AI Agents & Model Context Protocol

Trí tuệ Nhân tạo

và Giao thức Ngữ cảnh Mô hình

AI Agents - Định nghĩa

AI Agent là gì?

AI Agent (Trí tuệ nhân tạo tự chủ) là một hệ thống máy tính có khả năng tự động quan sát môi trường, đưa ra quyết định và hành động để đạt được mục tiêu cụ thể.

Đặc điểm cốt lõi

- **Tự chủ (Autonomy):** Hoạt động mà không cần sự can thiệp liên tục của con người
- **Nhận thức môi trường (Perception):** Thu thập thông tin từ môi trường thông qua cảm biến, dữ liệu hoặc API
- **Hành động (Action):** Tác động lại môi trường bằng cách đưa ra phản hồi, điều khiển thiết bị
- **Mục tiêu (Goal-directed):** Hành động hướng đến một kết quả cụ thể

AI Agent vs Large Language Model

Tiêu chí	AI Agent	LLM
Định nghĩa	Hệ thống tự động nhận thức môi trường, ra quyết định và hành động	Mô hình AI chuyên xử lý và tạo văn bản dựa trên dữ liệu ngôn ngữ
Tự chủ	Có thể hoạt động độc lập không cần con người	Chỉ phản hồi khi có đầu vào (prompt)
Hành động	Tác động vật lý/số (gửi email, điều khiển robot)	Chỉ tạo văn bản, không thực thi hành động bên ngoài
Công nghệ nền	Kết hợp nhiều AI: ML, NLP, Robotics, API	Tập trung vào xử lý ngôn ngữ (NLP)
Ví dụ	Alexa, Tesla Autopilot, chatbot đặt lịch tự động	ChatGPT, Gemini, Claude

Mối quan hệ

LLM có thể là một phần của AI Agent: Agent đặt lịch họp dùng LLM để hiểu yêu cầu, sau đó kết nối với Google Calendar để thực thi.

Thành phần và Cấu trúc AI Agent

1. Cảm biến (Sensors)

Vật lý: Camera, microphone, cảm biến nhiệt độ

Kỹ thuật số: API thời tiết, database, tín hiệu máy chủ

2. Bộ xử lý (Processor)

- Machine Learning
- Natural Language Processing
- Computer Vision
- Rule-Based Systems

3. Bộ tác động (Actuators)

Vật lý: Động cơ robot, màn hình, loa

Kỹ thuật số: Gửi email, cập nhật database, gọi API

4. Học tập (Learning)

- Reinforcement Learning
- Online Learning
- Transfer Learning

Ví dụ Google Nest: Microphone nghe lệnh → NLP hiểu lệnh → ML kiểm tra thói quen → Gửi tín hiệu điều hòa → Ghi nhớ preferences

Phân loại AI Agent (Phần 1)

1. Simple Reflex Agent

- Hoạt động dựa trên quy tắc "if-then" cứng nhắc
- Chỉ phản ứng với tín hiệu hiện tại
- **Ví dụ:** Thermostat - "Nếu nhiệt độ > 30°C → Bật điều hòa"

2. Model-Based Reflex Agent

- Duy trì mô hình nội bộ về thế giới
- Ra quyết định dựa trên lịch sử và dự đoán
- **Ví dụ:** Xe tự lái theo dõi vị trí các xe khác, biển báo

3. Goal-Based Agent

- Hành động để đạt mục tiêu cụ thể
- Sử dụng kế hoạch hành động và đánh đổi
- **Ví dụ:** Google Maps tìm đường ngắn nhất, tránh kẹt xe

Phân loại AI Agent (Phần 2)

4. Utility-Based Agent

- Tối ưu hóa "độ hữu ích" (utility function)
- Đánh giá kết quả qua hàm tiện ích
- **Ví dụ:** Hệ thống gợi ý sản phẩm ưu tiên lợi nhuận cao

5. Learning Agent

- Tự cải thiện hiệu suất qua kinh nghiệm
- 4 thành phần: Performance Element, Critic, Learning Element, Program Generator
- **Ví dụ:** Chatbot học từ feedback để cải thiện câu trả lời

Lựa chọn loại Agent phù hợp

Simple Reflex: IoT đơn giản | **Model-Based:** Xe tự lái | **Goal-Based:** Logistics | **Utility-Based:** Recommendation | **Learning:** Chatbot thông minh

Multi-Agent Systems (MAS)

Định nghĩa: Hệ thống bao gồm nhiều AI Agent tương tác với nhau để giải quyết các vấn đề phức tạp

Đặc điểm

- **Phân phối tính toán:** Chia nhỏ tác vụ lớn thành các tác vụ con
- **Giao tiếp giữa các Agent:** Chia sẻ thông tin, phối hợp hành động
- **Đàm phán và hợp tác:** Thương lượng để đạt mục tiêu chung
- **Khả năng chịu lỗi:** Nếu một agent gặp sự cố, các agent khác vẫn hoạt động

Ví dụ thực tế

Hệ thống giao dịch tài chính: Nhiều agent theo dõi thị trường, phân tích rủi ro, thực hiện giao dịch

Quản lý chuỗi cung ứng: Agent theo dõi kho hàng, dự báo nhu cầu, tối ưu hóa vận chuyển

Kiến trúc Multi-Agent Systems

Cấu trúc tổ chức

Hierarchical Structure

- Manager Agents
- Worker Agents
- Broker Agents

Network Structure

- Peer-to-Peer
- Hub-and-Spoke
- Mesh Network

Cơ chế phối hợp

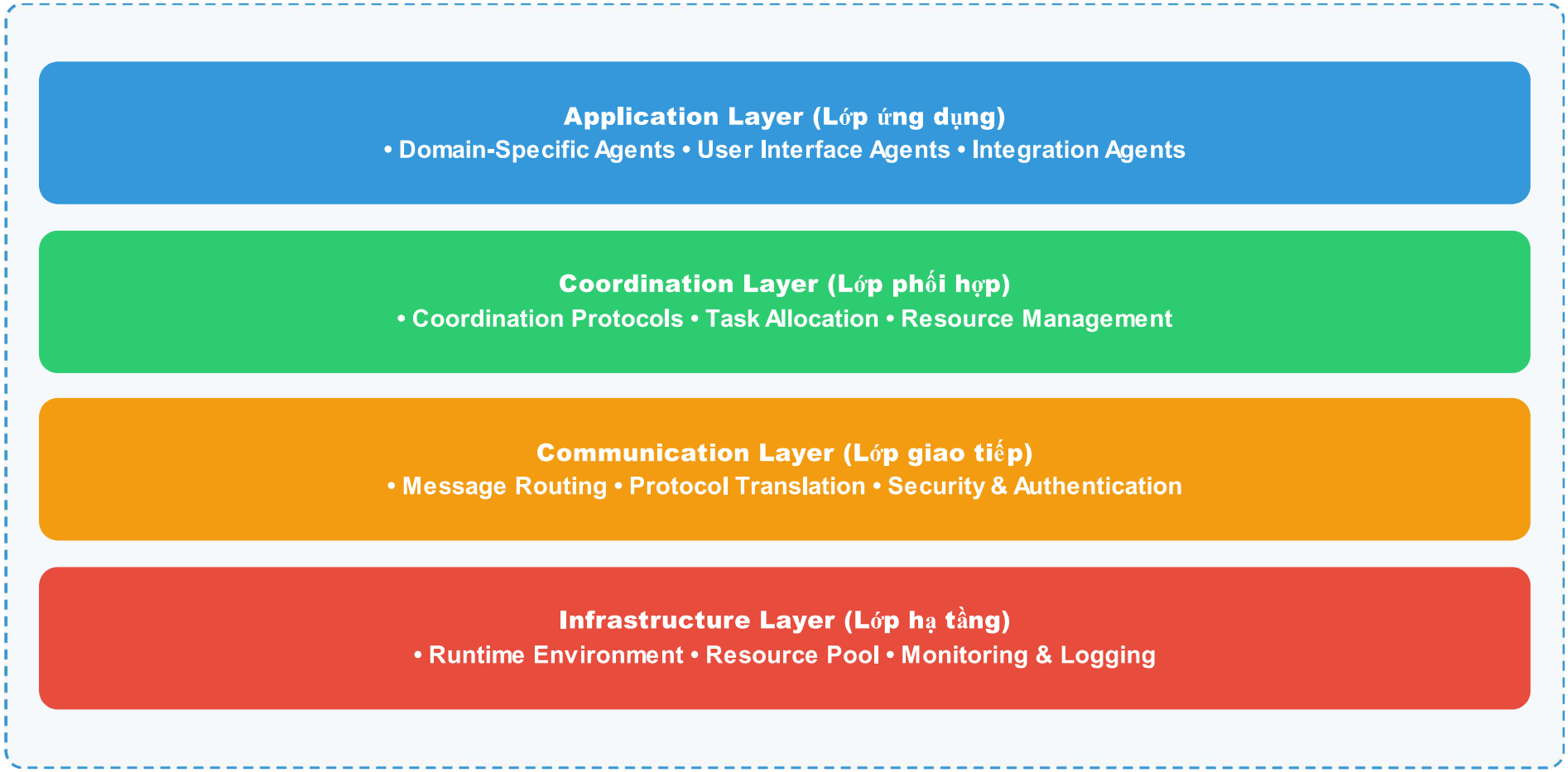
- Contract Net Protocol
- Auction Mechanisms
- Voting Systems
- Consensus Algorithms

Thành phần cốt lõi của MAS

Individual Agents: Knowledge Base, Reasoning Engine, Communication Module, Action Executor, Learning Component

Communication Infrastructure: Message Queue, Routing Mechanism, Protocol Standards (FIPA-ACL, KQML)

Kiến trúc phân lớp Multi-Agent Systems



Model Context Protocol (MCP)

MCP là gì?

Model Context Protocol (MCP) là một giao thức được thiết kế để quản lý ngữ cảnh mô hình, đóng vai trò cầu nối giao tiếp 2 chiều giữa mô hình AI và các công cụ, dịch vụ bên thứ 3.

Tại sao MCP quan trọng?

Không có MCP, các mô hình AI chỉ có thể sử dụng những gì học được qua huấn luyện:

- ❌ Không thể truy cập thông tin hiện tại từ internet
- ❌ Không thể lấy dữ liệu từ cơ sở dữ liệu
- ❌ Không thể sử dụng dịch vụ chuyên biệt
- ❌ Không thể lưu thông tin vào tệp
- ❌ Không thể kết nối với công cụ bên ngoài

✅ **MCP giải quyết tất cả những hạn chế này!**

Kiến trúc MCP

Mô hình Client-Host-Server

MCP Server cung cấp 3 phương thức

📝 Prompt

Các mẫu lệnh được xác định trước cho LLM

🔧 Tool

Các hàm cho phép mô hình thực hiện hành động

📊 Source

Dữ liệu có cấu trúc như tệp, database, lịch sử

Chi tiết Kỹ thuật MCP

JSON-RPC 2.0 Messages

1. Request

```
{ "jsonrpc": "2.0", "id": "string | number", "method": "method_name", "params": { /* Tham số */ } }
```

2. Response

```
{ "jsonrpc": "2.0", "id": "string | number", "result": { /* Dữ liệu kết quả */ } }
```

3. Notification

```
{ "jsonrpc": "2.0", "method": "notification_method", "params": { /* Tham số thông báo */ } }
```

MCP SDK

- Python SDK
- TypeScript SDK
- Java SDK
- C# SDK

Tổng kết

AI Agents

- Hệ thống tự chủ có khả năng nhận thức, quyết định và hành động
- 5 loại chính từ đơn giản đến phức tạp
- Multi-Agent Systems cho các bài toán phức tạp
- 4 thành phần cốt lõi: Sensors, Processor, Actuators, Learning

Model Context Protocol

- Cầu nối giữa AI models và thế giới bên ngoài
- Kiến trúc Client-Host-Server linh hoạt
- Sử dụng JSON-RPC 2.0 để giao tiếp
- Cung cấp Prompt, Source, Tool cho mô hình AI

Tương lai

Sự kết hợp giữa AI Agents và MCP sẽ tạo ra những ứng dụng AI mạnh mẽ, có thể tương tác với thế giới thực một cách tự chủ và thông minh. Điều này mở ra những khả năng vô hạn cho việc tự động hóa các quy trình phức tạp và nâng cao trải nghiệm người dùng.