

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、ユーザーが自身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）に自然言語で要求を記述できる利点に着目し、これをエージェントとして活用することで従来のプランニング手法と統合し、より柔軟かつ直感的に利用可能なサービス合成を実現する。しかし単一エージェント方式では、対象サービス領域が広い場合に知識の希薄化や技術仕様の理解不足が生じ、さらに専門知識をプロンプトへ付与する場合にはプロンプトの肥大化と文字数制限が新たな問題として浮上する。そこで本研究では、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、各エージェントを特定領域に特化させることでプロンプト制約を緩和し、より高精度なサービス推薦・合成を可能にするアプローチを提案する。

加えて、将来的にはユーザーの嗜好や利用履歴を踏まえた能動的なサービス推薦システムへの発展を見据え、エージェント間の協調プロトコルとして契約ネットプロトコル（Contract Net Protocol）を応用する。各 Web サービスに対して 1 つの LLM エージェントを割り当て、エージェントはサービスドキュメント、API 仕様書利用規約などの自然言語記述を解析し、機能、入出力パラメータ、制約、実行環境等を構造化した知識ベースを構築する。ユーザーから自然言語で要求が与えられると、各エージェントは要求内容との整合性や制約充足度を自己評価し、十分な適合性がある場合には入札として推薦文を生成する。入札は、サービス仕様・要求内容の関連度、対応可能な機能、期待される成果物の品質などを自然言語で説明したものである。入札が揃った後は、評価専用エージェントが各推薦文を比較・評価し、関連度スコアや専門領域における信頼性指標などを総合的に判断して最適なエージェントを落札者として選定する。最終的に選定されたエージェントによるサービス合成結果を正解データセットと照合し、適合率・再現率・F1 値などを用いて定量的に性能評価を行う。本研究

が提案する方式により、各エージェントはサービス仕様に基づいた自律的判断と説明能力を備え、入札・評価・選定というプロトコルに基づく協調によって、効率的かつ高精度なサービス合成の実現が期待される。

How to Write Your B.E./M.E. Thesis

Yuki Tobisawa

Abstract

This guide gives instructions for writing your B.E. or M.E. theses following the standard of the Department of Information Science. The standard includes the structure and format which you must obey on writing your theses.

This guide also explains how to use a L^AT_EX style file for theses, named `kuisthesis`, with which you can easily produce well-formatted results. Since this guide itself is produced with the style file, it will help you to refer its source file `guide.tex` as an example.

Note for graduate students: This document is written for students of old graduate school of information science, not for graduate school of informatics. Writers of master thesis belonging to graduate school of informatics must obey rules given by each department.

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	8
4.1	実験設定	8
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		10

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してきた。

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て、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採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ます。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、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、各カテゴリ内で個々の 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三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

結論は論文のまとめとして、研究成果の要点を簡潔に記述すべきである。これは当然、本論に置ける本質的な部分を圧縮したものとなるが、内容梗概とは異なり、論文の締めくくりにふさわしい格調のうちに完結するように努めなけ

ればならない。

また研究途上に派生した副次的な問題や、将来に残された研究課題があれば、それらについても触れることができ望ましい。

なお結論のあとに、研究上の指導、助言、援助を受けた人々に対して、謝辞を書くのが研究発表者の礼儀であるから、特別研究報告書や修士論文においても、この慣習に従うことが望ましい。

3.3.1 コントラクタエージェント

3.3.2 マネージャエージェント

3.4 プロトコル

3.4.1 マネージャ主導タスク分解

3.4.2 コントラクタ主導型タスク分解

3.4.3 協調型タスク分解

第4章 評価

4.1 実験設定

報告書および論文は、A4の用紙の片面に印字する。図表を含めた本文の長さはつきの基準による。ただし枚数の増減は1割以内とする。

- 特別研究報告書：25枚
- 修士論文 : 50枚（ただし英文の場合には60枚になってもよい）

なお図表の分量は全体の40%程度を限度とし、これを超過する場合は適宜付録にまわすこと。

日本文／英文にかかわらず、ワードプロセッサあるいは(LA)TeXのような製版ツールを用いて清書し、手書きは許されない。

論文の各ページの左端3cmと右端1cmは必ず空白とすること。

日本文の場合は12pt（あるいは相当の大きさ）のフォントを用い、1行当たり35文字、1ページあたり32行とする。ただしワードプロセッサを用いる場合、この基準が守れない場合は1ページ当たりの字数が同程度となるようにして、1行当たりの字数や、1ページあたりの行数を調整してもよい。

英文の場合は12pt（あるいは相当の大きさ）のフォントを用い、1行の幅を14.2cm、1ページあたり32行とする。

日本文／英文にかかわらず、章・節の見出しあは2行分とし、それ以下の小節の見出しあは1行分とする。また箇条書の前後や項目間に余分な空白は挿入しな

いこと。

4.1.1 実験データ

報告書および論文には、内容梗概のまえに「とびら」をつけ、全体を所定のファイルにじて提出する。

表紙には、報告書／修士論文の別、題目、指導教官名、所属学科／専攻名、氏名、提出年月を記入する。

とびらは本文と同じ用紙を用い、表紙と同様の事項を記載する。

内容梗概（日本文、英文とも）の始めには、題目および氏名を書く。また目次の始めにも題目を記載する。

4.1.2 評価指標

論文および報告書は、予告された期限までに完成し、教室事務室に提出すること。提出期日は厳守しなければならない。

4.2 実験結果

論文の執筆にとりかかる前に、構成や長さについて十分検討し、各研究室で行なわれる発表会などで他人の意見をよく聞いて記述する内容を吟味する。また最終的な形に仕上げるまでに、自分自身で何度も校正を重ね、論旨の飛躍や矛盾のないように注意するとともに、よく文章を練り、誤字や誤記を除くように心がけなければならない。さらに、できれば先輩に目を通してもらって、思い違いや不注意による誤りを訂正し、できるだけ完璧なものにするように努めるべきである。

- 4.2.1 マネージャ主導タスク分解
- 4.2.2 コントラクタ主導タスク分解
- 4.2.3 協調型タスク分解

第5章 考察

- 5.1 単一エージェントとの比較
- 5.2 コントラクタエージェント数
- 5.3 今後の課題

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・協調型の三種類のプロトコルを設計した。評価実験では、各プロトコルの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 数が与える影響を分析した。実験結果から、マルチエージェント方式が単一エージェント方式に比べて高精度なサービス合成を実現できることが明らかとなった。特に、コントラクタ主導型タスク分解プロトコルが最も優れた性能を示し、エージェント数の増加がサービス合成精度の向上に寄与することが確認された。今後の課題として、エージェント間の通信効率化や、動的なタスク割当て戦略の導入が挙げられる。これらの課題に取り組むことで、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。