

# ルールベースとLLMのハイブリッドアプローチによる 対話管理機構の提案

## A Proposal of Dialogue Management Framework Based on a Hybrid Approach of Rule-Based and LLM

矢吹 優翔<sup>1\*</sup> 黄 宏軒<sup>1</sup>  
Hiroto Yabuki<sup>1</sup> Hung-Hsuan Huang<sup>1</sup>

<sup>1</sup> 福知山公立大学 情報学部

<sup>1</sup> Faculty of Informatics, The University of Fukuchiyama

**Abstract:** This paper proposes a hybrid dialogue management framework that combines rule-based and Large Language Model (LLM) approaches. The dialogue flow is controlled by a rule-based state machine, while user input understanding and response generation are performed by LLMs. By constraining LLM outputs to predefined enumeration types, the non-deterministic outputs of LLMs are converted into deterministic formats, enabling reliable integration with rule-based condition evaluation. Dialogue scenarios are defined in XML scripts, allowing dialogue flow design without programming expertise and significantly reducing development costs. An application example of a tourism information system demonstrates the effectiveness of the proposed approach.

### 1 はじめに

対話システムには、ルールベースによるタスク指向型と、LLM (Large Language Model) による非タスク指向型の2つのアプローチが存在する [1]。ルールベースは出力の確実性と明示的な状態管理に優れるが、多様な言い回しへの対応や想定外の入力処理が困難である。一方、LLMは柔軟な言語理解と自然な応答生成が可能だが、出力が非決定的である。ここで非決定的とは、同じ入力に対して毎回異なる応答が生成される可能性があることを意味し、これは対話フローの制御を困難にする要因となる。また、LLMにはハルシネーション (存在しない情報の生成) のリスクもある。

タスク指向型対話システムは、レストラン予約や情報検索など、明確なゴールを持つタスクの遂行を目的とする。フレームベースの状態管理により、必要な情報 (スロット) を対話を通じて収集し、タスクを完了させる。このアプローチは、予約確認のように出力の正確性が求められる場面、複数ターンにわたって情報を漏れなく収集する必要がある場面、同じ入力に対して常に同じ応答が求められる場面でも有利である。しかし、「安い」「リーズナブル」「お手頃」など同じ意図を表す多様な言い回しを全て事前定義する必要があり、実

用レベルでは膨大なパターン定義が必要となる。また、想定外の入力 (雑談など) に対応できないという問題もある。

非タスク指向型対話システムは、雑談や相談など、明確なゴールを持たない自由な対話を目的とする。LLMを用いることで、多様な言い回しへの対応、想定外のトピックへの柔軟な応答、自然で人間らしい会話が実現できる。しかし、出力が非決定的であるため対話フローの制御が困難であり、ハルシネーションのリスクや、複数ターンにわたる状態管理の困難さという課題がある。図1に両手法の処理フローを示す。

実世界のアプリケーションでは、正確性が求められる処理と柔軟性が求められる処理が混在する。例えば、

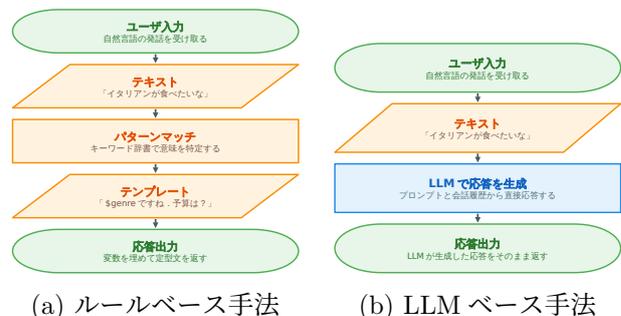


図1: 従来手法の処理フロー

\*連絡先: 福知山公立大学 情報学部  
〒620-0886 京都府福知山市字堀 3370  
E-mail: hhhuang@acm.org

観光案内システムでは正確な情報提供（ルールベース向き）と旅の相談への柔軟な対応（LLM 向き）の両方が必要である。カスタマーサポートでは定型的な問い合わせ処理（ルールベース向き）と顧客感情への共感（LLM 向き）を両立させる必要がある。どちらか一方のアプローチだけでは対応できないため、両者を組み合わせたハイブリッドアプローチが必要となる。本研究では、このような要求に応えるため、ルールベースと LLM を組み合わせたハイブリッド対話管理機構を提案する。対話フローの制御はルールベースで行い、ユーザ入力の理解と応答生成には LLM を活用する。入力理解においては、LLM の出力を事前定義された列挙型に制約することで、非決定的な LLM の出力を決定的な形式に変換し、ルールベースの条件判定と確実に接続できるようにする。これにより、LLM の柔軟な言語理解能力を活かしながら、対話フロー全体の制御性を確保する。

提案手法の特徴は以下の通りである。第一に、対話シナリオを XML スクリプトで定義するため、プログラミングの専門知識がなくても対話フローを設計できる。第二に、LLM が多様な言い回しを自動的に分類するため、従来必要だった膨大なパターン定義が不要となり、開発コストを大幅に削減できる。第三に、XML による明示的なフロー定義により、対話の再現性と編集可能性が確保される。

## 2 関連研究

対話システムにおけるルールベースと LLM の統合は、近年活発に研究されている分野である。柳本ら [2] は、ルールベースと LLM を組み合わせたユーザ適応型の観光案内対話システムを提案している。このシステムでは、Yes/No 分類器と感情推定器を用いてユーザ意図を分類し、対話ロボットによる観光案内を実現している。しかし、新しいドメインに適用する際には、分類器の再学習が必要となる。鈴木ら [3] は、プロンプトのみでタスク指向型対話を実現する Dialogue Flow Prompt という手法を提案した。この手法では、対話フローをプロンプト内に記述することで、追加の学習なしにタスク指向対話を実現できる。しかし、フェーズ制御には特定の一文の明言が必要であること、同じ指示を複数箇所に記述しないと LLM が従わないことが報告されており、制御可能性に課題がある。小林ら [4] は、ユーザの嗜好や人間関係を考慮する非タスク指向型対話システムを提案している。感情推定と嗜好情報の蓄積により適切な話題選択を実現しているが、予約処理などのタスク遂行機能は持たない。関 [5] は、用例ベースと生成 AI を併用したハイブリッド対話システムを提案している。このシステムでは、LLM を応答生成

に用いることで自然な応答を実現している。

本研究は、これらの既存研究と比較して以下の点で異なる。柳本らの手法が NLU に分類器を用いるのに対し、本研究では LLM の出力を事前定義された列挙型に制約する構造化出力機能 (Structured Outputs<sup>1</sup>) を活用することで、訓練データなしで任意のスロット抽出が可能となり、XML による対話フロー定義の変更のみで新ドメインに適応できる。鈴木らの Dialogue Flow Prompt がプロンプトの書き方に依存するのに対し、本研究では XML による明示的なフロー定義と列挙型による出力制約を組み合わせることで、開発者が意図した対話フローを保証する。関の研究が LLM を応答生成に用いているのに対し、本研究は LLM を入力理解と応答生成の両方に使用しつつ、対話フローの制御は XML で定義されたルールベースのステートマシンが担うことで、制御性と柔軟性を両立している。

## 3 提案手法

### 3.1 応答文の決定方式

本研究の基本的な考え方は「対話フローの制御はルールベースで行い、入力理解と応答生成は LLM で行う」というものである。ルールベースが状態遷移や条件判定を担当することで対話フローの確実性を確保し、LLM がユーザ入力の意図分類・エンティティ抽出および応答文の生成を担当することで多様な表現に柔軟に対応する。最も重要な点は、入力理解において LLM の出力を列挙型に制約することである。列挙型に定義された値しか LLM は返せないため、どのような自由な入力も来ても、出力は必ず事前定義された値のいずれかとなる。これにより、LLM の非決定的な出力をルールベースで確実に扱える決定的な形式に変換できる。

### 3.2 処理フロー

図 2 に提案手法の処理フローを示す。ユーザが自然言語で入力すると、まず LLM により意図とエンティティが構造化データとして出力される。この出力は事前定義された列挙型に制約されるため、必ず決定的な値となる。次に構造化データをインフォメーションステートに格納し、現在の状態に定義されたルールとの照合を行う。ルールにマッチした場合は条件に対応する効果 (変数への値の代入や状態遷移) を実行し、マッチしない場合はグローバルな既定ルールを照合する。最終的に、LLM 応答生成がインフォメーションステートを参照しながらテンプレートを自然言語に変換して応答を出力する。

<sup>1</sup>OpenAI, Structured Outputs, <https://platform.openai.com/docs/guides/structured-outputs>



図 2: 提案手法の処理フロー

### 3.3 全体構成

図 3 に提案システムの全体構成を示す。システムは大きく 3 つのモジュールから構成される。左側の LLM 自然言語理解モジュールが列挙型制約により入力からインテント分類とエンティティ抽出を行い、XML で定義した列挙型により LLM の出力を制約する。中央のステートマシンが XML ルールに基づいて状態遷移と条件照合を行い、インフォメーションステートで対話履歴や変数を一元管理する。右側の LLM 応答生成モジュールがテンプレートを自然な文に変換して出力する。

## 4 実装

### 4.1 シナリオ記述言語

対話フローは XML ベースのシナリオ記述言語で定義する。本実装では、黄らが提案した GSML (GECA Scenario Markup Language) [6, 7] をベースに拡張を行った。主要な要素は以下の通りである。列挙型定義 (Enums) では、LLM の出力を制約する値の集合を定義する。例えば、意図の種類や料理ジャンルの種類などを列挙する。状態定義 (Scene/State) では、会話のシーンと状態を定義し、各状態には複数のテンプレートを定義できる。テンプレート定義 (Category) では、条件 (Predicate)、応答 (Template)、副作用 (Effect)

を定義する。条件は InformationState の変数を参照して判定を行う。

図 4 に、入力例「イタリアンが食べたいな」に対するシステム内部の詳細な処理フローを示す。ユーザ入力はまず LLM により構造化データに変換され、InformationState に格納される。次に Predicate によるルール条件マッチが行われ、マッチした場合は Template の選択と Effect の実行 (変数への値の代入・状態遷移) が行われる。マッチしない場合は GlobalState のルールが照合される。最終的に、LLM がテンプレートと文脈から自然な応答を生成する。

### 4.2 システム構成

提案手法を Python で実装した。StateMachine は GSML シナリオを読み込み、状態遷移を管理する。InformationState は対話中の変数を保持する。DME (Dialogue Management Engine) は入力処理、条件判定、応答生成を実行する。入力理解の Structured Outputs には OpenAI GPT-4o mini を、応答生成には GPT-4o を使用した。

### 4.3 Structured Outputs による入力理解

LLM の Structured Outputs 機能を用いて、ユーザ入力を構造化データに変換する。GSML で定義された列挙型から JSON Schema を自動生成し、LLM の出力を制約する。例えば、PriceRange を Cheap, Medium, High, Any の 4 値で定義すると、ユーザが「安いところがいい」「リーズナブルで」「お手頃なところ」「コスト重視」といった多様な表現を用いても、LLM は必ずこの 4 値のいずれかを返す。これにより、パターン定義や訓練データの収集が不要となり、ルールベースの条件判定に確実に接続できる。

## 5 応用例：観光案内システム

### 5.1 シナリオ設計

観光案内は本機構の適用に適したタスクである。正確な情報提供 (営業時間、料金など) にはルールベースの確実性が必要であり、旅の相談や雑談への対応には LLM の柔軟性が必要だからである。以下では、レストラン検索機能を例にシナリオ設計の概要を示す。

まず、列挙型として意図 (検索、条件設定、確認、否定など) と料理ジャンル (和食、イタリアン、中華など) を定義する。次に、状態として「初期状態」「料理選択」「予算選択」「条件確認」「結果表示」などを定義する。各状態には、条件判定と応答テンプレートを定

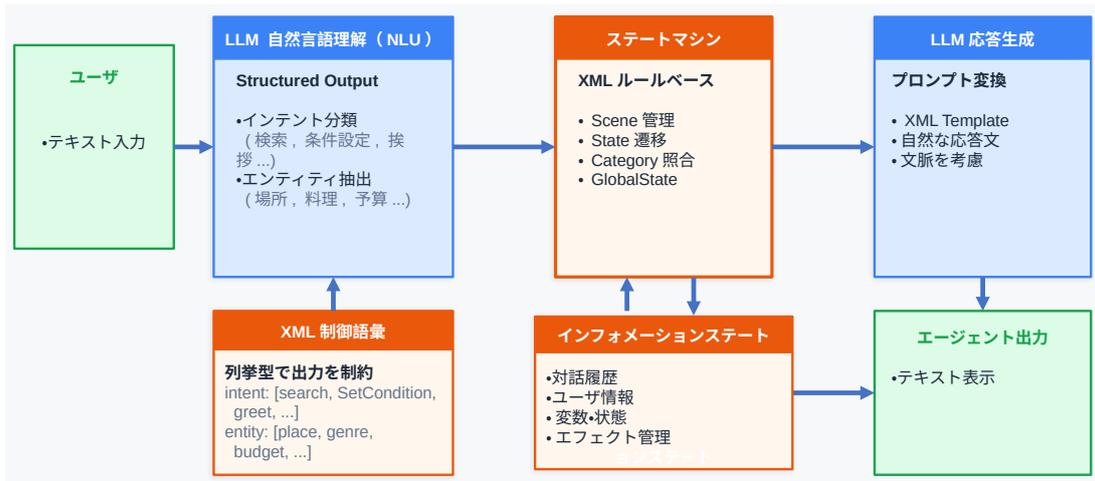


図 3: 提案システムの全体構成

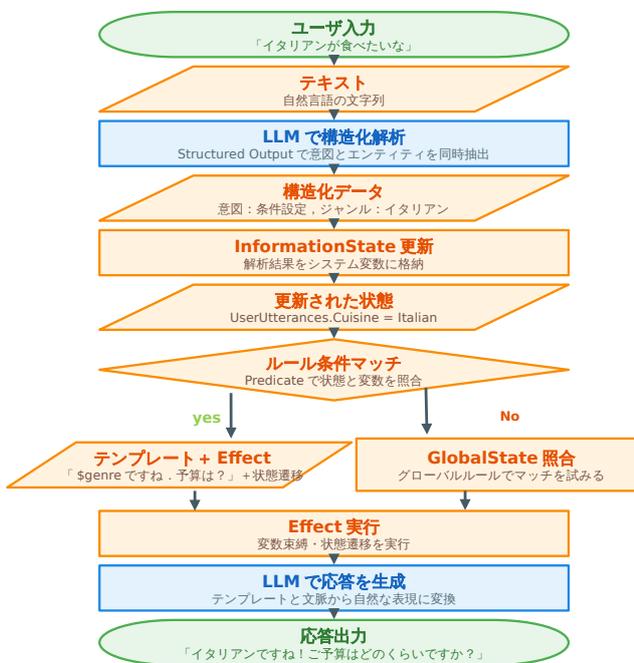


図 4: GSMLに基づく詳細な処理フロー (入力例:「イタリアンが食べたいな」)

義する。例えば、初期状態において「意図が条件設定かつ料理ジャンルが存在する」場合、「(料理ジャンル)ですね。予算はどのくらいですか?」と応答し、予算選択状態に遷移する。

## 5.2 動作例

以下に、提案システムの実際の動作ログを示す。この例では、ユーザがレストランを検索する対話を行っている。ログ中の各行は次の意味を持つ。[LLM] Parsed

は LLM による構造化出力の結果であり、ユーザ入力から抽出された意図 (意図の種類) とエンティティ (料理ジャンル・価格帯などの具体的な値) を示す。[Transition] はステートマシンの状態遷移であり、ルール条件にマッチした結果として移行先のシーンと状態を示す。[Effect] はルールの副作用であり、抽出されたエンティティをインフォメーションステートの変数に代入する操作を示す。

**ユーザ:** イタリアンが食べたいです  
 [LLM] Parsed: {Intention: SetCondition, Cuisine: Italian, Price: None, ...}  
 [Transition] -> Scene: Search, State: AskPrice  
 [Effect] BIND SelectedCuisine = Italian  
**システム:** イタリアンですね! 予算はどのくらいでしょうか?

**ユーザ:** 1000 円をお願いします  
 [LLM] Parsed: {Intention: SetCondition, Price: Cheap, ...}  
 [Transition] -> Scene: Search, State: AskSituation  
 [Effect] BIND SelectedPrice = Cheap  
**システム:** 安めですね! どんなシーンでご利用されますか?

**ユーザ:** 1 人です  
 [LLM] Parsed: {Intention: SetCondition, Situation: Solo, NumPeople: 1, ...}  
**システム:** 一人ですね! 条件を確認いたします。

この動作ログから、提案手法の特徴が確認できる。ユーザの「1000 円をお願いします」という入力に対し、

[LLM] Parsed の結果として価格帯が Price: Cheap という列挙型の値に自動分類されている。従来のルールベースでは「安い」「リーズナブル」などのキーワードを辞書に事前登録して対応するが、「1000 円」のような具体的な金額表現は想定しきれず、辞書に未登録の入力が来ると対話が進行できなくなる。提案手法では LLM が文脈を考慮して適切な列挙値に変換するため、こうした想定外の表現にも対応できる。変換された構造化データに基づき、[Effect] として変数への値の代入 (BIND SelectedPrice = Cheap) が実行され、[Transition] として次の状態 (AskSituation) への遷移が行われる。応答文は LLM がテンプレートと対話文脈から自然な文に変換して生成している。このように、LLM による柔軟な入力理解・応答生成と、ルールベースによる確実な対話フロー制御が両立していることが確認できる。

## 6 まとめ

本研究では、ルールベースと LLM を組み合わせたハイブリッド対話管理機構を提案した。対話フローの制御はルールベースで行い、入力理解と応答生成は LLM で行うことで、制御性と柔軟性を両立させた。

提案手法の特徴は以下の通りである。第一に、対話シナリオを XML スクリプトで定義するため、プログラミングの専門知識がなくても対話フローを設計でき、開発コストを削減できる。第二に、LLM が多様な言い回しを自動的に分類するため、従来必要だった膨大なパターン定義が不要となる。第三に、LLM の出力を事前定義された列挙型に制約することで、柔軟な入力理解と確実な状態管理を両立している。第四に、XML による明示的なフロー定義により、対話の再現性と編集可能性が確保される。今後の課題として、評価実験の実施、マルチモーダル対応がある。

## 参考文献

- [1] 白井宏美: 談話研究の知見に基づく対話システムの現状と課題, 日本音響学会誌, Vol. 78, No. 5, pp. 277–282, 2022 年 5 月.
- [2] Ryo Yanagimoto, Yunosuke Kubo, Miki Oshio, Mikio Nakano, Kenta Yamamoto, and Kazunori Komatani: User-Adaptive Tourist Information Dialogue System with Yes/No Classifier and Sentiment Estimator, Proceedings of the Dialogue Robot Competition 2023, 2023.
- [3] 鈴木順大, 石垣龍馬, 宿里晃太郎, 藤本拓真, 河窪大介, 酒造正樹, 前田英作: ただ一つのプロンプトによるタスク指向型対話システムの実現, 言語処理学会第 30 回年次大会, 神戸, 2024 年 3 月.
- [4] 小林峻也, 萩原将文: ユーザの嗜好や人間関係を考慮する非タスク指向型対話システム, 人工知能学会論文誌, Vol. 31, No. 1, pp. DSF-A.1–10, 2016 年 1 月.
- [5] 関陽介: 用例ベースと生成 AI を併用したハイブリッド対話システム, トランザクションデジタルプラクティス, Vol. 6, No. 1, 2025 年 1 月.
- [6] Hung-Hsuan Huang, Aleksandra Cerekovic, Igor S. Pandzic, Yukiko Nakano, and Toyoaki Nishida: The Design of a Generic Framework for Integrating ECA Components, Proceedings of 7th International Conference of Autonomous Agents and Multiagent Systems (AAMAS'08), Estoril, Portugal, pp. 128–135, May, 2008.
- [7] Hung-Hsuan Huang, Masato Fukuda, Toyoaki Nishida: Development of a Platform of RNN Driven Multimodal Interaction for Embodied Conversational Agents, 19th ACM International Conference on Intelligent Virtual Agents (IVA 2019), Paris, France, July, 2019.