ChatGPT

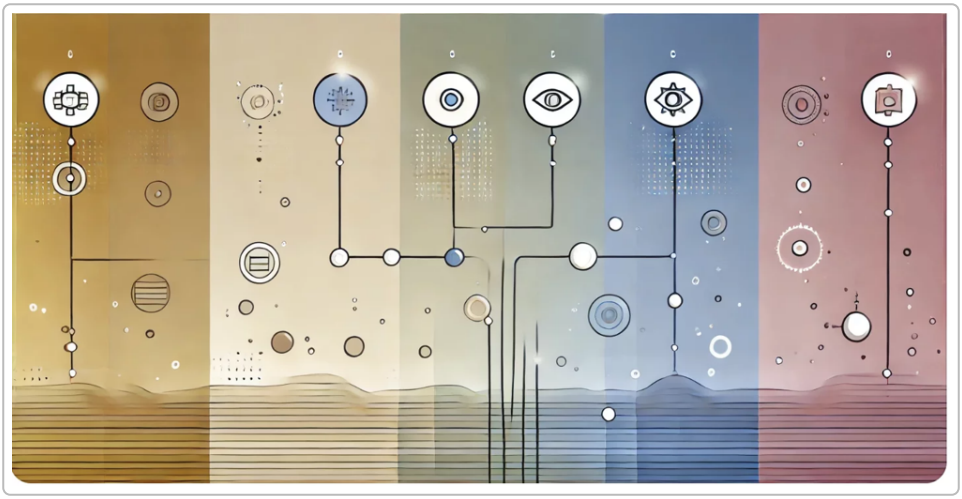

図: OpenAIが提唱するAI能力5段階の概念図（左からレベル1〜5）。チャットボットから始まり、最終的に組織全体を代行できるAIへと段階的に発展するイメージ。

OpenAIの5レベルにおけるLLM層とアプリケーション層の役割

OpenAIはAIの発展を5つのレベルで定義しています [1]。各レベルでは、大規模言語モデル(LLM)そのものが担う**「LLM層」**の役割と、そのモデルを取り巻く**「アプリケーション層」**（ツールやUI、メモリ管理などを含むシステム部分）の役割がどのように進化・分担されるかが大きく異なります。以下、レベル1（チャットボット）からレベル5（組織）までそれぞれについて、技術的・構造的観点からLLM層とアプリケーション層の役割と相互作用の詳細を解説します。

レベル1：チャットボット（Conversational AI）

LLM層（モデルの能力・設計）： レベル1では、人間と自然な対話ができる**チャットボット**が該当します [2] [3]。この段階のLLMは主に大量のテキストデータで事前学習された汎用言語モデルで、ユーザの発話に対し的確かつ一貫性のある応答を生成することに特化しています。モデルは文脈を理解し、人間らしい文章を作る能力に優れており、OpenAIのChatGPTやAnthropicのClaudeなどがその代表例です [2] [3]。モデルアーキテクチャとしてはTransformerに基づく数十億〜数千億パラメータ級の言語モデルで、主にテキストのみを扱います（**マルチモーダル**対応はまだ基本的になし）。この層では**スケーリング**（モデルサイズの拡大や学習データ増加）によって、応答の流暢さや知識量が向上しました。例えばGPT-3.5からGPT-4へのスケーリングにより、より複雑な指示への対応能力や創造性が大幅に向上しています [4]。モデルは**RLHF（人間フィードバックによる強化学習）**などの調整によってユーザの指示に従いやすく調教され、対話ボットとして適切に振る舞うよう最適化されています。

アプリケーション層（構造と機能）： レベル1のアプリケーション層は比較的単純で、ユーザとLLMとの**インターフェース**役を担います。典型的にはチャット画面や対話APIがこの層に相当し、ユーザの入力メッセージを受け取り、それをそのままままたは何らかのシステムプロンプトと結合してLLMに渡し、生成された応答をユーザに表示します。ここでは**UI/UX**が重要で、人が違和感なくチャットできるテキストインターフェースや音声対話システムが実装されます。アプリ層はLLMの**状態管理**も行いますが、その範囲はユーザごとの対話セッション管理や直近の対話履歴（コンテキスト）の保持程度です。例えばユーザとの過去数ターンの会話を

保持し、毎回LLMへのプロンプトに前の対話履歴を含めることで、一貫した対話を実現します。**メモリ**については、LLM自体には長期記憶は無く各応答ごとにコンテキスト内の情報しか参照できないため、アプリ層が必要に応じて過去の会話を要約したり、長い対話履歴の一部を削除したりしてコンテキスト長を管理します。基本的にレベル1では外部の**ツール呼び出し**やデータベース検索などは行わず、LLM単体の出力に依存します。アプリケーション層の制御ロジックはシンプルで、ユーザ入力→（システムプロンプト付与）→LLM応答→出力表示という直線的な流れです。

両層のインターフェース設計: レベル1ではLLMとアプリは主に**テキストプロンプト**を介してやり取りします。具体的には、アプリ層が「システムメッセージ」「ユーザメッセージ」などの形式でプロンプトを組み立てLLMに送り、返ってきた「アシスタントメッセージ」を解析してユーザに提示する形です。この際、アプリ層では**システムプロンプト**を用いてモデルの振る舞いをある程度制御します（例：「あなたは有能なアシスタントです…」等の指示を与える）。LLM層は与えられたプロンプト内の指示に従い応答文章を生成します。通信は同期的な問答であり、モデルから予期せぬ動作を引き出すような特殊なフォーマットは使いません。**制御方式**としては、出力されたテキストに対しアプリ側で内容のフィルタリング（不適切な発言の除去など）を行うケースがありますが、基本的にはLLMの出力をそのまま返します。つまり、インターフェースはAPI越しのテキスト入出力で単純明快であり、LLM層が生成したものを表示する以上の介入は最小限です。

進化と依存関係の推移: チャットボット段階では、LLM層の性能向上が全体システムの能力をほぼ決定づけます [3]。実際、2018〜2022年頃にかけてモデルのサイズ拡大と訓練データ増加による言語モデルの飛躍的進化が、高度なチャットボット実現の原動力となりました。アプリケーション層は当初から存在したルールベースのチャットボット（例えば定型応答システム）技術を土台に、LLMを差し替える形で発展しており、モデル改善がリードしアプリ側がそれに対応してきた歴史があります。例えば、GPT-3の登場により人間らしい対話が可能となったため、OpenAIはそれをチャットGPTという製品（アプリ層を含む形態）に昇華しました。その際、RLHFによる追加学習やプロンプト設計といった手法でアプリ層・モデル層双方が調整され、ユーザにとって違和感の少ない対話体験となるよう**共進化**しています。また、チャットボット運用を通じて集まったフィードバックはモデル再学習に活用され、モデル能力をさらに向上させるというサイクルも見られます。このように、レベル1ではLLM層の基盤技術が先導しつつ、アプリ層がそれを受けてユーザフレンドリーな形に整えるという関係です。

レベル2：推論者（Reasoners）

LLM層（モデルの能力・設計）： レベル2の**「推論者」**に達したLLM層では、人間の専門家（博士号レベル）に匹敵する高度な問題解決・推論能力が求められます [5] [6]。具体的には、与えられた課題に対し外部ツールに頼らず**論理的推論**だけで解答を導けるモデルです [5]。モデルのアーキテクチャ上はレベル1と大きな差はありませんが、より大規模なモデルや改良されたトレーニング手法を通じて、数学の証明、プログラミング問題、専門知識の応用など複雑なタスクを**人間と同等の正確さで解ける**ことを目指します [5]。このため、モデルには単なる言語生成能力だけでなく、**チェイン・オブ・ソート（CoT）**のような逐次思考能力が組み込まれている場合があります。例えば、モデルが内部で論証手順を逐次的に展開してから最終回答を出力するような**思考の連鎖**の訓練が行われることがあります。また、誤った回答（幻覚）を減らすため、ファクトチェック的な自己検証能力や不確実な場合に保留したり根拠を示したりする挙動も盛り込まれます [7]。レベル2到達にはモデルのさらなるスケーリングだけでなく、高度なファインチューニング（例えば数学や論理推論問題データでの追加訓練）や、新しいモデル構造（長い文脈を扱うための拡張メモリ機構など）の導入が考えられます。なお入力は主にテキストですが、必要に応じ図表や数式をテキスト符号化して取り扱うこともあります（マルチモーダル入力はこの段階でも必須ではありません）。重要なのは、この段階のLLMは**外部の計算機能や知識ベースにアクセスせず**に高度な問題を解決できるという点です [5]。言い換えれば、知識と推論力のすべてを自分のパラメータ内に内包する「自己完結型」の頭脳として機能します。

アプリケーション層（構造と機能）： レベル2におけるアプリ層は、基本的な対話インターフェースはレベル1と同様ですが、モデルの高度な推論能力を活かすための追加構造が見られます。まず**ツール呼び出し**に関しては、レベル2では「ツール非依存」で問題解決する前提のため、原則として外部APIやデータベースへのア

クセスは行いません [5]。その代わり、アプリ層では**プロンプトエンジニアリング**によってモデルの推論力を最大限引き出す工夫がなされます。例えば「一緒に段階的に考えてみましょう」という隠れ指示を与えてモデルに詳細なステップを吐かせ、最後に答えだけ抽出してユーザに返す、といった制御です。これはモデル内部のチェイン・オブ・ソートを誘発し誤答を減らすテクニックです。また、モデルが返した回答の**自己検証**をアプリ層でサポートすることもあります。具体的には、同じ質問を言い回しを変えてもう一度モデルに聞いて結果を突き合わせたり、モデルに「上の回答は妥当か？」と評価させたりする処理を挟み、信頼度の高い解答を選択する、といった手法です。これらはモデルの**幻覚問題**や不確実性を補うためのアプリ側対策であり、レベル2 AIが求める高い正確性を担保する狙いがあります [7]。**状態管理**については、対話型で連続して難問を解くようなケースでは、セッション内での前の議論内容を保持しつつ、新たな推論に活かす必要があります。アプリ層は引き続き直近の会話履歴をコンテキストに入れますが、レベル2では個々の問題が長大になる可能性があるため、過去対話を要約してコンテキストを圧縮する、あるいはユーザごとにセッションを分け大きな一問一答形式にするなどの工夫が考えられます。UI/UXの観点では、ユーザはレベル1以上に専門的な質問（例えば医学的助言や法律問題の相談など）を投げかけることが想定されるため、**説明可能性**も重視されます。そこでアプリ層はモデルから引き出した推論過程をユーザに分かりやすく可視化・要約して提示するなど、専門家アシスタント的なUIを提供することもあります。例えば「この結論に至った根拠：○○」のような部分を回答に添えることが考えられます（モデル自身が根拠文章を生成できる場合）。

両層のインターフェース設計: レベル2ではインターフェースは引き続きテキストベースですが、その使い方が高度化します。アプリ⇔LLM間の通信自体はテキストのプロンプトと応答という形ですが、先述のようにアプリ側で**隠れ命令**や追加質問を挿入し、モデルの内部思考を誘導・確認するという**マルチターン対話的なプロンプト設計**が行われる点がレベル1と異なります。例えば、一度のAPI呼び出し内で「Step 1: 問題を分析しなさい」「Step 2: 解答を計算しなさい」とシステムメッセージで段階を踏ませ、モデルに構造化された回答をさせるプロンプトを構築するケースもあります。また場合によっては、問題→解答を一度で出さず**対話的手順**で解かせる（ユーザから見ればモデルが自問自答しながら解くように見える）インターフェースもありえます。このような場合、アプリ層はその制御のため**ループ処理**や**テンプレート**を持ち、モデルの部分的な出力をチェックして次の入力を組み立て直すといったロジックを実装します。制御方式として、モデルの暴走や誤答を防ぐため回答に制約をかけることもあります。例えば「根拠を示さずに断定しないように」とプロンプトで指示したり、出力内容をアプリ側で検証し怪しい場合は「もう一度考えてください」と再度モデルに投げ返したりします。とはいえ、外部ツールを使わない範囲ではアプリ層ができることも限られるため、基本的にはモデル主体で推論が進み、アプリはその補助・結果整形を行う関係です。

進化と依存関係の推移: この段階への進化は**モデル能力の飛躍**がカギとなりました。レベル1からレベル2への移行期にあたる現在（2024〜2025年時点）、OpenAIはGPT-4の出現によって第二段階に差し掛かっていると述べています [5] [8]。GPT-4は高度な試験（バー試験や大学入試レベルの問題）で人間上位層に匹敵する成績を収め、より深い推論力を示しました [9] [4]。これはモデルのスケーリングや訓練手法改良によるLLM層の進歩が原動力となった部分です。一方、アプリケーション層もモデルの性能を引き出すべく、前述のようなプロンプト手法や検証機構を発展させ、**モデル主導**から**モデル＋アプリ協調**へと進化しています。依存関係としては、まず基盤となるモデルが十分に強くないとレベル2の高度な推論は実現できず、この意味でLLM層が先導します。しかしモデルが強力になるほど、それをどう安全かつ効果的に使うかというアプリ側の役割も増大しました。特に幻覚問題への対処や専門領域への適用など、モデル単体では難しい部分をアプリ層が補完する形での共進化が見られます [7]。総じて、レベル2では「モデル性能の向上が先、応用実装が後から追いつく」構図でしたが、徐々にアプリケーション側からモデルに追加機能（例えば思考プロンプトテンプレート）を埋め込むようなアプローチが重要になり、両者の相互依存が強まっています。

レベル3：エージェント（Agents）

LLM層（モデルの能力・設計）： レベル3では、AIが**エージェント**として振る舞います。すなわち、人間からの指示に対し**自律的に行動を計画・実行できるAI**です [10] [11]。この段階のLLM層には、単にテキストを出力するだけでなく**「次に何をすべきか」を判断する能力**が求められます [12]。モデル自体は言語モデルですが、外部のツールや環境とのインタラクションを見越した特殊な出力フォーマットを扱えるよう調整されます。

例えば、OpenAIのGPT-4では**関数呼び出し（Function　Calling）**機能が導入され、モデルが特定のJSONスキーマで出力を生成すると、対応する外部関数が実行される仕組みが備わりました [13]。こうした拡張により、LLMは「〇〇の検索を行う」「計算する」「ファイルを書き込む」といった**行動コマンドをテキスト経由で発行**できます。またモデルには、与えられた最終目標を達成するため**内部で計画立案や意思決定**を行う能力が期待されます。これには、一度の入力出力で完結せず逐次的に考えるReAct手法（Reason + Act）や、長期目標を管理するためのメタ認知的プロンプト（例えば「目標X達成までのステップを列挙せよ」）を理解・生成できることが重要です。モデルアーキテクチャそのものはTransformerベースで変わりませんが、**長大なコンテキスト**（何十KBもの履歴）を保持できるモデルが好ましいです。なぜならエージェントでは複数ターンに渡る対話・ツール使用履歴をコンテキストに入れておく必要があるからです。また場合によっては画像や表など非テキスト情報も扱う必要が生じるため、モデルが**マルチモーダル**能力（視覚入力の解析など）を備えるか、視覚専門モデルと協調できることも要求されます。総じて、レベル3のLLM層は高い推論力に加え「行動指示文」を適切に生成する訓練が施され、環境とやりとりしながら目標達成を目指す**エージェンシー**（主体性）を帯びたモデルと言えます。

アプリケーション層（構造と機能）： レベル3の到達には、アプリケーション層の構築する**エージェント環境**が決定的な役割を果たします [14]。この層ではまず、モデルに使用させる**ツール類（外部関数やAPI、データベース照会、ブラウザなど）**を定義し、その使い方をモデルに教えるためのインターフェースを提供します。例えば「ウェブ検索」「電卓計算」「ファイル読み書き」等の機能をそれぞれ関数として実装し、プロンプト中で `<関数名>(引数)` の形式で呼び出せるようモデルに説明します。アプリ層はモデルからそのような**アクション出力**が出てきた際にそれをパースし、対応する実処理を行い、結果をテキストにしてモデルにフィードバックします [15] [16]。この一連の流れ（**ツール使用のループ**）を管理するのがアプリ層の中心的な機能です [12]。例えばReActフレームワークでは、アプリが「Thought（思考）：…」「Action：検索(keyword)」というモデル出力を解析し、実際に検索APIを叩き、その結果テキストをモデルに次の入力として与えます。このようにエージェント実行時には**ループ型の制御フロー**がアプリ側に実装されます。加えて、エージェントが長期に動き続けられるよう**状態管理**が強化されます。具体的には、各ステップのツール使用履歴や中間結果を記録し、モデルに再入力するコンテキストを動的に組み立て直す仕組みが必要です。メモリ機構も拡充され、過去の重要情報をベクトルデータベース等に保存し必要時に検索してモデルに供給する**長期メモリ**が導入されます [17] [18]。例えばAuto-GPTなどの自律エージェントでは、得られたウェブ情報を要約してメモリに蓄積し、後の判断に利用しています。UI/UX面では、エージェントが裏で何をしているかユーザが追跡できるよう**思考過程の可視化ログ**を表示するインターフェースが用いられることがあります。ユーザは最終回答を待つだけでなく、途中経過（どのツールをどう使ったか）を確認できるため、長時間動作するエージェントへの信頼性向上につながります。また必要に応じてユーザが介入（Yes/Noの指示や追加ヒントを与える等）できるUIを設ける場合もあります [19]。このように、アプリ層は**エージェントのオーケストレーター**（指揮役）として振る舞い、LLMの出力に基づいて外部環境とのインタラクション全般を請け負います。

両層のインターフェース設計: レベル3ではLLMとアプリの通信方法は大きく高度化・多層化します。基本はテキストのやりとりですが、モデルが**アクション出力**と**対話出力**を使い分けられるよう、プロンプト設計とパーサが工夫されます。例えばプロンプトに「利用可能なツール一覧」と使用フォーマットの例をシステムメッセージで埋め込み、モデルが「 `関数名(args)` 」の形式で文章を出力したらそれを関数呼び出しと解釈する約束事を共有します [13]。OpenAIの関数呼び出しAPIでは、この部分がモデルとアプリ間で明示的な**JSONインターフェース**として定義されており、モデルは決められたJSONスキーマに沿って出力することで直接アプリ側の関数をトリガできます [13]。このように**フォーマット化された通信**が増えるのが特徴です [20]。通信は対話の各ターンで繰り返され、モデル→アプリへの命令、アプリ→モデルへの結果報告がインタラクティブに行われます。制御方式の観点では、アプリ層がモデルの行動にブレーキをかけたり方針を与えたりする仕組みも不可欠です。例えば許可されていないツール使用要求が出た場合アプリ側でエラー応答を返す、一定回数以上ループしたら強制終了させる、機密データにアクセスしようとしたらマスクする等の**ガードレール**を実装します。また、人間の許可なしに重要処理を実行しないよう、アプリが途中でユーザ承認を要求することもあります。LLM側から見ると、自身が何度も対話を繰り返すうちに与えられるシステムメッセージやツール結果が動的に変化していくため、より**能動的対話エージェント**として振る舞うことになります。まとめると、レベル3では自然言語が**「汎用インターフェース」**の役割を果たし、LLMがそれを介して他のシステムを自在に操作するためのプロトコルが確立されます [21] [22]。

進化と依存関係の推移: レベル2（純粋推論型）からレベル3（自律行動型）への移行は、**アプリケーション層のイノベーション**によって切り拓かれた側面が大きいです。すなわち、LLMそのものの能力向上に加え、「その能力をどう外界の操作につなげるか」というソフトウェアアーキテクチャ上の工夫が相乗して実現しました。GPT-4が登場した当初、それ自体は強力な推論者でしたが外部ツールを使うことはできませんでした。しかし開発者たちは、モデルに特殊フォーマットを出力させて関数呼び出しするアイデアや、ループを組んで目標達成まで実行させるフレームワーク（例えばAuto-GPTやBabyAGI）を次々に試みました。その結果、モデル側もそれに適応するよう微調整され、OpenAIも公式にプラグインや関数呼び出しといった機能をモデルに組み込みました [13]。これは**アプリ層が先導しモデル層が追随した例**と言えます。もっとも、OpenAI自身も「強力なエージェントには頑健な推論能力が前提」と述べており [14]、レベル3達成にはレベル2相当の知性が不可欠です。したがって、まずモデルが高い知的水準に達し（レベル2へ）、それを踏まえてアプリ側が環境とのインタフェースを整備し（レベル3へ）という**段階的共進化**が起きました。依存関係としては当初モデル能力がボトルネックでしたが、モデルが一定水準に達すると今度は「どんなツールを与えどう使わせるか」というアプリ設計がボトルネックになります。レベル3の現在では、各種エージェント実装（LangChain等のフレームワーク）が乱立しアプリ側の工夫が花盛りですが、同時にモデル側でも長大文脈対応やツール使用に最適化した新モデルが研究されています。今後はモデルとアプリの**協調的改良**（例えばモデルが自発的にツール選択・計画立案できるよう訓練しつつ、アプリ側もより高度なツール環境を用意する）が進むでしょう。一例として、Microsoftの提案した“HuggingGPT”は、LLMをコントローラとして画像・音声など多数の専門モデルを使い分けるエージェントアーキテクチャであり、複雑タスクに挑むためモデル層・アプリ層の協調進化の方向性を示しています [21] [22]。実運用面では、エージェントの暴走リスクへの対処など新たな課題も生じており、安全な制御手法（例えば人間の監督下での実行やポリシーベースの停止条件など）を含め、アプリケーション層の責務がますます重要となっています。

レベル4：イノベーター（Innovators）

LLM層（モデルの能力・設計）： レベル4の**「イノベーター」**は、AIが新しいアイデアや発見の創出に寄与できる段階です [23] [24]。LLM層には、人間の専門家を上回る創造性と問題解決力が期待されます。具体的には、単に既存知識から答えを導くだけでなく、**未知の問題に対して斬新な解決策を構想**したり、科学研究で新規の仮説を提案・検証したり、技術開発で発明を生み出したりする能力です [23]。このため、モデルは従来の言語・知識データに加え、科学技術・芸術など多岐にわたる分野の専門知識やパターンを学習している必要があります。モデルアーキテクチャとしては、**マルチモーダル**への対応が一層重要になります。例えば研究論文の図表を読み取ったり、設計図を出力したりするにはテキスト以外の情報処理能力が要るため、画像や数式、プログラムコードなど複数モードを統合的に扱えるモデルが求められます。実際、GPT-4は画像入力を受け付けるビジョン機能を持ち、視覚情報に基づく推論も可能になりました [25]。将来的なLLM（GPT-5以降）は音声や動画、センサーデータまで含めた統合モデルになる可能性があります [26]。さらに、モデル内部に大容量の**長期記憶機構**を組み込む試みも考えられます。レベル3までは長期記憶は主にアプリ層で管理していましたが、レベル4ではモデル自体が過去の知識を動的に保持・参照したり、新知識を継続学習できるようなアーキテクチャ（例えばメモリネットワーク、反復自己学習機構など）の研究が進むでしょう。モデル能力面では、例えばDeepMindのAlphaTensorというシステムは自ら新しいアルゴリズム（行列演算の高速手法）を発見しました [27]。これは強化学習エージェントとLLMは異なりますが、AIが未知の解法を創出する一例であり、レベル4が指向する「発明するAI」の可能性を示しています [27]。LLMにもこのような創発的創造力を発揮させるには、単なる教師あり学習を越え**自律的な試行錯誤**や**多段の思考戦略**を取り入れた学習が鍵となるでしょう（例えば自己対話しながら仮説検証するような訓練など）。総じて、レベル4のLLM層は極めて高度で汎用な知的能力を備え、複数領域にまたがる問題に対して人間の発想を超えるアウトプットを生み出せる存在となります。

アプリケーション層（構造と機能）： レベル4を実現・支えるアプリケーション層は、エージェント機構をさらに発展させ、**創造的プロセスを管理するプラットフォーム**となります。まずツール面では、科学技術計算、シミュレーション、データ分析、CAD設計、あるいは他のAIモデルの呼び出しなど、**高度で多様なツール群**を統合する必要があります。例えば、新薬分子を発見するなら化学シミュレータやデータベース検索ツール、設計した分子の評価には別のAIモデル（例えばAlphaFoldのようなタンパク質構造予測モデル）を呼び出

す、といった具合に、創造的タスク固有の専門ツールとの連携が不可欠です。アプリ層はそうしたツールをカタログ化し、適時にLLMエージェントが使えるよう**柔軟なAPIインターフェース**を提供します。状態管理もさらに複雑化します。レベル4のタスクは**一度で完結しない長期プロジェクト**（例えば新技術の研究開発は数ヶ月〜年単位）になる可能性があります。そのため、エージェントの状態（進行中のアイデア、検証した結果、残る課題など）を長期間保存・更新し続ける**プロジェクトメモリ**が必要です。具体的には、進行中の計画や得られたデータをデータベースやナレッジグラフに構造化して蓄積し、エージェントが適宜参照・更新できるようにします。UI/UXについては、人間の研究者や開発者がAIと協働できる環境が望まれます。例えば研究助手AIであれば、人間が実験の方向性を指示し、AIが候補を挙げ試験計画を立て、人間がそれを承認してAIが実行・結果分析し…という**共同作業インターフェース**が考えられます。AIが提案する新発想を人間が確認・評価できるよう、ダッシュボード形式でAIの思考過程や理由付けを閲覧できる仕組みも求められるでしょう。エージェンシー構築の観点では、レベル4ではエージェントが一層**自主的**に目標を設定・修正しうるようになります。アプリ層はAIが自律的にサブゴールを立てたり計画を変更したりすることを許容しつつ、それを全体目的と整合させる**ガバナンス機構**を備えます。例えば「目的Aを達成するために中間目標XとYを設定し実行します」とAIが決めた場合、その記録を残し、進捗をモニタし、必要なら人間が介入できるようにします。また安全性・倫理面で、創造的AIがリスクのあるアイデア（危険な発明など）を提案した際に制限をかける**制御モジュール**も組み込まれるでしょう。つまりアプリケーション層は、単一エージェントを超えて**AI研究者チーム全体を統括するマネージャー**のような役割を果たします。実装例として、Stanford大学のGenerative Agentsの研究では25体のエージェントが架空の街で相互作用し、新たな行動（パーティの企画など）を自発的に生み出しました。このように**マルチエージェント協調**による創発的な結果も確認されており、レベル4では必要に応じ複数のLLMエージェントがチームを組んで問題解決にあたる構成も考えられます。

両層のインターフェース設計: レベル4では、LLMエージェントと多様なツール・他のAI・人間とが**ハブ的に連結**された複雑なインターフェースとなります。コミュニケーション手法はテキストベースが中心ですが、含まれる情報はテキストにとどまりません。例えば、実験データの表はCSVからテキスト要約されて渡され、画像はキャプションや解析結果として渡されるか、画像自体を埋め込みベクトルにしてモデルに与えるなどの工夫が取られます。LLMが他の機械学習モデル（画像生成モデルなど）を呼び出す場合、その入力出力もテキストによるメタ記述でやり取りすることになります（HuggingGPTのように、「画像分類モデルを呼び出し**[画像ID]**を解析せよ→結果テキストを取得」といった一連をLLMがシナリオとして実行 [22] [28] ）。通信の頻度・量も大規模になり、並行的な処理が走ることもあり得ます。制御方式は、レベル3まで以上に慎重であるべきです。創造的AIは想定外のアウトプット（例: 倫理に反する提案）をする可能性があるため、アプリ側で**内容審査**を組み込み、不適切な提案はフィルタあるいは要修正箇所を指摘して再実行させるといったフィードバックループが設計されます。また、複数のエージェントが動く場合、それらの通信には**プロトコル**を定め、競合や無限ループを避ける調停役が必要です。例えば一つの共有メモリに各エージェントが書き込む場合、競合状態を防ぐロック機構をアプリ層が管理するといったことです。人間とのインターフェースも重要で、AIが出した結論や発明に対して人間が評価・承認・修正を行える双方向性が求められます。これはUI上のワークフローに組み込まれるでしょう。総合すると、レベル4ではインターフェース設計は「AI vs 環境（ツール・データ）」の対話に加え「AI vs AI（協調）」および「AI vs 人間（協働）」の三者間インタラクションを含むものとなり、その制御には高度なソフトウェアアーキテクチャとガバナンスが必要です。

進化と依存関係の推移:　レベル4は現在の延長上にはまだ存在しない**フロンティア**であり、その進化はモデル・アプリ両層の更なる共進化によって達成されると考えられます。まずモデル層のリードとして、GPT-4を超える次世代モデル群（例: GPT-5やGoogle Gemini等）がより汎用的な知能と創造性を示し始めることが重要です。その上で、アプリケーション層がそれらのモデルを研究開発プロセスに組み込む実験が進むでしょう。おそらく初期には、限定された領域でAIが画期的アイデアを出す事例（例えば新薬設計AIが有望な化合物を発見）が散発的に現れます。それらは人間研究者との協働環境（アプリ層）が整っていて初めて実現するでしょう。したがって、この段階では**どちらが先導するというより双方の準備が整う必要**があります。強いて言えば、モデル層の大幅なブレークスルー（創造性の発現）がなければレベル4には到達できません。一方で、たとえモデルに潜在的創造力があっても、それを現実世界の価値に結び付ける仕組みがなければ宝の持ち腐れです。そのため、産業界・研究界がAIを実験計画や設計プロセスに組み込むプラットフォームを構築し、AIに**試行と検証の場**を提供する流れが加速するでしょう。依存関係はダイナミックで、モデルが進歩すればそれを最大限活用するアプリが現れ、アプリの要求に応じて次のモデルがさらに強化されるというループ

が想定されます。安全面・倫理面の問題もより深刻になるため、人間社会の規制・ルール（外部要因）もシステム設計に影響を与えるでしょう。総じてレベル4は、真に汎用的な創造性を持つAI（AGIに近い存在）へのプレ段階であり、モデルとアプリケーションが**共同でイノベーションを生み出すシステム**が形作られていく段階と位置付けられます。

レベル5：組織（Organization/Organizers）

LLM層（モデルの能力・設計）： レベル5は、**AIが一つの組織の業務を丸ごと遂行できる**究極の段階であり、しばしば**AGI（汎用人工知能）**と同義的に語られます [1] [29]。この段階のLLM層（もはや単一のモデルか複数モデルの集合か議論がありますが）は、人間のあらゆる知的労働を代替・上回る能力を備えます [1]。知識範囲は全領域に及び、推論・計画は短期から長期まで階層的にこなし、創造性や判断力も一流の専門家集団に匹敵、さらには人間には難しい巨大システムの最適化や同時並行タスク管理も遂行できるでしょう。モデルアーキテクチャとして考えられるのは、一つには**巨大な統合モデル**（例えば数十兆パラメータ級でマルチモーダルかつ長期記憶内蔵のようなもの）が組織全体の知能として振る舞うケースです。もう一つは、複数の高度なLLMや専門モデルが**協調動作するモジュラー構造**です。後者は人間の組織が部署ごとに専門性を持つように、AIも「経営判断モジュール」「財務モジュール」「設計モジュール」等を持ち、それらが通信し合い全体として一つの組織AIを形成するイメージです。どちらにせよ、モデル層には自己改善・学習する能力が不可欠と考えられます。組織運営では環境が常に変化するため、AIが自律的に新知識を取り入れ自分のモデルパラメータや判断基準を更新していく（オンライン学習やフィードバックループによる自己成長）仕組みが求められます。これは現在の静的な事前学習モデルを超えた概念であり、メタラーニングや継続学習の研究領域です。また**社会性・対人能力**も重要な要素です。組織を運営するAIは、人間の取引先や顧客と交渉・協働する場面も出るでしょう。そのため自然言語だけでなく、感情的知性や倫理的判断、説得や交渉スキルまでモデルに備わっている必要があります。技術的には、強力なLLMにRLHFならぬ**人間社会からのフィードバック**を大量に与えて社会規範に沿った振る舞いを覚えさせる、複数モデル間で役割分担とコミュニケーションを学習させる、といった方向になるでしょう。レベル5のモデル層は、もはや単なる「言語モデル」の枠を超え、人類レベル以上の知的複合体となります。

アプリケーション層（構造と機能）： レベル5のアプリケーション層は、AIが実際に組織業務を遂行できるよう**包括的な業務インフラ**を提供します。具体的には、企業で人間が使うあらゆるITシステム・ツールとAIを統合し、AIが自由にアクセス・操作できるようにする必要があります。顧客対応ならメール・チャットシステム、経理なら財務会計システム、製造業なら工場のIoT制御や在庫管理データベース、研究開発なら実験設備の遠隔操作、といった具合です。アプリ層はそれらを統一的な**API群**として整備し、AIエージェントが組織内外のリソースをシームレスに使えるようにします。また、社内の知識や過去データはすべてAIが参照できる**統合ナレッジベース**として管理します。大量のドキュメントやデータも、アプリ層でベクトルDB化や知識グラフ化され、AIが質問して瞬時に引き出せるようになります。加えて、**複数エージェントの編成**も重要です。単一のAIが全業務を細部まで行うのではなく、役割ごとにエージェントを分けチームを構成させる方が効率的でしょう。アプリ層は「AIマネージャー」と複数の「AI部下」を作り、例えばマネージャーが指示を出し部下エージェントが個別タスク（資料作成や分析等）を実行し、結果をマネージャーが統合する、といった**マルチエージェント体制**を敷くことが考えられます。そのためのエージェントプラットフォーム（エージェント同士の通信、タスク割り当て、結果集約）もアプリ層で構築されます。状態管理は極めて広範になります。プロジェクト管理、スケジュール管理、リスク管理など、人間の経営管理が行うことをすべてデジタルにトレースし、AIがアクセスできる状態で維持します。言わば組織全体の**デジタルツイン**をリアルタイムで更新し続けるようなシステムです。UI/UXに関しては、原則AIが全て自動で行うので人間ユーザ向けのUIは不要…とも考えられますが、現実には人間のオーナーや監督者が存在しうるため、**AIによる意思決定の可視化・説明**を行うダッシュボードが提供されるでしょう。たとえば、「AI　CEO」が経営判断を下す際には、その根拠や見通しを人間の取締役に説明する責任があります。また外部の人間（顧客や取引相手）に対しては、人間と同様のインターフェース（メール文や対話など）で対応するため、その生成も行います。アプリ層はAIが作成した文章・資料を確認し必要なら修正提案するなど、**対外接点の品質管理**も行うかもしれません。最後に、**エージェンシー構築**の仕上げとして、組織AIには法的・倫理的制約を順守させる強力な**ガバナンス/コンプライアンスモジュール**が不可欠です。例えば勝手に違法な契約を結ばないよう契約書類生成は法律チェックAIが介在するとか、大きな資金移動は必ず人間の許可を要するワークフローにするといった**統制策**をアプリ層で敷き

ます。総合的に見て、レベル5のアプリケーション層はAIが**組織のOS(オペレーティングシステム)**として機能するための全環境を用意し、AIの判断・行動を安全に実世界へブリッジする極めて複雑な枠組みとなります。

両層のインターフェース設計: レベル5では、LLM（あるいは複数LLM）とアプリ層のインターフェースは、人間の組織における意思疎通と類似したものになります。AIマネージャーとAI部下エージェント間は自然言語や構造化メッセージでタスク指示・報告を交わすでしょうし、AIと各種ツール類のやり取りは基本的にレベル3・4同様API経由で機械的に行われます。ただ規模が桁違いに大きいため、通信には**優先度管理**や**スケジューリング**が必要です。無数の指示を同時並行的に処理する際、リソースの取り合いや整合性の問題が出ないよう、アプリ層がトランザクション管理やメッセージキューを用いて調停します。制御方式としては、最終防衛ラインとして**人間の関与**をどこまで残すかが重要です。完全自律AI組織では人間は一切関与しない可能性もありますが、多くの現実シナリオでは人間がオーバーライトできるスイッチや、人間への定期報告と承認プロセスが含まれるでしょう。インターフェース設計にはそうした**人的インタラクションのフック**も組み込まれます。例えばAIが立てた経営戦略プランを自動で社内掲示板に投稿し、人間役員がフィードバックできる、といった仕組みです。さらに、複数の組織AI同士がやり取りする場面（経済活動において複数企業のAIが交渉する等）も想定すると、AI間の共通インターフェース（商取引プロトコルなど）も社会的に標準化されるかもしれません。要するに、レベル5ではAIを**一個の意思決定主体**として扱うためのあらゆるインターフェースが統合され、人間社会との接点も内包した非常に高度な通信・制御体系が完成すると考えられます。

進化と依存関係の推移: レベル5は最終目標であり、到達にはなお多くの技術的ブレークスルーが必要とされています [1] [29]。その進化は段階4までの延長上にありますが、特に**モデル層の飛躍的進化**が要となるでしょう。すなわち、真の汎用人工知能（人間と同等かそれ以上の幅広い知能）が生まれることが前提条件です。OpenAIはこのAGI実現を10年以内と楽観視していますが [30]、専門家の見解は様々です。モデル層がAGIに近づくにつれ、逆にアプリケーション層の設計も**より重大**になります。というのも、極めて強力なAIを野放図に動かすわけにはいかず、適切に制御・調整して人類に利する形で働かせる必要があるからです。したがってレベル5では、安全性・制御の観点からアプリ層が主導権を握りつつ、モデル層の能力を最大化するという**二律背反的な共進化**が求められます。どちらが先導するかで言えば、AGIレベルの知性は人類の発明の中でも未知の領域であり、モデル研究者がそれを創り出した後、エンジニアが制御策を慌てて講じる…という展開も考えられます。一方で、安全なAGIを目指すなら制御アーキテクチャ（アプリ層）の設計思想をあらかじめ盛り込んだ上でモデルを育てる必要も説かれています。現在提案されている自己監督付きのLLMや人間模倣による価値アライメントなどの手法はその例です。最終的に、モデル層とアプリ層の境界はレベル5では曖昧になる可能性があります。モデル自体が一種のプラットフォーム化し、他のモデルを内包・管理するかもしれませんし、アプリ層の多くがモデル内に取り込まれる（例えばツール使用も内在化する）可能性もあります。要は**システム全体が一体となった知的エコシステム**となり、人間はその全体を相手にすることになるでしょう。OpenAIの5段階ロードマップは、自動運転のレベル分けになぞらえれば段階的アプローチですが、レベル5への道のりは連続というより質的飛躍を伴うと予想されます。それだけに、現在のレベル3前後の技術蓄積・教訓を活かしながら、モデルとシステムのデザイン両面から慎重にAGIを構築していくことが重要となります。

補足要約（一般読者向け）

以上のように、AI能力の**5段階**それぞれで「頭脳」に相当するLLMと、それを使う「周辺システム」の役割は大きく変化します。簡潔にまとめると:

- **レベル1（チャットボット）:** 人とおしゃべりするAIの段階です。言語モデルが文章を理解・生成し、アプリは主にチャット画面などのインターフェースを提供します。まだ道具は使わず、会話だけで対応します [2]。

- **レベル2（推論者）:** 専門家並みの問題解決力を持ったAIです。モデルが論理的に考えて正確な答えを出せるようになり、アプリ側はモデルの推論を引き出す工夫や誤りチェックを行います [5] [7]。この段階でも外部ツールに頼らず、モデルの内部知識で回答します。

- **レベル3（エージェント）：** AIが自律行動できる段階です。モデルは「〜を検索」「〜を計算」といった命令をテキストで出し、アプリがその命令に従い外部ツールを実行します [12]。こうしてAI自らインターネット検索やアプリ操作を行い、ユーザの目的達成まで何ステップも動きます [10]。アプリはそのための環境（ツール群やループ制御）を整備し、AIの行動を管理します。

- **レベル4（イノベーター）：** AIが新しいアイデアを生み出す段階です。モデルは創造的になり、科学発見や新製品の発明にも寄与できます [23]。アプリは研究開発プラットフォームのような役割を果たし、AIにシミュレーションやデータ分析の手段を与えて創造プロセスを支援します。AIと人間研究者が協働しやすい環境も用意されます。

- **レベル5（組織）：**AIが丸ごと一社分の仕事をこなす段階、すなわちAGIです [1]。非常に高性能なモデル群が、人間のように計画を立て決定を下します。アプリ層は会社のあらゆるシステム（メール、財務、製造設備など）とAIを繋ぎ、AIがそれらを自由に操作できるようにします。安全管理も厳重に行い、必要なら人間が監督します。要するに、AIが社長から社員まで務めるようなもので、人間は最終チェックや目標設定だけ関与するイメージです。

各レベルを通じて、**LLMそのものの進化（より賢く汎用に）**と、**それを取り巻くシステムの進化（より多機能に統合的に）**が車の両輪となっています。初期のレベルではモデルの言語能力向上が主導的でしたが、高度なレベルではモデルの賢さと同じくらい、周辺アプリの設計が重要になります。最終的にはモデルとアプリの境界が薄れ、一体となった高度なAIシステムが人間の能力を超える範囲で活躍する――それがOpenAIの描くレベル5、ひいてはAGIの姿です。人類はそこに到達する過程で、安全性や倫理にも十分配慮しつつ、この二層の協調進化を促していく必要があると言えます [31]。

---

1 2 5 10 23 30 What Are OpenAI's Five Levels of AI -- And Where Are We Now?
https://theaiinsider.tech/2024/07/12/what-are-openais-five-levels-of-ai-and-where-are-we-now/

3 6 7 11 14 24 29 31 Understanding OpenAI’s Five Levels of AI Progress Towards AGI - Quinte Financial Technologies
https://quinteft.com/understanding-openais-five-levels-of-ai-progress-towards-agi/

4 9 25 GPT-4 | OpenAI
https://openai.com/index/gpt-4-research/

8 OpenAI Sets Levels to Track Progress Toward Superintelligent AI - Bloomberg
https://www.bloomberg.com/news/articles/2024-07-11/openai-sets-levels-to-track-progress-toward-superintelligent-ai

12 19 20 Overview
https://langchain-ai.github.io/langgraph/concepts/agentic_concepts/

13 An introduction to function calling and tool use - Apideck
https://www.apideck.com/blog/llm-tool-use-and-function-calling

15 16 Bridging AI and real-time data with tools - DEV Community
https://dev.to/aws/bridging-ai-and-real-time-data-with-tools-5e0m

17 18 Memory for agents
https://blog.langchain.dev/memory-for-agents/

21 22 28 HuggingGPT: Revolutionizing Complex AI Task Management with LLM Integration | Neural Notes
https://medium.com/neural-notes/hugginggpt-ba3af0decc4a

26 The Multi-modal, Multi-model, Multi-everything Future of AGI
https://www.latent.space/p/multimodal-gpt4

27 DeepMind's AI System That Can Discover New Algorithms | AI Business
https://aibusiness.com/ml/deepmind-s-ai-system-that-can-discover-new-algorithms