

卒業論文

マルチエージェントシステムによる
Web サービス合成

指導教官 村上 陽平 教授

立命館大学情報理工学部
先端社会デザインコース

飛澤 佑季

2025年1月9日

マルチエージェントシステムによる Web サービス合成

飛澤 佑季

内容梗概

近年、Web サービスは生活や産業のあらゆる場面で不可欠な存在となり、複数の Web サービスを組み合わせて利用するサービス合成の重要性が高まっている。しかし従来手法では、ユーザーの目的や要求を時相論理・述語論理といった形式的表現で記述する必要があり、専門知識を持たない利用者にとって大きな障壁となっていた。

本研究では、大規模言語モデル (LLM) が自然言語で要求を記述できる点に着目し、柔軟かつ直感的に利用可能なサービス合成の実現を目指す。一方で単一エージェント方式では、多数の API 仕様や専門知識を 1 つのプロンプトに集約する必要があり、プロンプト肥大化や入力制限により推論精度が低下しやすい。

そこで、複数の LLM エージェントを協調させるマルチエージェントシステムを導入し、エージェントに API 仕様を分散保持させることでプロンプト制約を緩和する。具体的には、契約ネットプロトコル (Contract Net Protocol: CNP) に基づき、マネージャが要求を公告し、入札エージェントが担当可否と API 案を入札、マネージャが統合して合成計画 (Plan) を確定する。

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に対し、CNP に基づいてタスク分解を行う際に、要求の粒度や曖昧さに応じて分解方式をどのように選択すべきか、またその分解をマネージャ等のどの役割のエージェントが担うことが適切かを検証する必要がある。

入札エージェントの組織化 CNP では、入札エージェント数の増加に伴い公告・入札の通信量が増大し、スケーラビリティが低下しやすいという課題がある。このデメリットを緩和するため、入札エージェントに対してどの API を割り当てるべきか、また入札エージェント群のカテゴリ別階層化・集約を検討し、組織構造の違いが入札の精度や合成結果、ならびに通信負荷に与える影響を検証する必要がある。

1 つ目の課題に対しては、マネージャ主導・コントラクタ主導・協調型の 3 つのタスク分解プロトコルを比較する。マネージャ主導では、マネージャがタスク分解とカテゴリ分解を行ってタスク仕様を公告し、入札エージェントは担当

可否と API 案を入札する。コントラクタ主導では、入札エージェントが自身の担当 API 仕様に基づいてタスク分解とカテゴリ分解を推定して入札する。協調型では、マネージャがタスク分解を提示したうえで、入札エージェントが分解されたタスクに対してカテゴリ分解を行って入札する。本研究では、要求の粒度や曖昧さといった要求タイプに応じて、これら 3 つのプロトコルのうちどれが適するかを明らかにし、分担の違いが合成計画の整合性および計画生成精度に与える影響を分析する。

2 つ目の課題に対しては、入札エージェントの組織化により解決を図る。具体的には、入札エージェントに割り当てる担当範囲として、単一 API を担当させる設計に加え、API カテゴリ単位で複数 API をまとめて担当させる設計も比較対象とする。各エージェントは、割り当てられた API（またはカテゴリに属する複数 API）の仕様のみを参照して推論・入札を行うようにし、推薦精度および通信負荷に与える影響を検証する。

本研究では、提案手法の有効性を検証するため、ProgrammableWeb の複合サービスデータを用いた評価を行った。具体的には、複合サービスデータ 400 件をテストデータとし、対応する 909 件の Web API を対象として、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方式を適用してサービス合成を実行し、各方式の計画生成精度を比較した。生成されたサービス計画と正解計画を照合し、適合率、再現率、F1 スコアを算出することでタスク分解方式の性能を評価した。また、入札エージェントの組織化（単一 API 単位の割当と、カテゴリ単位で複数 API を割り当てる設計）についても、各エージェントの入札行動と推薦精度を比較し、役割分担の違いがサービス合成結果に与える影響を分析した。また本研究の貢献は以下の通りである。

タスク分解 マネージャ主導・コントラクタ主導・協調型の三つのタスク分解プロトコルを比較し、ユーザ要求の粒度に応じた適切な方式を分析した。適合率・再現率・F1 値で評価した結果、本手法は単一方式を上回る性能を示した。

入札エージェントの組織化 エージェント数と担当 API 数を変化させた実験を実施し、専門性の深さと分担効率の関係を評価した。その結果、API カテゴリを限定した専門エージェントを用いることで、単一エージェント方式と比較してサービス推論の正確性が向上し、大規模サービス集合に対してもスケラブルに動作することが明らかとなった

Web Service Composition via a Multi-Agent System

Yuki Tobisawa

Abstract

Web services have become indispensable in daily life and industry, and the importance of composing multiple Web services to realize value-added applications is increasing. However, conventional approaches often require users to specify goals and constraints in formal languages such as temporal or predicate logic, which poses a barrier to non-expert users.

This study aims to enable flexible and intuitive service composition by leveraging large language models (LLMs), which allow users to describe requirements in natural language. A single-agent approach tends to aggregate many API specifications and domain knowledge into one prompt, which can lead to prompt bloating and degraded inference quality due to context-length limitations.

To mitigate this issue, we introduce a multi-agent LLM system in which API specifications are distributed across agents. Based on the Contract Net Protocol (CNP), a manager agent announces a user request, contractor agents bid with feasibility and candidate APIs, and the manager integrates bids to finalize a composition plan (Plan).

We address two challenges: (1) task decomposition from natural-language requirements under CNP, including how to select a decomposition protocol and which agent role should perform it; and (2) organizing contractor agents to reduce communication overhead while maintaining recommendation accuracy. We compare three task decomposition protocols—manager-led, contractor-led, and cooperative—and also compare assigning agents per API versus per API category.

For evaluation, we use 400 mashup services from ProgrammableWeb as test data and 909 corresponding Web APIs. We execute service composition under each protocol and measure plan-generation quality by comparing generated plans to ground-truth plans using precision, recall, and F1 score. We further analyze how different agent organizations affect bidding behavior, recommendation quality, and communication cost.

マルチエージェントシステムによる Web サービス合成

目次

第1章	はじめに	1
第2章	サービス合成	3
2.1	サービス合成手法	3
2.1.1	水平型サービス合成	3
2.1.2	垂直型サービス合成	3
2.2	LLM を用いたサービス合成	4
第3章	契約ネットプロトコルに基づくサービス合成	5
3.1	契約ネットプロトコル	5
3.2	システム概要	6
3.3	LLM に基づくエージェント	7
3.3.1	コントラクタエージェント	8
3.3.2	マネージャエージェント	8
3.4	プロトコル	8
3.4.1	マネージャ主導タスク分解	8
3.4.2	コントラクタ主導型タスク分解	8
3.4.3	協調型タスク分解	8
第4章	評価	9
4.1	実験設定	9
4.1.1	実験データ	9
4.1.2	評価指標	9
4.2	実験結果	9
4.2.1	マネージャ主導タスク分解	10
4.2.2	コントラクタ主導タスク分解	10
4.2.3	協調型タスク分解	10
第5章	考察	10
5.1	単一エージェントとの比較	10
5.2	コントラクタエージェント数	10

5.3 今後の課題	10
第 6 章 おわりに	11

第1章 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る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や、WSDL やサービス説明文、サービスネットワークをテキストマイニングやグラフ埋め込みによって特徴ベクトル化し、機能ごとにクラスタリング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大規模言語モデル (Large Language Model; 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、エージェント数や担当 API 数が与える影響を評価することで、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。

そこで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、契約ネットプロトコル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。

LLM エージェントの役割分担 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。

第2章 サービス合成

本章では、複合サービスの実現に向けたサービス合成手法について説明する。複合サービスとは、インターネット上に存在する複数の Web サービスを組み合わせ、新たなサービスのことである。近年では、さまざまな Web サービスが API を通じて連携可能となっており、ユーザ要求に応じた柔軟なシステム開発が可能となっている。

複合サービスの代表例として Expedia を挙げる。Expedia は航空券、宿泊、移動手段など複数の API を統合することで、旅行計画・予約を一括提供している。航空券の検索では航空会社情報を提供する API、予約確定時には決済 API、旅の計画閲覧には地図 API を利用し、単独では実現不可能な利便性を提供している。このような複合サービスを開発するためには、ユーザ要求に応じて必要な機能を提供できる Web サービスを適切に発見する必要がある。しかし膨大な数の Web サービスの中から目的に合致したものを探索することは容易ではない。本章では、サービス合成手法について説明する。

2.1 サービス合成手法

従来のサービス合成手法では大きく、水平サービス型と垂直サービス型の 2 つに分類できる。以下でそれぞれについて説明する。

2.1.1 水平型サービス合成

水平型サービス合成とは、同一の機能領域に属する複数のサービスから適切なものを選択し、代替性や冗長性を高めるために組み合わせる手法である。例えば、Expedia で航空券予約 API を選択する際、複数の航空会社 API の中から価格・便数・信頼性などを基準に最適なサービスを選択することが挙げられる。このような選択を自動化することで、サービス品質の向上や障害耐性の確保が期待できる。

2.1.2 垂直型サービス合成

垂直型サービス合成とは、異なる機能領域を階層的に結合し、ワークフロー全体を構築する手法である。例えば、旅程計画 → 予約 → 決済 → 地図表示 → 渡航情報通知といったように、ユーザ行動に沿って複数 API が連携することで完全な体験を提供する。Expedia の例では、航空券検索 API、宿泊施設予約 API、決済 API、位置情報 API が連携して、旅行計画という 1 つのタスクを段階的に

実現している．このような垂直型の連携では，サービス間の入力/出力データの整合性が重要となる．

2.2 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・組み合わせる手法である．従来機能要件を論理ベースで記述し，形式的なフロー設計が必要であったが，LLM の自然言語理解能力を活用することで，機能要件を推定し，それに基づいてタスク分割をすることができる．そして API 使用に関しても，LLM が API ドキュメントを解析し，適切な呼び出し方法を生成することで，ユーザは専門知識なしにサービス合成を実現できる．

第3章 契約ネットプロトコルに基づくサービス合成

本章では、LLMを用いた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について説明する。マルチエージェントを用いる中で本研究では契約ネットプロトコル (Contract Net Protocol; CNP) を採用し、エージェント間の協調メカニズムとして利用する。

3.1 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、分散環境におけるタスク割当てと協調計算を、交渉に基づくメッセージ交換によって実現する枠組みである。CNP では、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 (manager)、タスクの実行を引き受ける側のエージェントをコントラクタ (contractor) と呼び、両者の間でタスクの告知、入札、契約締結、結果報告といった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。

典型的な CNP の相互作用は、次のようなステップで構成される。

1. タスク告知 (task announcement / call for proposals)

マネージャは、自身が保持するタスクの内容 (目的, 前提条件, 必要資源, 締切, 評価基準など) を記述した告知メッセージをブロードキャストし、コントラクタ候補に対して提案を呼びかける。

2. 入札 (bidding / proposal submission)

告知を受け取った各コントラクタは、タスクを遂行可能かどうか、現在の負荷や利用可能資源を考慮して評価する。実行可能であれば、予想コスト、所要時間、達成可能な品質などを含む入札 (プロポーザル) をマネージャに送信し、実行困難と判断した場合は辞退メッセージを返す。

3. 契約締結・割当て (award / contract)

マネージャは、収集した複数の入札を比較し、評価基準に基づいて最も望ましいコントラクタを選定する。選定されたコントラクタには契約成立を示すアワードメッセージを送り、その他のコントラクタには不採択を通知する。これにより、当該タスクに関する一時的なマネージャ-コントラクタ関係が形成される。

4. 実行と結果報告 (task execution / status reporting)

コントラクタは割り当てられたタスクを実行し、中間状態や完了結果をステータス報告メッセージとしてマネージャに送信する。マネージャはこれらの報告に基づいてタスクの進捗を把握し、必要であれば再割当てなどの制御を行う。

このように、CNP は単なるタスク割当てアルゴリズムというよりも、タスク告知から結果報告に至るまでのエージェント間通信の流れとメッセージ種別を明確に定義したコミュニケーション・プロトコルとして位置づけられる。各エージェントは、あるタスクに関してはマネージャとして振る舞い、別のタスクに関してはコントラクタとして振る舞うことができるため、システム全体としては固定的な中央管理者を持たない柔軟な階層構造を動的に形成できるという特徴を持つ。また、タスク告知の範囲や入札の条件を調整することで、探索空間の絞り込みや通信量の制御が行える点も利点とされている。本研究では、このCNP をエージェント間の協調メカニズムとして採用し、サービス合成におけるタスク（機能分割や API 選択など）の割当てに応用する。

3.2 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アーキテクチャに基づき、自然言語で与えられたマッシュアップサービスの要件から、利用すべき Web サービス API の組み合わせを推薦することを目的として設計されている。システムは、ユーザからの要件文を受け取るマネージャエージェント、各 API カテゴリや個別 API に専門家した複数のコントラクタエージェント、およびそれらが参照する API の説明文から構成される。エージェント間の協調アルゴリズムとして、CNP を採用し、タスクの公募、入札、選択という枠組みの中でサービス推薦を行う。本システムでは、サービス合成の過程を次の4つのステップからなる共通フローとして定義する。

1. 与えられた要件に基づいて、マッシュアップサービスのコア機能を推論する。
2. コア機能と利用可能な API に基づいて、利用可能なカテゴリから関連する API カテゴリを特定し、提案する。
3. 説明と要件に基づいて、各カテゴリの API をマッチングする。
4. 最適な API を選択し、理由とともに推奨する。

ここでコア機能推論（ステップ 1）は、要件文からユーザが最終的に達成したい目的や必要となる機能単位を抽出する過程であり、カテゴリ特定（ステップ

2) は、そのコア機能を実現し得る API カテゴリ（例：決済、地図、予約管理など）を列挙する過程である。ステップ 3 では、個々の API 説明文を要件と照合し、適合性を評価する。最後にステップ 4 で、複数の候補 APIの中から最適な組み合わせを選択し、なぜその API が要件に適合すると判断したのかを自然言語で説明する。

契約ネットプロトコルの観点から見ると、上記の 4 ステップは単一のエージェントが一括して実行されるのではなく、マネージャエージェントとコントラクタエージェントの間で分担されるタスクとして扱われる。マネージャエージェントは、要件文の受理とタスクの公募、入札結果の集約と最終的な候補選択を担う役割を持ち、コントラクタエージェントは、自身が担当する API カテゴリや API 群に関する知識に基づいて、コア機能の解釈、カテゴリ候補の提案、具体的な API 候補の提示といった推論を行い、入札としてマネージャに提示する。

本研究では、この共通の 4 ステップフローは固定したまま、どのステップをどのエージェントが担当するかを変更することで、マネージャ主導のトップダウン方式、中間的な役割分担方式、コントラクタ主導のボトムアップ方式という 3 つのプロトコルを実装する。トップダウン方式では、マネージャが主に上位の機能分割を行い、コントラクタがカテゴリ予測や API 推薦を担う構成とし、ボトムアップ方式では、コントラクタエージェントが要件文からの機能分割を含めた全ステップを引き受ける構成とする。中間方式では、機能分割とカテゴリ選択の一部をマネージャが先行して行い、残りのステップをコントラクタが担当する。

このように、本システムは「4 ステップのサービス合成フロー」と「CNP に基づくタスク割当て」を組み合わせた枠組みとして設計されている。各ステップの内部で LLM がどのようなプロンプトと知識表現に基づいて推論を行うかは 3.3 節で、また三つの方式におけるステップ担当の違いと、それが推薦精度や実行時間、トークン消費量に与える影響については、3.4 節および評価章において詳述する。

3.3 LLM に基づくエージェント

本研究では、ユーザ要求（自然言語）と API 説明文を入力として、サービス合成のための合成計画（Plan）を生成する推論主体として LLM エージェントを用いる。単一エージェント方式では、多数の API 仕様を 1 つのプロンプト

に集約する必要がある、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。提案手法では、入札エージェントごとに参照できる API 仕様を限定（単一 API または API カテゴリ）し、担当領域の知識を局所化することで、プロンプト制約の緩和と専門性の確保を図る。

また、CNP の公告・入札・選択という枠組みを用いることで、推論の責務（タスク分解、カテゴリ特定、API マッチング、統合）を役割ごとに分担し、要求の粒度や曖昧さに応じた推論手順を実現する。

3.3.1 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、公告（タスク仕様）を受け取り、自身の担当 API 仕様に基づいてタスク遂行の可否を判断し、実行可能な場合は候補 API とその根拠を入札として返す。プロトコルによって公告の抽象度が異なるため、必要に応じて要求文からタスク分解やカテゴリ推定を補助的に行い、入札に含める。

3.3.2 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、CNP に従って公告内容を構成してコントラクタへ配信する。入札を収集した後、提案の整合性（入出力の接続可能性、要求充足の妥当性など）を確認し、最終的な合成計画（Plan）として統合・確定する。

3.4 プロトコル

3.4.1 マネージャ主導タスク分解

マネージャ主導では、マネージャがユーザ要求からタスク分解およびカテゴリ分解を行い、各サブタスクの仕様を公告する。コントラクタは公告された仕様に対して、担当 API の範囲内で実行可能性と候補 API を入札する。

3.4.2 コントラクタ主導型タスク分解

コントラクタ主導では、マネージャは比較的抽象度の高い要求を公告し、各コントラクタが担当 API の知識に基づいてタスク分解・カテゴリ推定を含む提案を行う。マネージャは複数の提案を統合し、整合的な Plan を確定する。

3.4.3 協調型タスク分解

協調型では、マネージャがタスク分解を提示したうえで公告し、コントラクタが各サブタスクに対してカテゴリ分解と候補 API を提案する。トップダウン（タスク分解）とボトムアップ（カテゴリ・API 推定）を組み合わせることで、

要求の曖昧さに対する頑健性と整合性の両立を狙う。

第4章 評価

4.1 実験設定

本研究では、提案する CNP ベースのマルチエージェント方式について、タスク分解プロトコル（マネージャ主導・コントラクタ主導・協調型）と入札エージェントの組織化（単一 API 単位／カテゴリ単位）を比較する。いずれの設定においても、ユーザ要求（マッシュアップ要件）を入力として合成計画（Plan）を生成し、正解計画との照合によって計画生成精度を評価する。また、公告・入札に伴う通信負荷についても、メッセージ数等の指標により比較する。

4.1.1 実験データ

評価データには ProgrammableWeb の複合サービス（マッシュアップ）データを用いる。複合サービスデータ 400 件をテストデータとし、対応する 909 件の Web API を対象として、各プロトコルおよび組織化方式でサービス合成を実行する。

4.1.2 評価指標

生成されたサービス計画と正解計画を照合し、適合率（Precision）、再現率（Recall）、F1 スコアを算出することで計画生成精度を評価する。加えて、入札エージェントの組織化に関しては、各エージェントの入札行動と通信負荷の変化も分析する。

4.2 実験結果

各方式で生成された Plan を正解計画と照合し、計画生成精度を比較した。その結果、要求タイプ（粒度・曖昧さ）によって有利なタスク分解方式が異なる傾向が確認され、単一方式に固定するよりも、要求タイプに応じて方式を切り替える運用が有効であることが示された。また、入札エージェントの組織化により、専門性と分担効率のトレードオフが観測され、API カテゴリを限定した専門エージェントを用いることで、単一エージェント方式と比較して推薦の正確性が向上し、大規模サービス集合に対してもスケーラブルに動作することが確認された。

4.2.1 マネージャ主導タスク分解

マネージャ主導方式の結果を示す。タスク分解をマネージャが一貫して行うことで、合成計画の整合性を保ちやすい一方、要求が曖昧な場合には、コントラクタ側の知識を十分に引き出せない場合がある。

4.2.2 コントラクタ主導タスク分解

コントラクタ主導方式の結果を示す。コントラクタが担当領域の API 仕様に基づいて分解を提案するため、局所的な知識を反映した提案が得られる一方で、提案間の統合にはマネージャ側での整合性確認が重要となる。

4.2.3 協調型タスク分解

協調型方式の結果を示す。マネージャによるタスク分解とコントラクタによるカテゴリ分解を組み合わせることで、整合性と柔軟性の両立を図る。

第5章 考察

5.1 単一エージェントとの比較

単一の汎用 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。提案手法では API 仕様を分散保持させることで、担当領域の文脈を局所化し、推薦の正確性を維持しながらスケールさせる設計の有効性を確認した。

5.2 コントラクタエージェント数

エージェント数と担当 API 数を変化させた場合、専門性の深さ（担当範囲の狭さ）と分担効率（エージェント数の少なさ）にトレードオフが生じる。カテゴリ単位の担当は、単一 API 単位と比較して通信量を抑えつつ、一定の専門性を維持できる設計として有効である。

5.3 今後の課題

今後の課題として、(1) 公告範囲の動的制御や入札の事前フィルタリング等による通信効率化、(2) 要求タイプ推定に基づくプロトコル自動選択、(3) Plan の入出力整合性検査の強化（型・スキーマの導入）などが挙げられる。

第6章 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。また、入札エージェントの組織化として、単一 API 単位の割当てに加え、カテゴリ単位で複数 API をまとめて担当させる設計を比較した。

ProgrammableWeb の複合サービスデータを用いた評価により、要求タイプ（粒度・曖昧さ）に応じてタスク分解方式を選択することが、単一方式に固定する場合よりも有効であることを確認した。さらに、API 仕様を分散保持する専門エージェントを用いることで、単一エージェント方式と比較して推薦の正確性を高めつつ、大規模サービス集合に対してもスケーラブルに動作することを示した。

今後は、通信効率化、プロトコル自動選択、Plan の整合性検査の強化などに取り組むことで、より実用的で柔軟なサービス合成システムの実現を目指す。