-with-langchain-and-granite#228874318



Example: AI Agent Fixing a Bug

User says: “Fix failing test in repository in /path/to/file”

Orchestrated AI Workflow

1. Planner Agent breaks task into subtasks
2. Code Agent edits source file
3. Tool executes test suite
4. Verification Agent checks results
5. If tests fail → Reflect and retry
6. If tests pass → Commit changes

AI Agent Orchestration Frameworks

- LangChain - Tool calling abstraction
- LangGraph - Graph-based agent workflows
- AutoGen - Multi-agent conversations
- CrewAI - Role-based agent teams

n8n

http://n8n.io/

Autogen: A programming framework for agentic AI

https://github.com/microsoft/autogen

Demo

https://github.com/imranraad07/CS-5001-AI-Augmented-SE/tree/master