

契約ネットプロトコルを用いたLLMマルチエージェントシステムによる複合サービス推薦

飛澤 佑季
情報理工学部
立命館大学
草津, 日本

村上 陽平
情報理工学部
立命館大学
草津, 日本

Abstract—自然言語で記述されたユーザの要件定義から複合サービスを構成する適切な Web サービス (API) を選択することは、サービス自動合成における重要課題である。近年、大規模言語モデル (LLM) を用いた単一エージェント型 API 推薦手法が提案されているが、多数の API 仕様を單一プロンプトに集約する必要があり、API 数の増加に伴ってプロンプト長が線形に増大する。その結果、コンテキスト長制限による API 仕様の入力切り詰め (truncation) や、長文コンテキストによる推論精度の低下といったスケーラビリティ問題が生じる。本論文では、このスケーリング上の制約を緩和するために、API 仕様を複数の LLM エージェントに分散保持させるマルチエージェント型サービス推薦手法を提案する。各コントラクタエージェントは単一 API の仕様のみを保持し、契約ネットプロトコル (CNP) に基づいて、マネージャから提示されるサービス構成要素に対し、自身の API が適合すると判断した場合に推薦の入札を行う。これにより、API 数の増加と單一プロンプト長の線形増大との直接的な依存関係を緩和する。さらに、コア機能分解、カテゴリマッピング、API マッチング、API 選択というサービス合成の推論過程を個別の推論タスクとして外在化 (externalization) し、これらの推論タスクの分担方式を設計変数として定義する。具体的には、マネージャ主導型、コントラクタ主導型、協調型の 3 方式を構築し、推論タスクの事前割当て方式が推薦性能に与える影響を比較する。ProgrammableWeb データセットを用いた評価実験により、トップダウン型のマネージャ主導方式が単一エージェント方式および他のマルチエージェント方式と比較して安定的に高い推薦精度を示すことを確認した。さらに、推論タスクの分担設計が推薦精度と推論コストのバランスを決定することを明らかにし、トーケン制約下における実用的な分散推論設計の指針を示す。

Index Terms—サービス合成、マルチエージェントシステム、大規模言語モデル、契約ネットプロトコル、API 推荐

I. はじめに

近年、サービスコンピューティングの発展により、多種多様な Web サービスを組み合わせて新たな付加価値を提供する複合サービスが盛んに構築されている。翻訳サービスと音声認識サービスを連携させたリアルタイム音声翻訳や、地理情報サービスと天気情報サービスを統合した地図表示サービスなど、個々の Web サービスでは実現できない機能を柔軟に実装できる。しかし、公開されている Web サービスのうち、複合サービスに実際に利用されているものは一部にとどまり、ユーザが膨大な候補の中から目的に適したサービスを選択することは依然として大きな負担となっている。

従来の複合サービス合成は、水平型サービス合成と垂直型サービス合成の二つのアプローチに大別される。前者は、所与のワークフローに対して各タスクを実行するのに適した Web サービスの組み合わせを選択する枠組みであり、ワー-

クフロー設計のコストが高いという課題を持つ。後者は、人工知能のプランニング技術を用いて、論理式で与えられたゴール状態に到達するサービス実行系列を生成する手法であるが、ユーザが時相論理や述語論理といった形式的表現で要求を記述しなければならず、エンドユーザにとって扱いやすいとは言い難い。この問題を軽減するため、自然言語で記述された要件定義からサービスの組み合わせを推定する手法が提案されてきた。

一方、近年急速に発展している LLM は、自然言語で記述された要求から計画を生成し、外部ツールや API を呼び出す能力を備えており、ユーザに形式的な記述を要求することなくサービス合成を行う枠組みとして期待されている。しかし、単一の汎用 LLM エージェントに広範なサービス領域を一括して扱わせる場合、個々のサービス仕様や実行環境に関する専門的な知識の理解が浅くなりやすい。また、多数の API 仕様や利用例を一つのプロンプトに詰め込む必要があるため、プロンプトの膨張とコンテキスト長制限に起因する性能劣化が生じるという問題がある。

本研究では、これらの課題を解決するため、複数の LLM エージェントからなるマルチエージェントシステムに基づくサービス合成手法を提案する。各エージェントを特定の API カテゴリや機能領域に特化させることで、プロンプト内で扱う知識を局所化しつつ、専門性の高いサービス推薦を実現することを目指す。さらに、タスク割当てメカニズムとして契約ネットプロトコル (CNP) [1] を導入し、マネージャエージェントとコントラクタエージェントがタスク分解と入札・応札を通じて協調的にサービス合成を行う枠組みを設計する。本論文では、マネージャ主導・コントラクタ主導・協調型という三種類のタスク分解プロトコルを比較し、サービス合成精度や単一エージェント方式との性能差を評価することで、LLM ベース・マルチエージェントによるサービス合成の有効性と課題を明らかにする。

本手法の実現にあたり、取り組むべき課題は以下の 2 点である。

- **タスク分解**: 自然言語で記述されたユーザ要求から、CNP に基づいて適切なタスク分解を行う。要求の粒度や曖昧さに応じて、マネージャ主導・コントラクタ主導・協調型といった異なるタスク分解戦略を切り替えつつも、一貫したサービス合成結果を得られる仕組みが必要となる。
- **コントラクタエージェントの組織化**: エージェントを特定の API カテゴリや機能領域に特化させることでプロンプトの膨張を抑えつつ、サービス仕様や実行環境

に関する専門的な知識を十分に反映させる必要がある。また、タスク分解プロトコルの違いがエージェント間の情報共有や推論過程に与える影響を明らかにすることも求められる。

以下、第 II 章でサービス合成の関連研究について説明する。第 III 章では、CNP に基づくマルチエージェント方式とタスク分解プロトコルの設計を示す。第 IV 章では、評価指標と実験設定を述べ、評価結果を示す。第 V 章で考察を行い、第 VI 章で結論と今後の展望を述べる。

II. 関連研究

A. 水平型サービス合成

水平型サービス合成とは、あらかじめ与えられたワークフロー（タスク列）に対し、各タスクを実行可能な Web サービス候補の中から、合成全体として望ましい組合せを選択する枠組みである。Zeng らは、同一の機能を提供するサービスが多数存在する状況を想定し、機能差ではなく QoS（非機能的特性）に基づいてサービスを選別する合成方式を提案した [2]。具体的には、価格・応答時間・可用性などの非機能情報から各サービスの QoS ベクトルを定義し、ユーザが与える制約や嗜好（重み付け）を用いて候補を評価する。

B. 垂直型サービス合成

垂直型サービス合成は、人工知能のプランニング技術を用いて、ユーザが指定したゴール状態に到達するための Web サービスの実行系列を生成する手法である。Carman らは、サービス合成を計画問題として捉え、セマンティック型マッチングと探索・実行を統合した逐次的な合成を組み合わせることで、不完全な情報の下でも柔軟に目標達成へ近づくアルゴリズムを提案した [3]。

C. LLM を用いたサービス合成

LLM を用いたサービス合成では、自然言語で記述されたユーザ要求から、適切な Web サービスを選択・組み合わせる。LLM の自然言語理解能力を活用することで、要求の意図や制約を推定し、タスク分割や API 選択を行える。

一方で、LLM を単純にプロンプト利用するだけでは、API の最新知識不足やハルシネーションにより推薦精度が低下することが指摘されている。Qin らは Llama-3.2-3B を基盤とする LLMAR を提案し [4]、指示学習と多段階ファインチューニングによって API ドメイン知識をモデル内部に注入する枠組みを示した。また、Matsumoto らは、推論過程を階層的に分解して段階的に進める Guided Reasoning Chains (GRC) を提案し [5]、要求の抽象度に応じてコア機能抽出、カテゴリ選択、候補列挙、最終推薦という流れで推論を行うよう LLM を誘導する。

III. 提案手法

A. 契約ネットプロトコル

契約ネットプロトコル (Contract Net Protocol; CNP) は、Reid G. Smith によって提案された分散問題解決のための高水準通信プロトコルであり [1]、タスクを外部に委託したい側のエージェントをマネージャ、タスクの実行を引き受けた側のエージェントをコントラクタと呼ぶ。両者の間でタスクの告知 (Call for Proposals)，入札 (Proposals)，契約締結 (Award)，結果報告という一連のやり取りを明示的なプロトコルとして規定している。

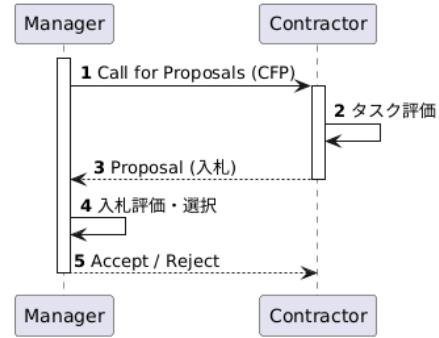


Fig. 1. CNP のシーケンス図

B. システム概要

本研究で提案するサービス合成システムは、LLM を用いたマルチエージェントシステムに基づき、自然言語で与えられた複合サービスの要件から、利用すべき Web API の組み合わせを推薦する。

本システムは、ユーザ要件を受理しタスクを編成するマネージャエージェント、各 API に専門化した複数のコントラクタエージェント、および各エージェントが参照する API 説明文から構成される。CNP を採用し、公告・入札・選択の流れのものと推論タスクを分担し、最終的な API 推薦結果を得る。

サービス合成の推論過程を、以下の 4 ステップからなる共通フローとして定義する。

- 1) **コア機能分解**: 要件文からユーザが最終的に達成したい目的を推定し、必要となる機能単位を抽出する。
- 2) **カテゴリマッピング**: 抽出した機能を実現し得る API カテゴリ（例：決済、地図、予約管理など）を推定する。
- 3) **API マッチング**: 推定カテゴリに属する API 説明文と要件に基づいて、適合する候補 API を抽出する。
- 4) **API 選択**: 抽出した候補 API を比較し、要件適合性の観点から採用 API を確定する。

これらは単一のエージェントが一括で実行するのではなく、CNP による公告・入札・選択の枠組みにおいて、マネージャとコントラクタの間で分担される推論タスクとして扱われる。

C. LLM に基づくエージェント

1) **コントラクタエージェント**: コントラクタエージェントは、個別 API に専門化された LLM エージェントであり、マネージャから公告されたタスク仕様を入力として受け取る。各コントラクタは、自身が参照可能な API 説明文のみを用いて推論を行い、当該タスクを担当可能か否かを判断する。担当可能と判断した場合、適合する API 候補とその判断根拠を入札としてマネージャに返す。担当不可能と判断した場合には、辞退メッセージを返す。

2) **マネージャエージェント**: マネージャエージェントは、ユーザ要求を受理し、サービス合成全体を統括する役割を担う LLM エージェントである。マネージャは、推論タスクを構成し、CNP に従ってコントラクタエージェントへ公告する。入札を受信した後、複数のコントラクタから提示された API 候補と根拠を比較・統合し、要求全体との整合性を確認した上で、最終的に採用する API 集合を確定させる。

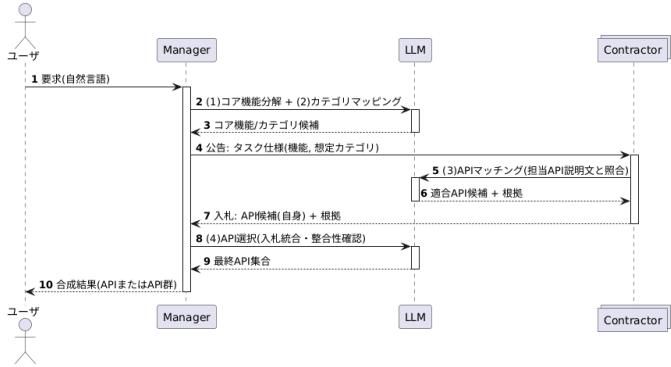


Fig. 2. マネージャ主導プロトコルのシーケンス図

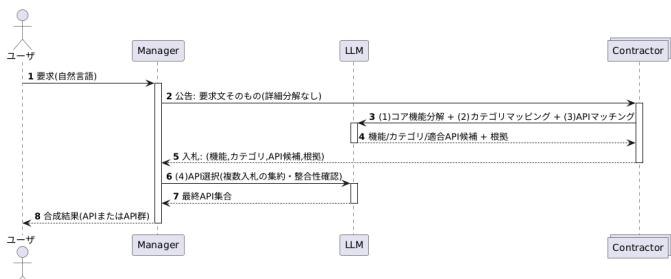


Fig. 3. コントラクタ主導プロトコルのシーケンス図

D. タスク分解プロトコル

本研究では、推論タスクの責任分担をCNP上の設計変数とみなし、マネージャ主導型、コントラクタ主導型、協調型の3方式を設計・比較する。

1) マネージャ主導型: マネージャ主導型では、サービス合成に必要な推論の大部分をマネージャエージェントが担当する。

- **マネージャの担当:** (1) コア機能分解, (2) カテゴリマッピング, (4) API選択
- **コントラクタの担当:** (3) APIマッチング

マネージャが要件文を解析し、必要なコア機能および対応するAPIカテゴリを事前に確定した上で、タスク仕様としてコントラクタに公告する。各コントラクタは、公告された機能及びカテゴリに基づきAPIマッチングを行い、適合するAPI候補と根拠を入札として返す。マネージャは入札を比較・統合し、最終的なAPI集合を確定させる。

2) コントラクタ主導型: コントラクタ主導型では、推論の大部分をコントラクタエージェントに委ねる。

- **マネージャの担当:** (4) API選択
- **コントラクタの担当:** (1) コア機能分解, (2) カテゴリマッピング, (3) APIマッチング

マネージャはユーザ要求文そのものをタスクとして公告し、詳細な分解やカテゴリ推定は行わない。各コントラクタは自身の担当API知識に基づいて要求文を解釈し、コア機能分解およびカテゴリマッピングを行った上で、適合するAPI候補と根拠を入札として提示する。マネージャは入札を集めし、最終的に採用するAPI集合を確定させる。

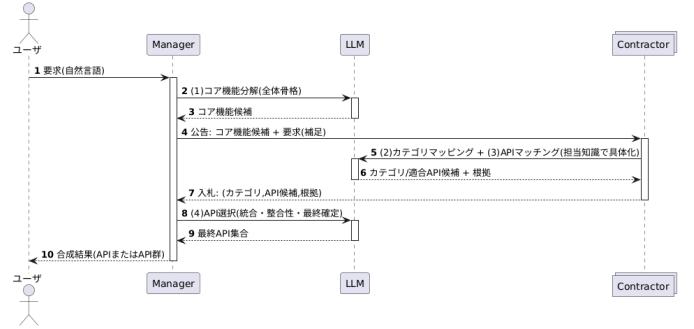


Fig. 4. 協調型プロトコルのシーケンス図

3) 協調型: 協調型では、マネージャとコントラクタが推論タスクを分担する。

- **マネージャの担当:** (1) コア機能分解
- **コントラクタの担当:** (2) カテゴリマッピング, (3) APIマッチング, (4) API選択

マネージャがユーザ要求からコア機能分解を行い、抽出した機能単位をタスクとして公告する。各コントラクタは、公告された機能に対してカテゴリ候補を推定し、担当API知識に基づいてAPIマッチングおよび選択を行い、採用すべきAPI候補と根拠を入札として返す。マネージャは各入札を統合し、最終的なAPI集合を確定させる。

IV. 評価

A. 実験設定

CNPベースのマルチエージェント方式について、3つの推論タスク割り当てプロトコルと単一エージェント方式を比較する。コントラクタエージェントの組織化については単一API単位の設計を採用し、カテゴリ単位で複数APIを担当させる設計は今後の課題とする。LLMにはgpt-oss20Bを用いた。

1) 実験データ: ProgrammableWebに掲載されている複合サービス(Mashup)データおよび原子サービス(Web API)データを使用した。複合サービスの構成サービスが原子サービスデータに含まれないレコードを除外するフィルタリングを行い、4,284件の複合サービスデータを得た。その中から複合サービス100件を抽出し、対応する909件のWeb APIを対象として評価データセットとした。原子サービスデータおよび複合サービスデータの構成概念をそれぞれ図5、図6に示す。

2) 評価指標: 適合率(Precision), 再現率(Recall), およびF1スコアをAPI名一致に基づく集合評価として算出する。最終的な推薦結果だけでなく、各プロトコルの以下の段階で評価を行う。

- **カテゴリ割当て時点:** 推定カテゴリに属する全APIを候補集合 P とし、正解集合 G と照合する。
- **入札時点:** コントラクタが入札として提示したAPI候補の集合を評価する。
- **最終推薦:** マネージャが全入札を集約・統合し確定したAPI集合を評価する。

加えて、各プロトコルの入出力トークン数を計測し、平均値で比較した。

```

{
  "サービス名": [
    [
      "メインカテゴリ",
      "サブカテゴリ",
      "サブカテゴリ"....,
    ],
    "サービス説明文",
    [
      "サービス提供者"
    ],
    [
      "サービス利用者",
      "サービス利用者"
    ],
  ],
}
  {
    "複合サービス": [
      [
        "カテゴリ",
        "カテゴリ"
      ],
      "サービス説明文",
      [
        "使用しているAPI"
      ],
      1
    ],
  }
}

```

Fig. 5. 原子サービスデータの構成概念

Fig. 6. 複合サービスデータの構成概念

TABLE I
各段階における推薦結果の比較

段階	方式	適合率	再現率	F1 値
カテゴリ割当て	マネージャ主導	0.111	0.607	0.177
	コントラクタ主導	0.145	0.357	0.184
	協調型	0.047	0.347	0.078
	単一エージェント	0.174	0.310	0.208
API マッチング	マネージャ主導	0.047	0.485	0.079
	コントラクタ主導	0.020	0.570	0.031
	協調型	0.034	0.260	0.058
	単一エージェント	0.106	0.121	0.105
API 選択	マネージャ主導	0.159	0.298	0.193
	コントラクタ主導	0.034	0.116	0.049
	協調型	0.083	0.223	0.114
	単一エージェント	0.137	0.141	0.130

B. 結果

各段階における推薦結果を表 I に、各方式のトーケン数を表 II に、最終推薦 API 数の平均を表 III に示す。

V. 考察

A. 単一エージェントとの比較

单一の汎用 LLM エージェントに多数の API 仕様を集約する方式では、プロンプト肥大化と文脈長制約により推論精度が低下しやすい。一方、マネージャ主導方式は、タスク分解とカテゴリ割当て、入札、統合という段階的なプロトコルにより、担当領域の文脈を局所化した提案が集約されるため、再現率と適合率の両面で改善が見られた。表 I に示す通り、単一エージェント方式は適合率 0.137、再現率 0.141 であるのに対し、マネージャ主導方式は適合率 0.159、再現率 0.298 を示した。カテゴリ割当て時点の F1 値は単一エージェント 0.208 に対しマネージャ主導 0.177 と単一の方が高いが、最終推薦ではマネージャ主導が 0.193 で単一の 0.130 を上回る。これは単一エージェントが多数の API を一括処理するため誤候補が混入しやすく、適合率が低下しやすいことによると考えられる。対照的にマネージャ主導は、入札と統合の段階で候補を段階的に精査できるため、最終的な精度を押し上げられる。

TABLE II
各方式のトーケン数（平均）

方式	入力	出力	総計
マネージャ主導	9,843	2,717	12,560
コントラクタ主導	1,952,788	999,576	2,952,364
協調型	522,411	71,828	594,240
単一エージェント	14,324	28,580	42,904

TABLE III
最終推薦 API 数（平均）

方式	平均 API 数
マネージャ主導	3.19
協調型	3.57
コントラクタ主導	4.61
単一エージェント	5.94

B. プロトコル間の比較

1) 推薦精度の観点: マネージャ主導型が最もバランスの取れた性能（適合率 0.159、再現率 0.298）を示した。トップダウンでカテゴリを絞り込むことで探索空間を適切に限定し、無関係な API からのノイズを効果的に排除できた。協調型（適合率 0.083、再現率 0.223）は柔軟性を持つものの、システム全体としての一貫性が保ちにくく、統合時の情報損失が影響した。コントラクタ主導型（適合率 0.034、再現率 0.116）は 3 方式中最も低い精度を示した。ただし、API マッチング時点の再現率は 0.570 と高く、局所知識が効いた可能性がある。一方で統合段階では無関連な候補が増えやすく、最終精度が低下した。以上の結果は、完全な自律分散よりも、ある程度の中央制御を残した階層的な意思決定の方がサービス合成には有効であることを示唆している。

2) 推論コストと効率性的観点: マネージャ主導型の総トーケン数（約 1.2 万）は、協調型（約 59 万）の約 50 分の 1、単一エージェント（約 5 万）と比較しても約 4 分の 1 に留まった。マネージャ主導型が初期段階でタスクと対象カテゴリを明確に定義し、関係するコントラクタのみに入札を求める選択的な通信を実現しているためである。コントラクタ主導型の総トーケン数は約 295 万に達し、マネージャ主導型の約 236 倍に相当する。

3) 責任分担と柔軟性のトレードオフ: マネージャ主導型はマネージャの負荷・責任が大きく、初期のカテゴリ判定を誤ると挽回が困難というリスクを持つ。しかし、実験結果からは、そのリスクよりも探索効率向上のメリットが上回ることが示された。協調型やコントラクタ主導型はコントラクタの自律性を重視し、マネージャの判断ミスを局所的な知識で補完できる可能性を持つが、統合の複雑さと通信コストが増大する。要件が極めて曖昧でカテゴリ定義自体が困難なケースにおいては、協調型のような柔軟なアプローチが再評価される可能性も残されている。

VI. おわりに

本論文では、LLM を用いたマルチエージェントシステムに基づく Web サービス合成手法を提案した。契約ネットプロトコルに基づくタスク分解メカニズムを導入し、マネージャ主導・コントラクタ主導・協調型という三種類のタスク分解プロトコルを設計・比較した。

本研究の貢献は以下である。

- **推論タスク割り当て**：推論プロセスをCNPに基づく責任分担モデルとして形式化し、3方式を比較した。マネージャ主導ではトップダウンでカテゴリを絞り込み、探索空間とノイズを抑えつつ高い適合率・再現率を達成した。協調型では柔軟性はあるものの、通信コストと統合時の情報損失が課題となった。
- **コントラクタエージェントの組織化**：API知識を单一LLMに集約するのではなく、コントラクタに分散保持させることで、コンテキスト長の制約とハルシネーションを軽減できることを示した。1API単位で専門化した構成でも、大規模なAPI群に対するスケーラビリティと更新耐性を確保できることを確認した。

今後の展望として、コントラクタに複数APIを担当させた場合の精度・通信量・推論負荷を評価し最適な粒度を求めるここと、また異なる基盤モデルで同一プロトコルを検証しモデル依存性と頑健性を明らかにすることに取り組む。

謝辞

本研究を行うにあたり、熱心なご指導、ご助言を賜りました村上陽平教授並びに松本賢司さんに深く感謝を申し上げます。

REFERENCES

- [1] R. G. Smith, “The contract net protocol: High-level communication and control in a distributed problem solver,” *IEEE Trans. Comput.*, vol. C-29, no. 12, Dec. 1980.
- [2] L. Zeng, B. Benatallah, A. H. H. Ngu, M. Dumas, J. Kalagnanam, and H. Chang, “QoS-aware middleware for Web services composition,” *IEEE Trans. Softw. Eng.*, vol. 30, no. 5, pp. 311–327, 2004.
- [3] M. A. Carman, L. Serafini, and P. Traverso, “Web service composition as planning,” in *Proc. ICAPS 2003 Workshop on Planning for Web Services*, 2003, pp. 1636–1642.
- [4] S. Qin, Y. Zhao, H. Wu, L. Zhang, and Q. He, “Harnessing the power of large language model for effective Web API recommendation,” *IEEE Trans. Ind. Inform.*, vol. 21, no. 7, Jul. 2025.
- [5] K. Matsumoto and Y. Murakami, “Guided reasoning chains for API recommendation,” in *Proc. ICSOC Workshops*, 2025.