

RAG ベース AI エージェントを使用した 3D メッシュモデル作成の自動化 MESHLAB 編

株式会社アストライアーソフトウェア

2025年7月28日

概要



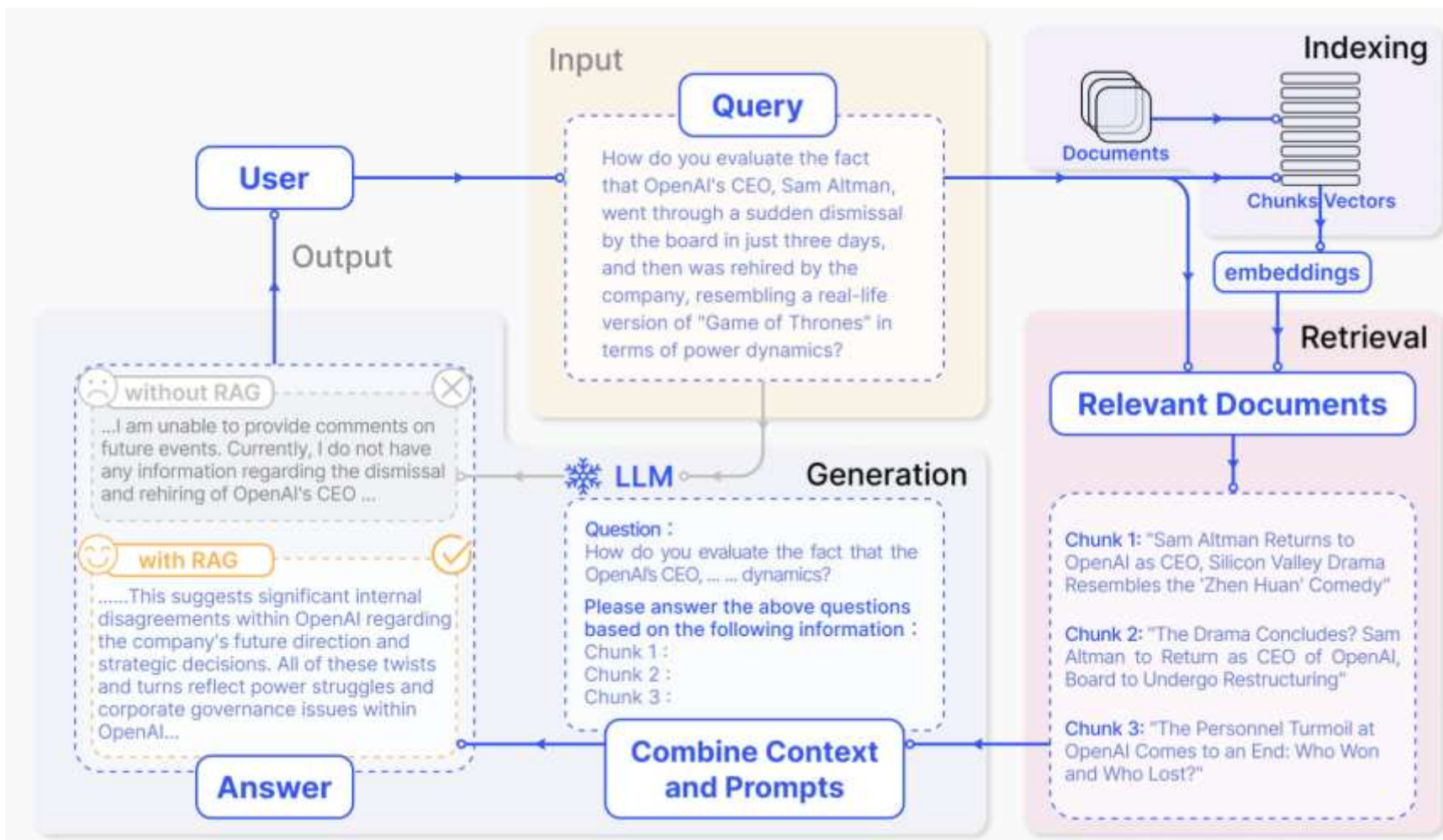
- メッシュモデルを作成するAIエージェントを、RAG (Retrieval-Augmented Generation) 技術とMCP (Model Context Protocol) サーバー技術を使って作成しました。
- RAGの知識データベースには、公開されている汎用メッシュモデル作成ツールMeshLabソフトウェアのAPI仕様書を使用し、またMeshlabのMCPサーバーも開発しました。
- この開発により、ユーザーはMeshlabのAPI仕様の知識がなくても、LLM (Large Language Models) が理解しうる適切なキーワード含めたプロンプトを入力するのみで、メッシュモデルの作成が可能になります。
- 作成したAIエージェントのデモを含めて紹介します。

RAGの概要



- **LLM**は事前学習データに依存しているため、初期状態は最新の最新情報を提供する能力が不足しています。
- **RAG**は、外部データベースから知識を取得し、組み込むことで、この情報ギャップを埋めます。

RAGの概要

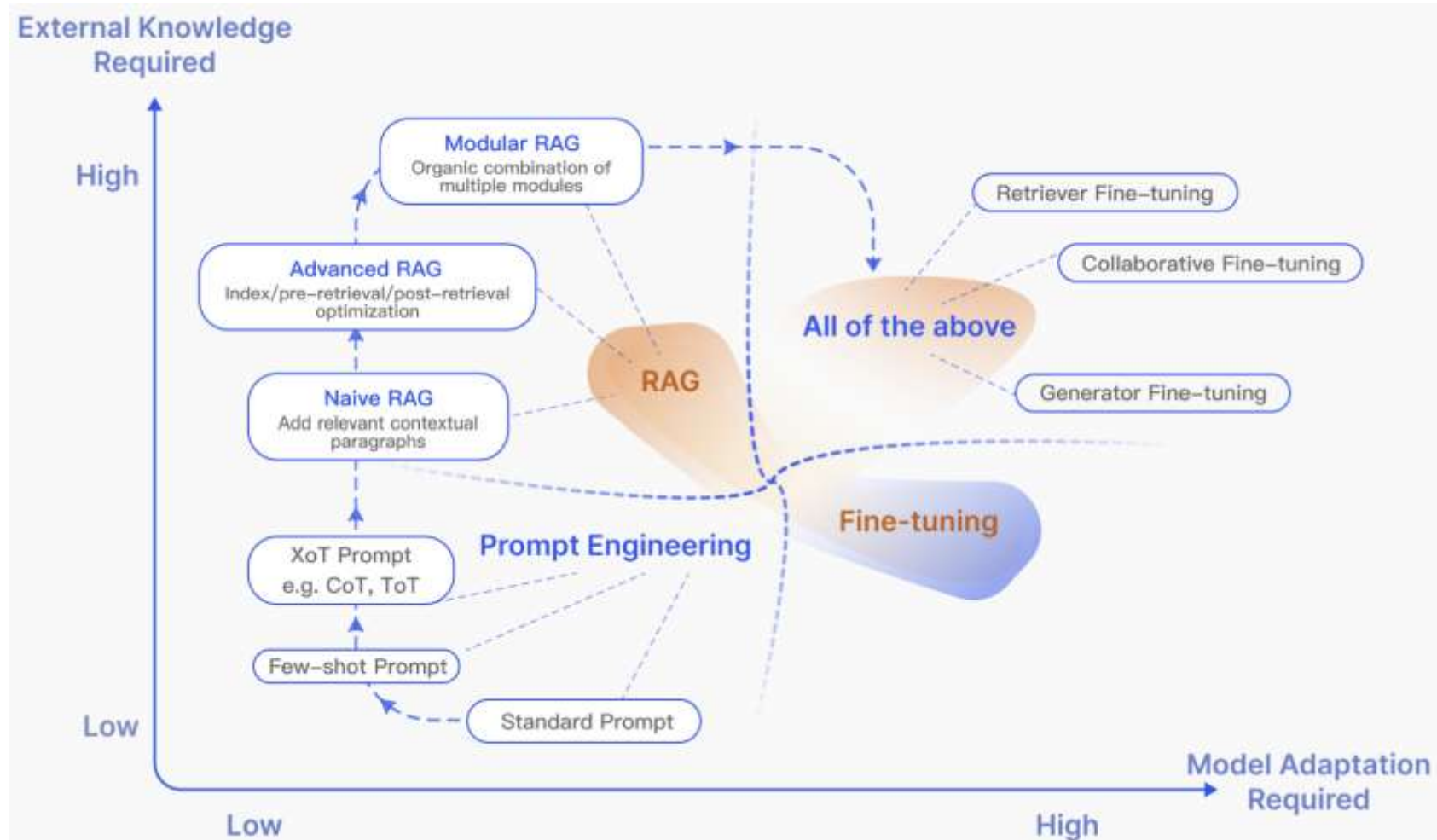


RAGとファインチューニングの比較



- **RAG**は、情報検索用にカスタマイズされた教科書をモデルに提供することに例えられ、正確な情報検索タスクに最適です。
- **RAG**は、リアルタイムの知識更新と、高い解釈可能性を備えた外部知識ソースの有効活用を提供することで、動的な環境において優れた性能を発揮します。ただしデータ取得に関して、推論レイテンシの高さ、倫理性などの検討事項が伴います。
- ファインチューニングは、学生が時間をかけて知識を習得していくことに例えられ、特定の構造、スタイル、フォーマットの再現が必要なシナリオに適しています。
- ファインチューニングはより静的であり、更新のために再トレーニングが必要になりますが、モデルの挙動とスタイルを詳細にカスタマイズできます。

RAGとファインチューニングの比較

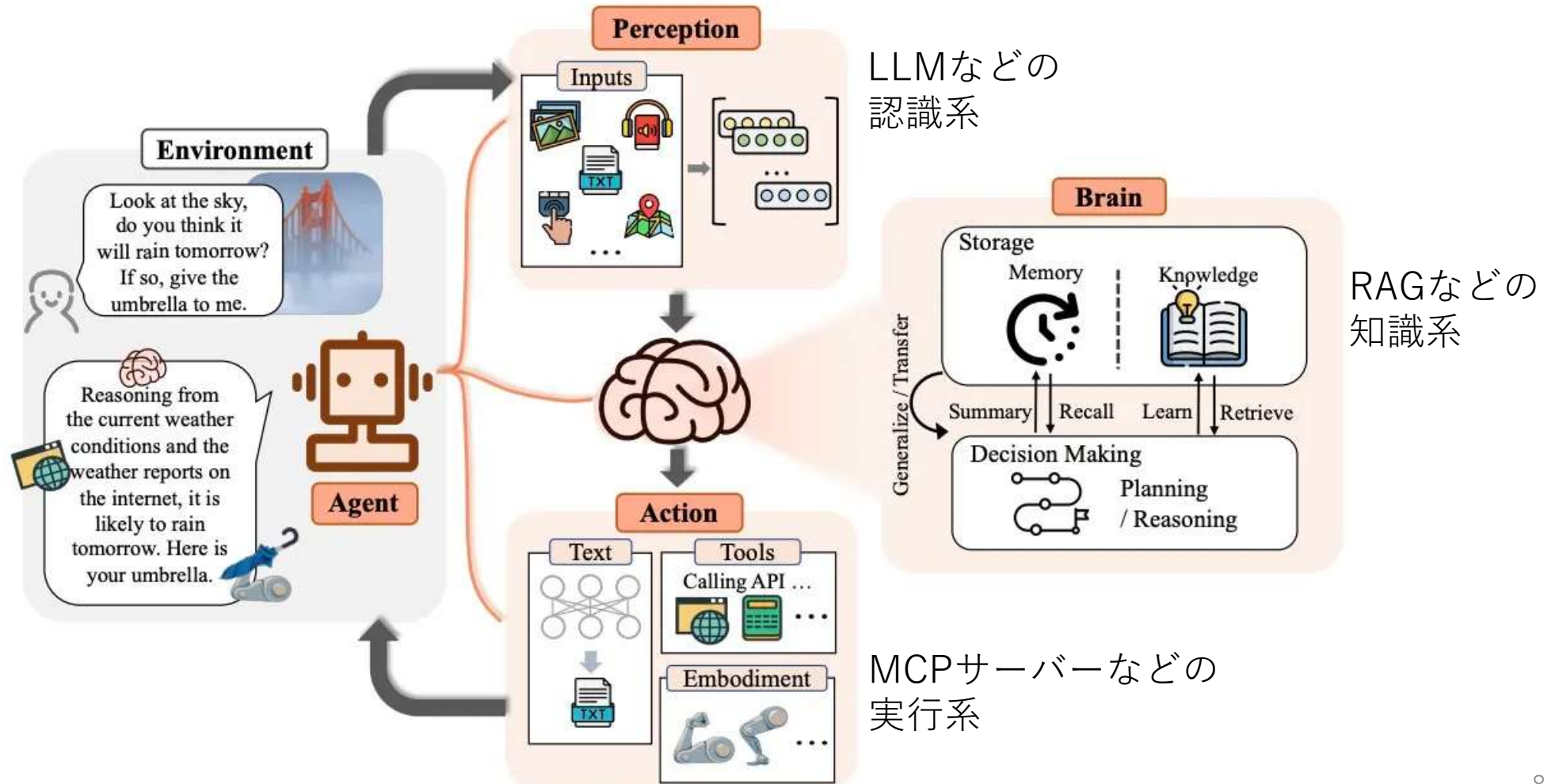


AIエージェント

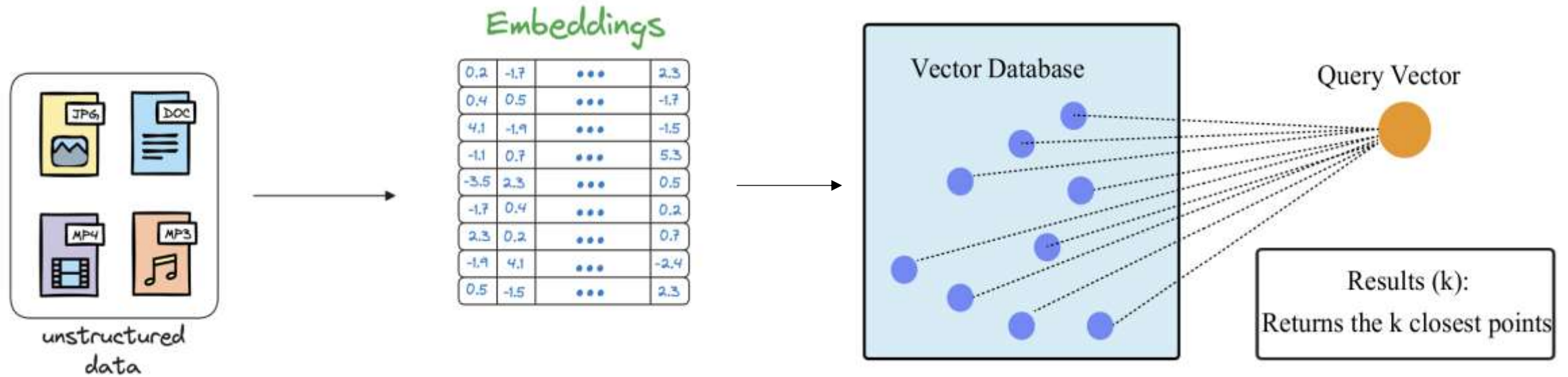


- AIエージェントとは、クエリに対する最適なシーケンスを言語モデルに決定させる特別なチェーンです。
- AIエージェントを使用する場合、開発者はユーザーの入力、利用可能なツール、そして望ましい結果を達成するための可能な中間ステップを提供します。すると、言語モデルはアプリケーションが実行可能な一連のアクションを返します。
- AIエージェントは、次ページに示すようにLLMなどの認識系、RAGなどの知識系、MCPサーバーなどの実行系の処理を順次行うことでクエリに対する処理を実施します。

AIエージェント



ベクトル埋め込み



- 非構造化データ（文書、画像、音声、動画など）を、その意味と関係性を捉える複数の多次元ベクトルに埋め込みます。
- 異なるデータタイプを多次元空間内の点として表現し、類似するデータポイントが密集する構造を実現します。これらの数値表現は、機械がデータをより効果的に理解・処理するのに役立ちます。

AIエージェントの作成事例

MeshLab機能の実行



- AIエージェントの作成事例として、一般公開されている汎用メッシュモデル作成ツールMeshLabソフトウェアを使ったメッシュモデル作成AIエージェントの事例を紹介します。
- Meshlabソフトウェアは、Python言語で利用できるAPIツールを準備しており、この仕様は以下のURLで公開されています。
 - https://pymeshlab.readthedocs.io/en/2022.2/filter_list.html
- 今回はこの仕様書をテキストファイルで入手し、RAGの知識データベースとしてベクトル埋め込みをしました。
- これによりPythonやPymeshlab APIの仕様を知らなくても、適切なプロンプトをLLMモデルに与えることによってMeshlabソフトウェアの機能を使用できるようになります。

Pymeshlab仕様とLLMプロンプト例



Pymeshlab 記述仕様書

compute_matrix_from_scaling_or_normalization

MeshLab filter name: 'Transform: Scale, Normalize'

Generate a matrix transformation that scale the mesh. The mesh can be also automatically scaled to a unit side box.

Parameters:

axisx : float = 1

X Axis: Scaling

axisy : float = 1

Y Axis: Scaling

axisz : float = 1

Z Axis: Scaling

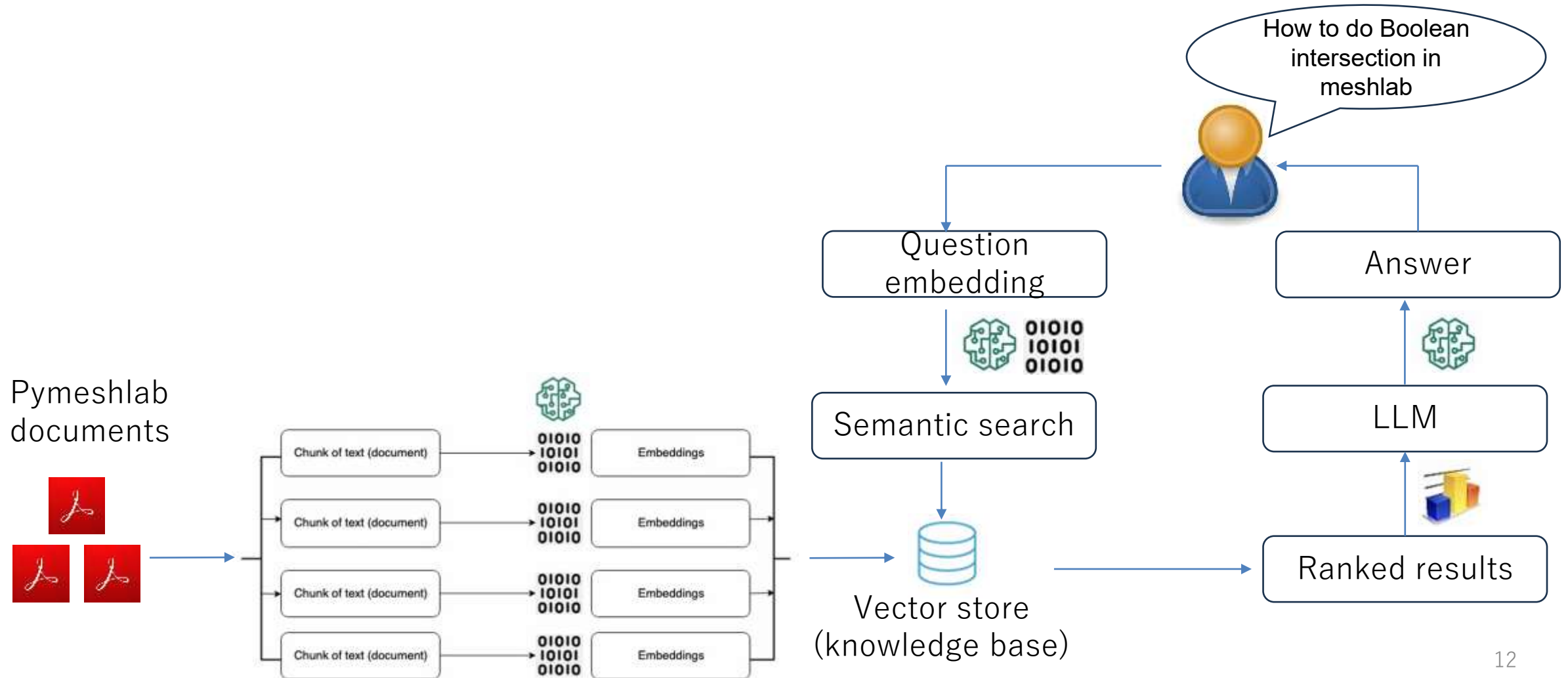
uniformflag : bool = True

Uniform Scaling: If selected an uniform scaling (the same for all the three axis) is applied (the X axis value is used)

一連のLLMプロンプト例

- create a cube size 50
- create a cube size 40
- transform scaling normalization model1 x axis 2 without uniform
- translate model2 unit 20 in x direction
- boolean difference between model0 and model1
- create a cone radius1 15, radius2 15, height 80
- boolean difference between model2 and model3
- isotropic remesh model4 length 3

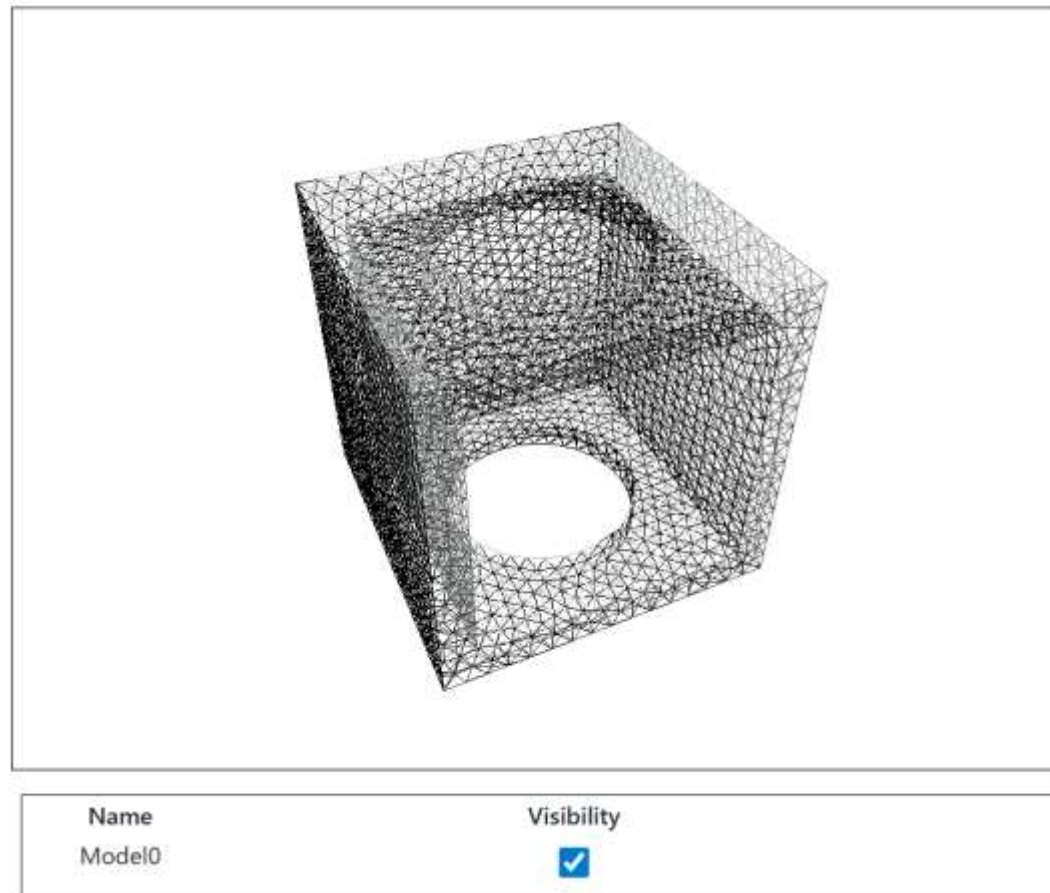
AIエージェントの処理フロー例 MeshLab機能の実行



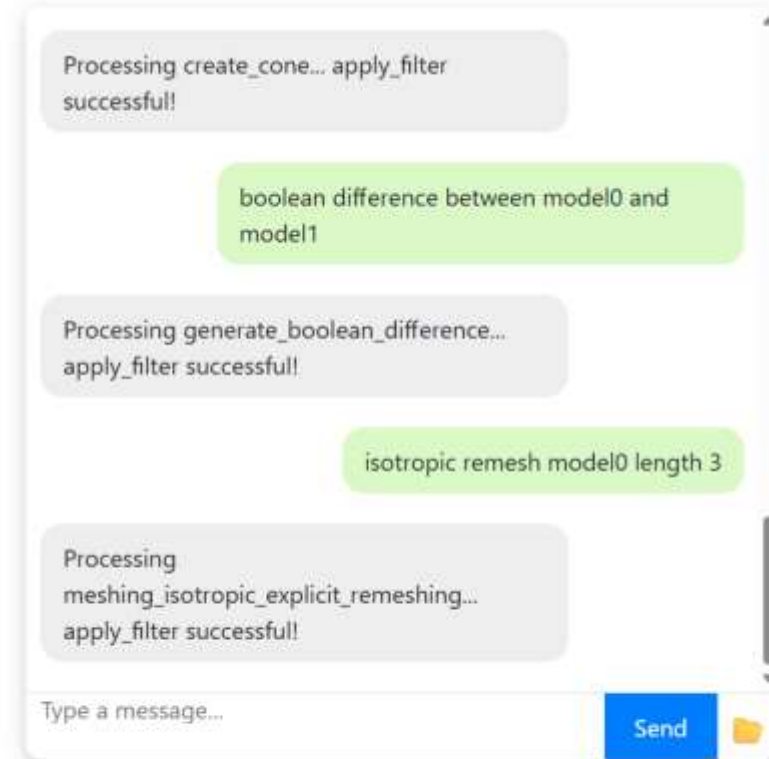
MeshLab AIエージェントの実行画面



メッシュモデル表示画面



LLMプロンプト入力画面



例題に使用したプロンプト（例）

- create a cube size 50
- create a cube size 40
- transform scaling normalization Model1 x direction 2 units without uniform
- translate Model1 20unit in x direction
- boolean difference between Model0 and Model1
- *ここで画面右下のDeleteボタンでmodel0 と model1を削除します。
- create a cone radius1 15, radius2 15, height 80
- boolean difference between Model0 and Model1
- *ここで画面右下のDeleteボタンでmodel0 と model1を削除します。
- isotropic remesh Model0 length 5



以上