MultiAgentDemoPresentation

April 9, 2025

1 Connecting the Bots:

1.1 Multi-agent AI applications

1.2 Jacob Albrecht

1.2.1 Applied Intelligence & Analytics, Bristol Myers Squibb

April 12, 2025

jacob.albrecht2@bms.com

1.3 Agenda for Today's Workshop

1.3.1 Overview of multi agent systems

- Capabilities of MAS for pharma industry
- Ingredients of MAS applications

1.3.2 Building a deep research agent (demo)

- Surf the web and run code *automatically*
- Get your computer ready to follow along

These slides

1.4 Caveat

AI Agents is a moving frontier: this snapshot of capabilites from Q1'25 will become obsolete

1.5 This workshop uses the Autogen Framework

Autogen is Microsoft's agentic framework designed to enable the development of intelligent agents. It provides tools and capabilities for creating autonomous systems that can interact, learn, and adapt to various tasks and environments. This framework is part of the broader effort to advance AI technologies and their applications in real-world scenarios.

To follow along you'll need:

- Linux, Mac, or Windows machine with Python 3.10 or above
- An API key - this model uses keys from OpenRouter.ai. OpenRouter provides *many* models through an "OpenAI compatible" interface

- `pip install openai autogen-agentchat autogen-ext[openai,magentic_one] autogenstudio`
- `playwright install` is needed for web browsing

1.5.1 OR

Quick start using GitHub

- Use/ create a personal GitHub.com account
- Create your own fork of the repository at https://github.com/chepyle/multiagent-demo
- Run the repository in a GitHub Codespace

1.5.2 Creating a github codespace (recommended)

1. Go to https://github.com/chepyle/multiagent-demo - contains all workshop materials
2. Click "fork project" to create your own copy
3. Click Code > Codespaces > Start Codespace from main branch
4. Wait for codespace to boot and package installation (~5min). A VSCode session will appear in browser.
5. Type `./run.sh` at the terminal to create an app, use password `sapa`
6. Click the link or go to https://localhost:8081 if running on your local machine

1.6 AI applications for the Pharmaceutical Industry

Healthcare is an huge opportuntiy for AI applications

- General Research e.g.
  - AI Scientist
  - OpenAI's Deep Research
  - Google's AI co-Scientist
- Discovery
  - Drug Design
- Development
  - Molecular Property Prediction
  - Process Simulation
- Manufacturing
  - Process Monitoring
  - Supply Chain
- Commercial / Enterprise
  - IT Pipelines
  - Forecasting
  - Data Science

1.7 A new front for AI: Multiagent Frameworks

AI that "does stuff": enabling control of software or other AI models

A rapidly emerging space, there are a number of popular libraries and frameworks, many using low-/no-code interfaces:

- AutoGPT
- Microsoft: Autogen, Copilot
- AG2: The "other" autogen
- Langflow
- Flowwise
- Crew AI
- Amazon: Bedrock flows

Note: Coding tools > No code tools

1.8 Core Concepts

- Multi-Agent Building Blocks
  - Tools
  - Models
  - Agents
  - Multi-Agent Teams
  - Orchestration
  - Termination

1.9 Tools

Single purpose functions

e.g. calculator, database connection, user input

1.10 Models

Large Language Models

e.g. GPT-4o, DeepSeek-R1, Claude, Llama, and others

Models have different capabilities: * Function calling : can run code commands (tools) * JSON Output : can return computer-readable structured results * Vision : Can accept multimodal (image + text) inputs

Access is through an application programming interface (API) with secret token

A token has been generated and shared for this workshop, be careful: it is like giving out your credit card!

1.11 Agents

Agents are the combination of Models and Tools, along with a descripton and set of instructions

e.g. Assistant Agent, Web Surfer Agent

1.12 Multi-Agent Teams

Tasks can be decomoposed and assigned to multiple agents with roles and personas

e.g. Writer/Editor , Resarcher/Summarizer/Verifyer teams

More examples from Autogen Studo creator at https://multiagentbook.com/labs/

1.13 Orchestration

Teams of agents can be overseen and roles assigned by an Orchestrator Agent

1.14 Task

User supplied objective to guide the team's activity

1.15 Termination

Setting the condition to stop the task

e.g. Approved result, Max # of attempts, Timeout, Max # of Tokens, User intervention

1.16 Magentic One - Example Architecture

1.17 Future of agents

Model Context Protocol (MCP) released by Anthropic is gaining popularity as a lightweight standard for interfacting LLMs and applications

A16Z article on MCP shows a recent snapshot of the landscape:

1.17.1 Safety Caveat

Autonomous agents carry risks and uncertainties! * Arbitrary code execution * High token utilization * Sending out LLM-generated results into the internet

Be sure to: * Run in a sandboxed environment e.g. local docker or codespace * Place limits on token utilization costs * Utilize a human-in-the-loop via `UserAgent` for sensitive tasks

1.17.2 Follow along in a github codespace (recommended)

Go to https://github.com/chepyle/multiagent-demo - contains all workshop materials

Click "fork project" to create your own copy

Click Code > Codespaces > Start Codespace from main branch

Wait for codespace to boot and package installation (~5min)

Type `./run.sh` at the terminal to launch workshop app, use password `sapa`

Click the link or go to https://localhost:8081

1.17.3 Autogen Studio Layout

1.18 Practical Considerations

What LLM to use?

Performance: Read articles, or check leaderboards e.g. lmarena, Artificial Analysis

Cost, Latency: compare at https://openrouter.ai/models

Capabilities: Support for Tool use, output structured information and/or logprobs

Implementing code

Once a multi-agent system yield promising results, it can be exported as code for automatic testing

1.19 Practical Considerations

Common Errors

`openai.NotFoundError: Error code: 404 - {'error': {'message': 'No endpoints found that support tool use...`

The selected model does not support tools, try using OpenAI, Gemini, Qwen-Turbo

`Exception: 'NoneType' object is not subscriptable`

Model call did not return a result. Can be due to latency on the provider side, but likely hitting limits of free-tier models.

"I made a change to the team, but I dont see a difference"

The teams and components are represented by JSON, making changes to components may not update uses elsewhere. Edit JSON directly to make sure changes are implemented.

Other Bugs

Autogen Studio is still in development, some features may be brittle

OpenRouter.ai Models may not be fully compatible with Autogen's OpenAI expectations

2 Our Tasks:

Web search:

"Search the web and summarize the current state of GLP-1 drug development, create a markdown-formatted report to document the findings, be sure to include references to your sources"

Coder/Analyst/Web:

Research "innovation from China related to oncology" , write code to create a database to store the key players and research subjects, with the goal of refreshing the database over time

3 Example Agent Files

A finished version of the workshop tasks are available in the `./app.zip` folder.

To use in autogen studio run: `./run.sh`, password `sapa`

Code examples are in `./src/` Be sure to use the environment variable `OPENROUTER_API_KEY`

4 After the Workshop

The code repository https://github.com/chepyle/multiagent-demo will remain public, but the API keys will be deactivated.

To use this code after the workshop, be sure to replace any api keys and base urls with the info from your LLM provider of choice

As Autogen Studio (currently v 0.4.2) changes, this code will become obsolete, but the learning never stops!