MultiAgentDemoPresentation

March 28, 2025

1 Connecting the Bots: multi-agent AI applications

1.1 Jacob Albrecht

1.1.1 Applied Intelligence & Analytics, Bristol Myers Squibb

April 12, 2025

2 Agenda for Todays Workshop

2.1 Overview of multi agent systems

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

2.2 Building a deep research agent (demo)

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

2.2.1 Caveat

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

3 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
- `pip install openai autogen-agentchat autogen-ext[openai,magentic_one] autogenstudio`

3.0.1 OR

Quick start using GitHub

- Use/ create a personal GitHub.com account
- Create a codespace from the repository at https://github.com/chepyle/multiagent-demo

3.1 Running locally

1. Download project from https://github.com/chepyle/multiagent-demo
2. create python virtual environment
3. run `pip install -r requirements.txt` to install packages

3.2 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

3.3 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

3.4 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

3.5 Core Concepts

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

3.6 Tools

Single purpose functions

e.g. calculator, database connection, user input

3.7 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!

3.8 Agents

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

e.g. Assistant Agent, Web Surfer Agent

3.9 Multi-Agent Teams

Tasks can be decomoposed and assigned to multiple agents with roles

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

3.10 Orchestration

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

3.11 Task

User supplied objective to guide the team's activity

3.12 Termination

Setting the condition to stop the task

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

3.13 Magentic One - Example Architecture

3.14 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:

3.14.1 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

3.15 Quick Start: Tutorial

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 load
5. Type `./run.sh` at the terminal to create an app
6. Click the link or go to https://localhost:8081 if running on your local machine

3.15.1 Autogen Studio Layout

4 Our Tasks:

Web search: Use RoundRobinTeam to "Search the web and summarize the current state of GLP-1 drug development"

Coder/Analyst/Web: Create a database-backed tool to periodically monitor "innovation from China related to oncology"

5 Example Agent Files

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

To use in autogen studio run: `./run.sh final_app`

6 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- the learning never stops!