

AI 特許紹介(82)
AI 特許を学ぶ！究める！
～AgentLite 特許～

2025 年 11 月 10 日
河野特許事務所
所長弁理士 河野英仁

「AI 特許紹介」シリーズは、注目すべき AI 特許のポイントを紹介します。熾烈な競争となっている第 4 次産業革命下では AI 技術がキーとなり、この AI 技術・ソリューションを特許として適切に権利化しておくことが重要であることは言うまでもありません。

AI 技術は Google, Microsoft, Amazon を始めとした IT プラットフォーマ、研究機関及び大学から毎週のように新たな手法が提案されており、また AI 技術を活用した新たなソリューションも次々とリリースされています。

本稿では米国先進 IT 企業を中心に、これらの企業から出願された AI 特許に記載された AI テクノロジー・ソリューションのポイントをわかりやすく解説致します。

1.概要

特許出願人 Salesforce

出願日 2024 年 6 月 10 日

公開日 2025 年 8 月 21 日

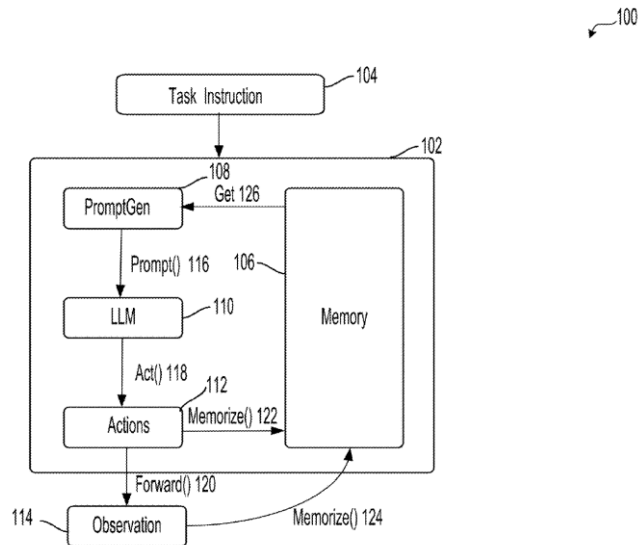
公開番号 US20250265443

発明の名称 タスク指向の階層型エージェントアーキテクチャを構築するためのシステムと方法

443 特許は、LLM エージェントの推論、アーキテクチャ、およびマルチエージェントの開発を容易にするよう設計されたマルチエージェントプラットフォームである AgentLite 技術に関する。

2.特許内容の説明

下記図は、個別エージェント 102 のアーキテクチャ 100 を示す説明図である。



個別エージェント 102 は、プロンプト生成モジュール (PromptGen) 108、LLM モジュール 110、アクションモジュール 112、およびメモリモジュール 106 を含むコントローラを含む。コントローラは、ユーザからタスク指示 104 を受信し、タスク指示 104 によって示されたタスクを完了するために、個別エージェント 102 を取り囲む環境に対してアクションモジュール 112 によってアクションを実行する。

例えば、タスク指示 104 は「エレキギターを購入する」である。タスク指示 104 に基づいて、コントローラは「[URL] Amazon.com」、「[検索] エレキギター」、「[クリック] アイテム(1)」、「[クリック] チェックアウト」などのアクション 118 を実行する。プロンプト生成モジュール 108 は、タスク指示に基づいてプロンプト 116 を生成し、LLM モジュール 110 は出力としてアクション 118 を生成する。プロンプト生成モジュール 108 は、エージェントの役割記述、タスク指示 104、プロンプト生成の制約、エージェントのアクション、例など、1つ以上の要素に基づいてプロンプト 116 を構築する。

アクションモジュール 112 はアクション 118 を解析し、アクション 118 を実行する。コントローラは、実行されたアクションをメモリモジュール 106 に記憶する (122)。また、コントローラはアクションを環境に転送し (120)、観測 114 を取得する。観測値はメモリモジュール 106 に保存される。

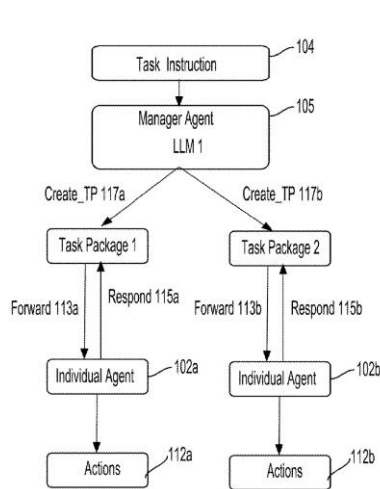
アクションモジュール 112 がアクション 118 を実行した後、環境は、コントローラに伝達される観測 114 を介して、環境の状態に関する更新情報を提供する。観測 114 は、メモリモジュール 106 に格納される。メモリモジュール 106 は、観測 114 と個々のエージェント 102 の履歴アクションを記憶する。プロンプト生成モジュール 106 は、履歴アクションと対応する観測を取得し (126)、これらを連結してプロンプト生成モジ

ジュール 108 に入力し、プロンプト 116 を生成する。生成されたアクション 118 と返された観測 114 は、次にメモリモジュール 106 に記憶される (124)。

例えば、タスク指示 104 は「エレキギターを購入する」である。LLM モジュール 110 への最初のプロンプト 116 は、タスク指示 104 と所定のプロンプトテンプレートのみに基づいて生成される。プロンプトテンプレートは、LLM モジュール 110 が環境に対して実行するアクションを決定する目的、及び、環境に対して実行され得るアクションの想定される種類を記述する。タスク指示 104 は、プロンプト 116 を提供するために、定義済みテンプレートプロンプトに追加される。LLM モジュール 110 によって生成されるアクション 118 は、「[URL] Amazon.com」とすることができる。結果として得られる観測 114 は、利用可能なリンク及び検索バーを含む利用可能なテキストフィールドを含む、Amazon.com ウェブサイトの説明となる。この観測 114 はメモリ 106 に格納される。

プロンプト 116 は、プロンプトテンプレートおよびタスク指示 104 に加えて、最初に実行されたアクションとその結果の観測 114 を含むように更新される。この更新されたプロンプト 116 に基づいて、LLM モジュール 110 は、例えば「[検索]エレキギター」などの次のアクションを生成する。このプロセスは、タスクが完了するまで、アクションモジュール 112 によって実行される追加のアクションと観測 114 を用いて反復され続ける。コントローラは、タスクが完了したことを判定する。例えば、各観測の後、ニューラルネットワークベースのモデル（例えば、LLM モジュール 110）に、観測 114 がタスク指示 104 に記述されたタスクの完了を表すかどうかを尋ねるプロンプトが与えられる。

下記図は、マネージャエージェント 105 と複数の個別エージェント 102a および 102b とを含む階層型エージェントアーキテクチャ 101 を示す説明図である。



マネージャエージェント 105 は、タスクをサブタスクに分解し、サブタスクに基づいて個別エージェントを選択し、対応する個別エージェントにサブタスクを送信して完了させる。各個別エージェントは、マネージャエージェント 105 のチームに追加される際に、マネージャエージェント 105 のアクションの一種として参照される。

マネージャエージェント 105 は、ユーザからタスク指示 104 を受信する。マネージャエージェント 105 は、タスク指示 104 を分解し、タスク指示 104 から第 1 のサブタスクを生成する。マネージャエージェント 105 のコントローラは、様々な機能を備えた複数の利用可能な個別エージェントから個別エージェント 102a を選択し、それぞれの API を介して個別エージェント 102a との通信接続を構築する。その後、マネージャエージェント 105 のコントローラは、第 1 のサブタスクに対応するタスクパッケージ 1 (create_TP 117a) を作成し、タスクパッケージ 1 を個別エージェント 102a に転送する (113)。タスクパッケージ 1 は、個別エージェント 102a への入力となり、個別エージェント 102a にアクションを生成させ、そのアクションは対応するアクションモジュール 112a によって実行される。

タスクパッケージ 1 の実行後、個別エージェント 102a は、応答メッセージ 115a を送信することにより、マネージャエージェント 105 に応答する。応答メッセージ 115a には、第 1 サブタスクの完了ステータス、個別エージェント 102a の識別情報（または個別エージェント 102a の LLM）、及び、マネージャエージェント 105 の識別情報等が含まれる。

個別エージェント 102a から応答メッセージ 115a を受信すると、マネージャエージェント 105（またはマネージャエージェント 105 の LLM モジュール）は、応答メッセージ 115a とタスク指示 104 に基づいて第 2 のサブタスクを生成する。同様に、マネー

ジャ 105 のコントローラは、様々な機能を備えた複数の利用可能な個別エージェントから別の個別エージェント 102b を選択し、それぞれの API を介して個別エージェント 102b との別の通信接続を構築する。その後、マネージャエージェント 105 は、第 2 のサブタスクに対応するタスクパッケージ 2 (create_TP 117b) を作成し、タスクパッケージ 2 を個別エージェント 102b に転送する (113b)。タスクパッケージ 2 は、個別エージェント 102b への入力となり、個別エージェント 102b にアクションを生成させ、対応するアクションモジュール 112b によって実行させる。

タスクパッケージ 2 の実行後、個別エージェント 102b は応答メッセージ 115b を送信することによりマネージャエージェント 105 に応答する。応答メッセージ 115b には、第 2 のサブタスクの完了ステータス、個別エージェント 102b の識別情報（または個別エージェント 102b の LLM）、及び、マネージャエージェント 105 の識別情報等が含まれる。

下記図は、個別エージェント及びマネージャエージェントを定義するコード例を示す説明図である。

```
from agentlite.agents import BaseAgent

name = "agent_name"
role = "describe the roles of this agent"
actions = [Action1, Action2]
agent = BaseAgent(name=name, role=role, actions=actions)
```

(I)

```
from agentlite.agents import ManagerAgent

# assuming we have already designed three agents
# as a team for manager agents to control
team = [agent_1, agent_2]
name = "manager_agent"
role = "controlling multiple agents to complete task"
manager = ManagerAgent(name=name, role=role, TeamAgents=team)
```

(II)

(I)は、階層型エージェントアーキテクチャ構築モジュール 230 によって、個別エージェントを「agent_name」という名前、役割「describe the roles of this agent(このエージェントの役割を記述する)」、および、アクション 1 とアクション 2 に特化するように初期化する擬似コードの例を示している。

(II)は、階層型エージェントアーキテクチャ構築モジュール 230 によって、マネージャ

ャエージェントを初期化し、個別エージェント 1 と個別エージェント 2 を含むチームを構築し、「manager_agent」という名前を持ち、「controlling multiple agents to complete task(複数のエージェントを制御してタスクを完了する)」という役割を持つようにするための擬似コードの例を示している。

このモジュール方式は、多様なアクションを必要とする複雑なタスクに対処するために、マネージャエージェントの監督下で複数の専門エージェント（例えば、個別エージェント）を組み込むことができるマルチエージェントシステムの組み立てを容易にする。コンポーネントの総合的な能力を活用することで、システムの効率性が向上し、スケーラブルなマルチエージェントシステムの潜在能力が高まる。

3.クレーム

443 特許のクレーム 1 は以下の通りである。

1. タスクを実行するための複数のニューラルネットワークモデルの階層構造を構築する方法において、

データインターフェースを介してタスク指示を受信し、

第 1 ニューラルネットワークモデルによって、タスク指示から第 1 サブタスクを生成し、

第 1 サブタスクに基づいて、複数のニューラルネットワークモデルから第 2 ニューラルネットワークモデルを選択し、

第 1 アプリケーションプログラミングインターフェース(API)を介して、第 1 ニューラルネットワークモデルと第 2 ニューラルネットワークモデルとの間に第 1 接続を構築し、

第 1 ニューラルネットワークモデルによって、第 2 ニューラルネットワークモデルに準拠した形式で第 1 サブタスクパッケージを生成し、

第 1 接続を介して、第 1 サブタスクパッケージを実行する第 2 ニューラルネットワークモデルから第 1 出力を受信し、

第 1 ニューラルネットワークモデルによって、タスク指示および第 1 出力に基づいて第 2 サブタスクを生成し、

少なくとも部分的に第 2 のサブタスクに基づいて、複数のニューラルネットワークモデルから選択された 1 つ以上のニューラルネットワークモデルによってタスク命令が共同で実行されるようにする。

4. 本特許に関連する論文

本特許に関する論文“AgentLite: A Lightweight Library for Building and Advancing

Task-Oriented LLM Agent System”¹が、Salesforce の Zhiwei Liu 氏らにより公表されている。

本論文では、新しいAI エージェントライブラリ **AgentLite** を提案している。**AgentLite** は、オープンソースとして下記 Github にて公開されており、LLM エージェントの推論、アーキテクチャ、およびアプリケーションを容易にするための軽量でユーザーフレンドリーなプラットフォームを提供している。

<https://github.com/SalesforceAIResearch/AgentLite>

AgentLite は、エージェントのタスク分解能力を強化し、マルチエージェントシステムの開発を容易にするように設計されたタスク指向フレームワークである。下記図は、**AgentLite** ライブラリとエージェント構築用の既存ライブラリでサポートされている機能の比較を示す説明図である。

Library	AgentLite	AutoGen	LangChain	Camel	CrewAI
Task Decomposition	✓	-	-	✓	✓
Multi-Agent Orchestration	✓	✓	-	✓	✓
Extendable Reasoning Types	✓	-	✓	-	-
Memory Module	✓	-	✓	✓	-
Prompter Module	✓	-	✓	✓	-
# (Core) Code Lines	959	8,966	248,650	4,987	1,504

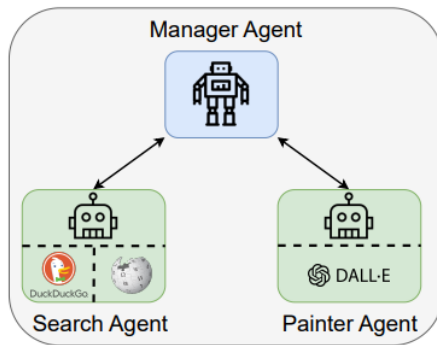
✓ はライブラリで本質的にサポートされていることを示し、- はサポートされていないことを示す。

論文には、様々な形態の階層型エージェントアーキテクチャ、及び、各エージェントのコードが示されている。

(1) オンラインペインター

下記図は、マネージャエージェントと 2 つの個別エージェントを用いたオンラインペインターアプリケーションを示す。

¹ Zhiwei Liu, et al. “AgentLite: A Lightweight Library for Building and Advancing Task-Oriented LLM Agent System” arXiv:2311.09235v2 [cs.LG] 3 Jun 2024



A1: Online Painter

任意のオブジェクトが与えられた場合、オンラインペインターアプリケーションは、まず関連する視覚的特徴をオンラインで検索し、次にその詳細な特徴に基づいてオブジェクトの画像をペイントする。オンラインペインターアプリケーションは、3つのエージェント（マネージャエージェント、検索エージェント、及び、ペインターエージェント）によって構成される。検索エージェントとペインターエージェントはそれぞれ個別のエージェントである。検索アクションとペイントアクションは、それぞれ検索エージェントとペインターエージェントに追加される。検索エージェントには、情報収集に使用する2つのアクションを含めることができる。

例えば、DuckSearch アクションはクエリに基づいて DuckDuckGo API を介してオンライン情報を収集し、WikipediaSearch アクションは Wikipedia で関連情報を検索する。これらの2つのアクションにより、検索エージェントはペイントされたオブジェクトに関する情報を取得する。ペインターエージェントは、DuckSearch アクションと WikipediaSearch アクションからの説明に基づいて絵を描く DALL-E アクションを実行する。次に、マネージャエージェントを使用して、オンラインペイントタスクを制御する。マネージャエージェントは、タスクパッケージを介してタスクを直接受信し、検索エージェントと Painter エージェントで構成されるチームメンバーのためにタスクをさまざまなステップに分解する。下記図は、ペインターエージェント及びマネージャエージェントのコード例を示す説明図である。

```

name = "Painter_agent"
role = "You are a painter and can draw a picture with Paint action."
actions = [Paint()]
PaintAgent = BaseAgent(name=name, role=role, actions=actions)

```

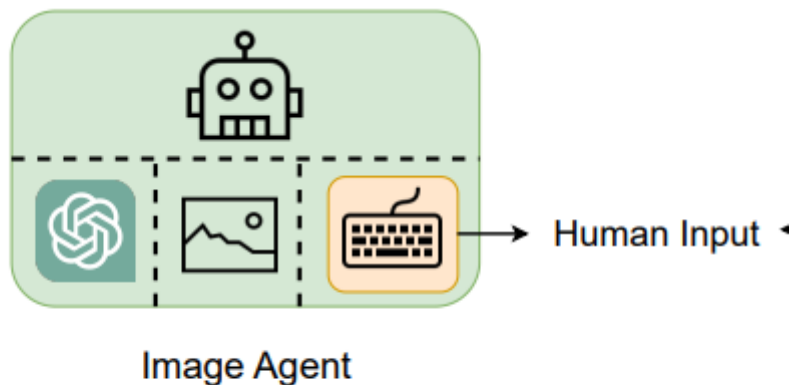
```

manager = ManagerAgent(TeamAgents=[PaintAgent, SearchAgent])

```

(2)画像エージェント

下記図は対話型画像理解アプリケーションを示す説明図である。



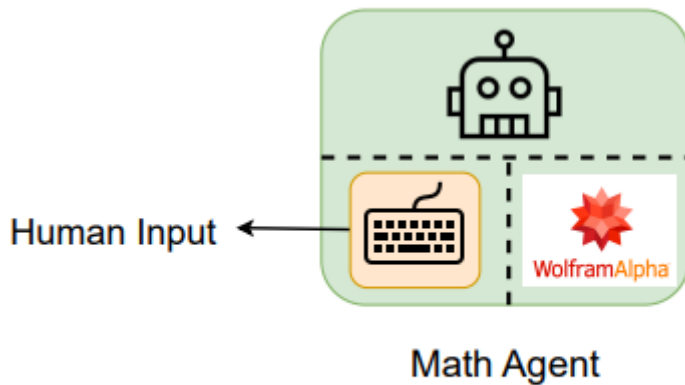
A2: Interactive Image Understanding

対話型画像理解アプリケーションは、人間が介入する指示を備えたマルチモダリティアプリケーションである。対話型画像理解アプリケーションには、画像エージェントが含まれる。画像が与えられると、アプリケーションは画像に基づいて人間からの質問に回答する。人間は、「画像には何がありますか?」「橋の色は何ですか?」など、自然言語を使用して、アプリケーションに複数回質問をする。ImageDisplay アクションは、人間のユーザに画像を表示し、ImageQuery アクションは画像に基づいて質問に答える。ImageQuery アクションは、GPT-4-visionpreview API によってバックエンド化されている。HumanInput アクションによって強化された画像エージェントは、人間からの指示を取得してそれに従う。下記図は、画像エージェントのコード例を示す説明図である。

```
from agentlite.agents import BaseAgent
name = "Image_agent"
role = "You are image agent to view and answer questions from image."
actions = [HumanInput(), ImageQuery(), ImageDisplay()]
ImageAgent = BaseAgent(name=name, role=role, actions=actions)
```

(3)数学エージェント

下記図は数学問題解決アプリケーションを示す説明図である。



ding

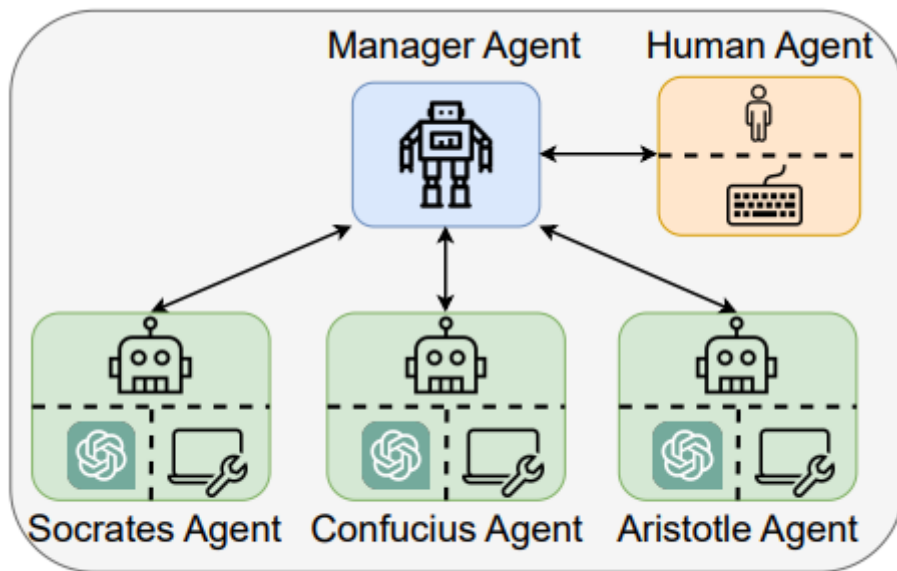
A3: Math Solving

数学問題解決アプリケーションは、数学エージェント（例えば、個人エージェント）を用いて構築されたアプリケーションである。数学エージェントは、人間の入力から数学の問題を受け取り、問題を解く。人間の入力は、「 $75 \times 34 + 12 =$ 」のような直接的な数式、または「75 に 34 を掛けて 12 を足すとどうなるか」のような問題の説明である。WolframAlpha は数学コパイロットエージェントの一つである。数学エージェントは、HumanInput アクションを使用して人間から数学の問題を取得し、WolframAlphaSolver アクションを使用して問題を解く。本アプリは、基本演算から方程式の解決や微積分まで、幅広い数学の問題を解くことができる WolframAlpha API と統合されている。下記図は、数学エージェントのコード例を示す説明図である。

```
from agentlite.agents import BaseAgent
name = "MathAgent"
role = "You can answer math questions by WolframAlphaSolver action."
actions = [HumanInput(), WolframAlphaSolver()]
MathAgent = BaseAgent(name=name, role=role, actions=actions)
```

(4) 哲学者エージェント

下記図はソクラテス、孔子、アリストテレス等の哲学者の思考を模倣するように構成された哲学者チャットアプリケーションを示す説明図である。



A5: Philosophers Chatting

哲学者チャットアプリケーションは、マルチエージェントフレームワーク、例えば、人間エージェントに加えて、ソクラテスエージェント、孔子エージェント、及び、アリストテレスエージェントをチームメンバーとするマネージャエージェントにより構成される。人間は毎回、マネージャエージェントに「人生で何を追求すべきか」といった哲学に関する質問をする。マネージャエージェントは質問を受け取ると、ソクラテス、孔子、及び、アリストテレスエージェントに個別に意見を尋ね、結果を要約して質問に答える。マネージャエージェントは、4つの異なるエージェントをチームに統合することでインスタンス化される。マネージャエージェントは、関連する質問の特定、これらの問い合わせを適切なエージェントに転送、そして議論を要約する議論の流れを調整する。下記図は、ソクラテスエージェントのコード例を示す説明図である。

```
from agentlite.agents import BaseAgent
name = "Socrates"
role = "You are Socrates. You are very familiar with Socrates's Book and Thought. Tell your opinion on behalf of Socrates"
Socrates_Agent = BaseAgent(name=name, role=role)
```

以上

著者紹介

河野英仁

河野特許事務所、所長弁理士。立命館大学情報システム学博士前期課程修了、米国フランクリンピアースローセンター知的財産権法修士修了、中国清华大学法学院知的財産夏

季セミナー修了、MIT(マサチューセッツ工科大学)コンピュータ科学・AI 研究所 AI コース、生成 AI ビジネスコース修了。

[AI 特許コンサルティング](#)、[医療 AI 特許コンサルティング](#)の他、米国・中国特許の権利化・侵害訴訟を専門としている。著書に「世界のソフトウェア特許(共著)」、「FinTech 特許入門」、「[AI/IoT 特許入門 3](#)」、「[ブロックチェーン 3.0](#)(共著)」がある。