

ユーザの潜在的嗜好・要求を可視化するゴールモデル自動生成手法に関する研究調査

代表研究者 中川博之 岡山大学 学術研究院環境生命自然科学学域 教授

1 はじめに

近年の SNS やスマートフォン、センサデバイス等の普及に伴い、ユーザの行動はライフログと呼ばれるように、明示的あるいは潜在的に履歴が蓄積されている。しかしこれらの情報は十分に連携・活用されているとは言えず、現状はユーザが明示的に利用したいサービスを指定するか、購買履歴、検索履歴などの明示的な情報をもとにサービスが推薦されている。本研究では、ユーザの潜在的な要求や嗜好をゴールモデルとして構造化・可視化するためのゴールモデル自動生成手法を検討する。このようなゴールモデルが自動生成できることにより、各ユーザの要求・嗜好に合理的に合致するサービスを合理的に発見、推薦することが可能となる。

本調査研究では、特に、(1) ユーザレビューに基づいたゴールモデル生成法、(2) 要求記述からのゴールモデル生成法の2種類のゴールモデル自動構築手法を扱うことで、ユーザの潜在的嗜好や要求を可視化する手法の実現可能性について追及した。

2 ゴールモデル

ゴールモデル (Goal Model) は、システムやプロジェクトが達成すべき目標を捉え、記述するために要求工学の分野で用いられる構造的な表現である。このモデルは、ゴール、サブゴール、それらの関係性を明確に階層構造で表現し、関係者がシステムの目的全体を理解するのに役立つ。

ゴールモデルには、機能ゴール (functional goals) —すなわちシステムが「何をするか」—と、非機能ゴール (non-functional goals) —例えば性能、セキュリティ、ユーザビリティなど—の両方が含まれる。これらのゴールをマッピングすることで、開発プロセスを戦略的な目的と一致させることが可能となる。

ゴールモデルでは、満たされるべき要求が「ゴール」として扱われる。これらのゴールは階層的に構造化され、親ゴール (parent goals) とサブゴール (sub-goals) の関係が定義される。親ゴールは抽象的な目的を表し、サブゴールはより具体的で詳細な目標である。サブゴールがすべて達成されたとき、親ゴールも達成されたとみなされる。

ゴールモデルの記法は複数存在し、要求工学分野でさまざまに用いられている。代表的な記法には、KAOS [1]、i* [2]、NFR Framework [3]、AGORA [4] などがある。本論文では、唯一のルートゴールを持ち、機能ゴールを中心にゴールの洗練化を進める KAOS に基づいたゴールモデルの記法を用いる。

ゴールモデルを用いた要求分析および要求記述の利点は以下のとおりである。

要求間の競合の特定：ゴールモデルは、ユーザ要求間の潜在的な競合を体系的に明示できる。たとえば、あるユーザはアプリに強固なセキュリティ機能を求める一方で、別のユーザは利便性や簡単なアクセスを重視することがある。強化されたセキュリティは使いやすさを損なう可能性があるため、これらの要求は相反することがある。ゴールモデルを使えば、こうした対立を明確化し、設計段階でバランスの取れた解決策を見出すことが可能となる。

複数の要求に共通する目的の発見：ゴールモデルにより、異なるユーザ要求を共通の親ゴールの下にグループ化することで、それらの背景にある目的を明らかにできる。たとえば、「読み込みの高速化」「ナビゲーションの簡素化」「直感的なデザイン」といった要求は、「ユーザ体験の向上」という親ゴールに統合され得る。このように整理することで、開発者はシステムのユーザビリティ最適化に集中しやすくなる。

トレードオフと優先順位付けを支援：複数のユーザ要求が同一の親ゴールに関連付けられると、ゴールモデルは開発者がそれらを比較・調整しやすくなる。たとえば、あるレビューではセキュリティ向上 (例：多要素認証) を求め、別のレビューでは利便性 (例：パスワード入力 of 簡素化) を重視している場合、両者は

「ユーザ体験の向上」という親ゴールに結びつく。ゴールモデルは、影響の大きさや重要度に応じてどの機能を優先すべきかを判断する助けになる。これにより、リソースを効率的に配分し、ユーザの中核的な期待に応える設計が可能となる。

追跡・追跡性の提供：ゴールモデルは、ゴール間の関係を追跡することで、言及された要求や課題を追跡・記録する手段を提供する。レビューをゴールにマッピングすることで、それぞれの要求に対する改善や対応状況を追跡でき、ユーザ満足度と開発者への信頼性向上につながる。

これらの利点により、本論文では自然言語で記述されたユーザの要求・嗜好から自動的にゴールモデルを生成する手法を検討する。

3 レビューからのゴールモデル自動生成

3-1 概要

本研究では、ユーザレビューから要求を抽出し、ゴールモデルによる可視化を提供することでゴールモデルを自動的に生成する手法を検討した。この手法では、まずユーザレビューを入力として与え、要求を抽出する。そして、抽出された要求を「ゴール」として扱い、それらを用いてゴールモデルを構築する。このゴールモデルは、抽出された要求を記述するとともに、それらの関係性を可視化する。最終的な出力として生成されるゴールモデルは、ユーザ要求を俯瞰的に理解する助けとなるものである。

以下に、提案手法の全体的な処理の流れを説明する。

■ **前処理：**最初に、レビュー中の単語を語の原型へ変換（レマタイズ）し、ストップワード（stop words）を除去する。ストップワードには「a」「an」「the」などの一般的な語のほか、「app」「like」「use」など、ユーザ要求を記述する上で意味の薄い単語も含める。さらに、5語未満の短いレビューも除去する。これは、このようなレビューは文脈や詳細が不足しており、有益な情報を提供しないことが多いためである。前処理後に残ったレビューを、ゴールモデル生成の入力として使用する。

■ **ゴールモデル生成：**提案手法は、トップダウンでゴールを洗練しながらゴールモデルを生成していく。最上位のゴール（ルートゴール）は、すべてのレビューを包含する。そして、そのレビュー集合を段階的により小さなレビュー集合に分割し、それぞれをサブゴールとして扱う。この分割（洗練）を繰り返すことで、末端の葉ゴール（leaf goals）に至るまで、階層構造を構築する。本研究では、ゴールの洗練手法として、LDA トピックモデルを用いたクラスタリング手法、距離ベースのクラスタリング手法、LLM（大規模言語モデル）を用いた手法の3種類の手法を用いる。これらはいずれも、クラスタリングの要領でゴールモデルを構築するものである。このうち、LLMを用いた手法は最も高精度かつ人間にとって分かりやすいゴールを生成することができるが、大量のレビューには対応できない。一方、LDA および距離ベースの手法は大量のレビューを分類することが、精度はレビュー数に依存する。そのため、ゴールに含まれるレビュー数に応じて、どの手法を使うかを動的に切り替える。

3-2 ゴールモデル生成手順

図1は、提案手法の全体的な構成を示している。ユーザレビューの件数に基づいて、以下の3つの手法のいずれかを適用する：

- LDA トピックモデルを用いたクラスタリング手法（多数レビュー向け）
- 距離ベースのクラスタリング手法（中程度レビュー向け）
- LLM ベースの手法（小数レビュー向け）

具体的なゴールモデル生成手順は以下のとおりである。

4. LLMによるゴール記述の洗練化

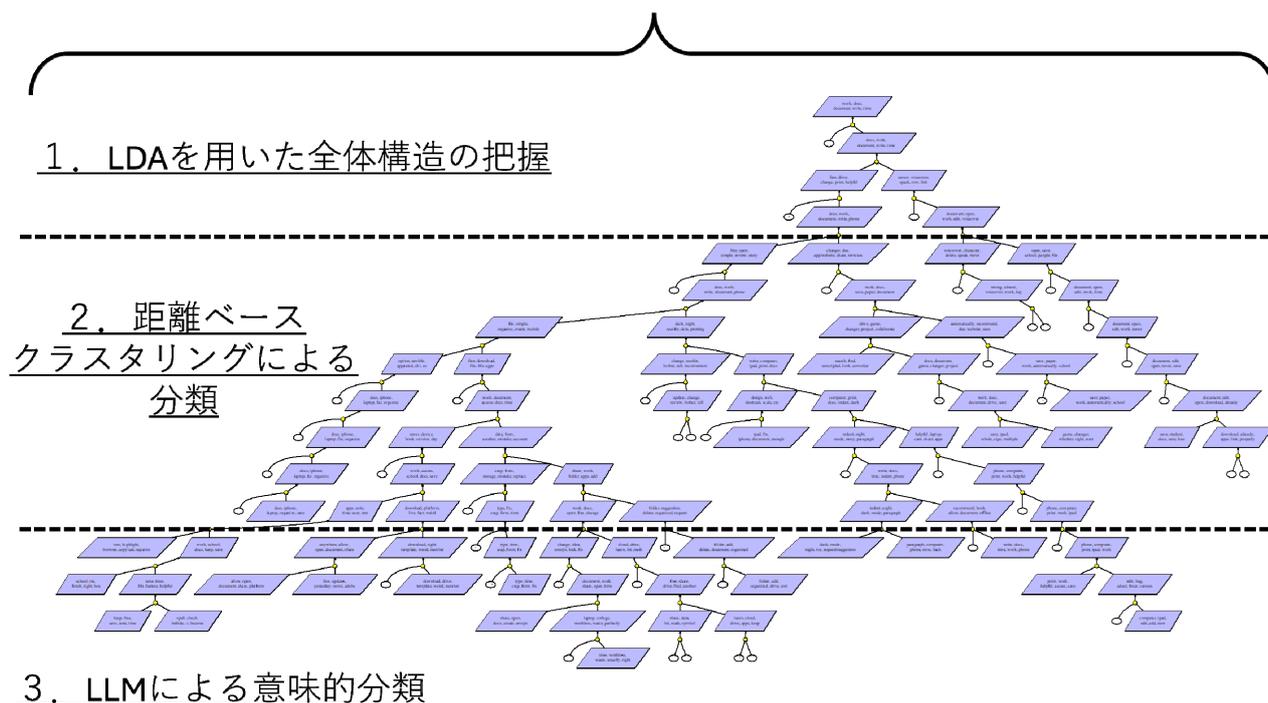


図1：ユーザーレビューに基づいたゴールモデルの自動生成

■ ゴールモデル生成手順

1. 初期の洗練化、つまり分類は大量のレビューを扱う必要があるため、**LDA トピックモデルを用いたクラスタリング手法**を適用し、レビューを複数のトピックに分類する。このトピックがサブゴールに相当する。各トピックをゴールに見立てて、ゴール分解とクラスタリングとを対応付ける。
2. レビューを分類すると、対応する各ゴールに含まれるレビュー数が減少する。レビュー数が一定以下になると、LDAの精度が落ちるため、**距離ベースのクラスタリング手法**に切り替える。
3. LDAや距離ベースの手法に比べて、CPT-4などの大規模言語モデル（LLM）はより分かりやすいゴール記述を生成することができる。ただし、トークン数の制限により、多くのレビューを統合して扱うことができない。そのため、レビュー数が一定数（例えば100件）未満にまで減少したタイミングで、**LLMを用いた要約・分類**に移る。
4. 以上の処理でゴールモデルの構造が決定するが、LDAと距離ベースのクラスタリングによって生成されたゴールにも、LLMを用いてランダムに選んだレビューの要約をゴールのラベル（ゴール記述）として付与する。

ここで、トピックモデル（Topic Models）とは、文書中に潜在する意味的な構造を発見するための統計的手法である [5]。トピックモデリングは教師なし学習の一種であり、データ中のパターンを認識することでトピックを抽出・特定する。単語の出現頻度や共起パターンを分析することで、文書に含まれるさまざまなトピックを同定する。代表的なトピックモデルには、LDA（Latent Dirichlet Allocation） [6]、PLST [7]、Pachinko Allocation Model [8] などがあり、レビューの分類やクラスタリングによく用いられている。本研究では、最も広く使われている手法であるLDAを用いる。ここで、「Latent（潜在的）」は直接観測できない隠れた意味構造を、「Dirichlet（ディリクレ）」は各文書におけるトピック分布や、トピックにおける単語分布がディリクレ分布に従うという仮定を、「Allocation（割り当て）」は各単語がどのトピックから生成されたかを確率的に割り当てることを意味する。他のトピックモデルと比較して、LDAは、潜在的なトピック構造を的確に捉える能力が高く、精度と安定性に優れている点、計算効率が良く、処理時間が短いため、大規模なデータセットに対しても迅速に結果を出すことができる点、解釈性が高い、つまり、生成されるトピ

ックと単語分布が明確な意味構造を持っており、分析者がトピックの内容を理解しやすい点で優れているとされている [9].

3つの手法の具体的内容は以下のとおりである.

LDA トピックモデルを用いたクラスタリング手法 (多数レビュー向け): 大量のレビューがある場合, LDA (Latent Dirichlet Allocation) を用いて, レビュー文書群を複数のトピックに分類する. この処理はトップダウン方式で行われ, 親ゴールに含まれるレビューに LDA を適用して, 複数のトピック (=サブゴール) に分解する. トピック数 K を事前に定義すると, LDA は K 個のトピックを生成し, それぞれが親ゴールのサブゴールに対応する. 各レビューは, トピックごとの単語分布に基づいて, トピック分布 (確率) が与えられ, 最も確率の高いトピックがそのレビューの代表トピックとされる. このような確率的なフレームワークにより, レビュー文書群の中に潜む意味構造 (トピック) を捉えることができる. 各トピックは, 類似した内容のレビュー群に対応し, これをルートゴール付近のゴール群とする. この際, 各トピックに属するレビューを GPT-4 に入力し, それらを要約・命名させることで, ゴールラベルを生成する. これにより, 各トピックが「どのような要求に関連しているか」が明示される. 最後に, トピック間の関係性を分析し, ゴール階層構造を構築する.

距離クラスタリング手法 (中程度レビュー向け): レビュー数がある程度まで減少するまでは LDA を用いてゴールモデルを階層的に拡張していくが, レビュー数が減少し, LDA の精度が距離ベース手法よりも低下するレベルに達した場合は, 距離ベース手法に切り替える. レビュー数が中規模の場合, LDA のような確率的な手法よりも, 意味ベクトルに基づいたクラスタリングの方が精度が高い. まず, SBERT (Sentence-BERT) を用いてレビューを意味ベクトルに変換し, クラスタリング (たとえば階層的クラスタリング) を行う. 次に, 各クラスタに含まれるレビューを GPT-4 などの LLM に要約させ, ゴールラベルを生成する. 最後に, クラスタ間の包含関係に基づき, ゴール階層構造を構築する.

LLM ベースの手法 (少数レビュー向け): レビュー数が少ない場合, LDA やクラスタリングによる意味分類は困難である. その一方で, LLM を用いた意味的な分類が可能となる. 従って, GPT-4 などの LLM を用いて, レビューのグループ化やゴール記述の決定, レビューの意図や目的の抽出をスクリプトを利用することで実施する. ゴール記述の決定は, ゴールモデル上方に位置するゴール群に対しても有効であり, 関連するレビューをいくつか選択して, ゴール記述を洗練化する.

3-3 ゴールモデル構築実験

提案手法の有効性を実証するために, 複数アプリケーションのレビューを対象に評価実験を実施した. 提案手法は, レビュー集合をクラスタリングし, クラスタごとにゴール (目標) として構造化することで, 最終的にゴールモデルを生成するものである. 本評価では, 主に次の3つの観点から有効性を検証した.

- ・クラスタリング結果の意味的一貫性
- ・ゴールモデルの階層構造の妥当性
- ・ユーザーレビューに基づいたゴールの抽出 (生成) 精度

評価対象には, 実際に多くのレビューが集まるアプリとして, コミュニケーションアプリケーションである LINE と, フリーマーケットアプリであるメルカリを選定した. これらのアプリケーションに対して, Google Play ストアからレビューを収集した. 収集時の条件として, 評価期間は過去1年間, 5語未満のレビューやスパムとみなされるレビューを除外し, 最終的に, LINE 約4,000件, メルカリ約3,500件のレビューを評価対象として用いた.

提案手法の有効性を検証するために, 以下の3手法に対してゴールモデルを構築し, 得られたゴールモデルを評価した.

- ・LDA 単独: レビュー全体に対して LDA のみを適用し, トピックをゴールとして扱う.

- ・ 距離クラスタリング単独：SBERT + 階層的クラスタリングのみでゴールを構築.
- ・ 提案手法 (ハイブリッド)：LDA → クラスタリング → LLM をレビュー数に応じて切り替える方式.

意味的一貫性の評価には、トピックまたはクラスタの意味的一貫性を計測するために、トピックコヒーレンスコア (C_v) を用いた。これは、同一クラスタ内の上位単語間の意味的一致度を測る指標であり、数値が高いほど意味的に優れた分類であることを示すものである。実験結果から、いずれのゴールモデルでも、ハイブリッド手法が、他の手法よりも高いコヒーレンスを持つことが確認された。これは、レビュー数に応じた手法選択と LLM による要約ラベリングが効果的であることを示している。

意味的一貫性の定量評価に加え、ゴールモデルの品質に関する主観的評価も実施した。要求工学の知識を持つ3名の大学院生に対し、各モデルの(i)ラベルの明瞭性(わかりやすさ)、(ii)階層構造の妥当性、(iii)開発での有用性の3つの観点から5段階評価で採点してもらった。結果、提案手法に対する平均スコアは他の手法よりも優れていた。被験者からは、「サブゴールに付けられた名前が直感的で、レビュー内容と対応している」、「レビューを意味的にまとめているので、似た意見が集約されていて見やすい」、「LLM を使うことで、モデル全体が自然に読める構造になっている」といった肯定的なフィードバックが得られた。ユーザレビューに基づいたゴールの抽出精度については、再現率と適合率、F 値を用いた評価を実施した。その結果、提案手法は、高いF 値によるゴール抽出(生成)が実現されていることが確認できた。

以上の評価より、提案手法はコヒーレンス、可読性、構造の妥当性のいずれにおいても既存手法より優れており、ユーザレビューから高品質なゴールモデルを生成できることが確認された。さらに、本研究では、GoogleDocs や YouTube など様々なアプリケーションのゴールモデル構築実験も実施した。図2は YouTube のレビューから生成したゴールモデルである。

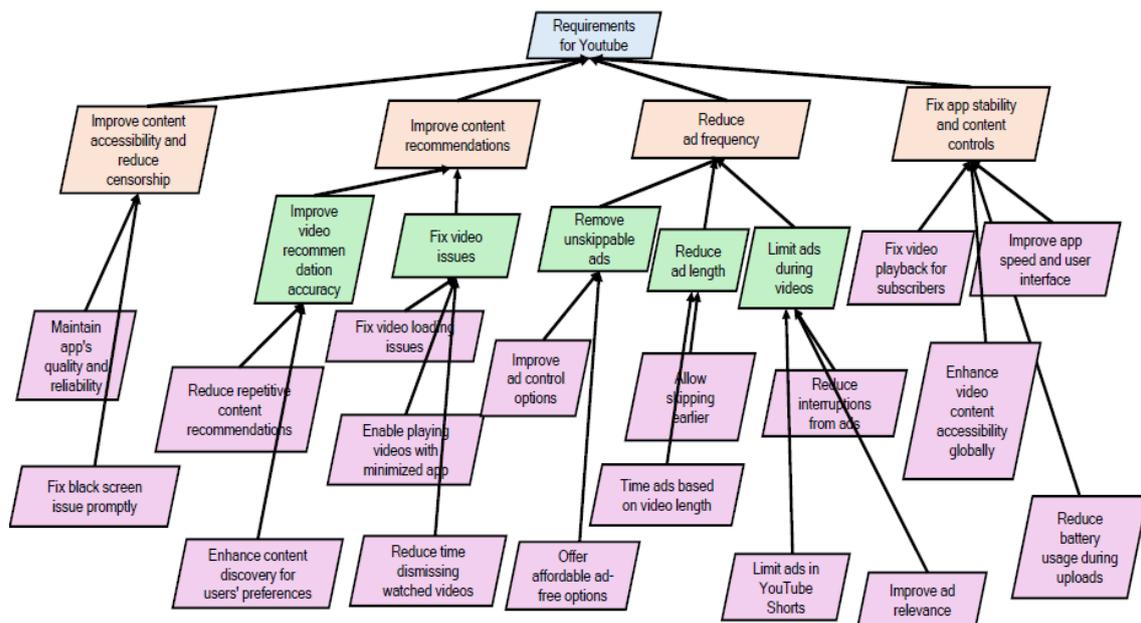


図2：YouTube のレビューから生成したゴールモデル

4 LLM を用いたゴールモデル構築エディタ

4-1 概要

ゴールモデルは俯瞰的な要求間の関係を定義するものであるが、ゴールモデル構築にあたっては、構成要

素であるゴールや、ゴール間の関係を見出す必要がある。これを人力で行う場合、ゴールやゴール同士の関係が見逃される可能性があり、正確性や妥当性は保証されない。現在の研究において自然言語から要求に関わる部分のみを抽出し、抽出した要求同士の関連性を推測するプロセスの自動化は困難である。本調査研究では、LLM を用いたゴールモデル自動生成エディタを設計、実装した。本エディタは、開発予定のシステム概要を決められた命令文と共に LLM に入力し、返ってきた文を自然言語処理により更に整形することで、視覚的に把握しやすいゴールモデルを自動的に作成する。これにより、従来研究の課題であった、自然言語からの自動要求抽出、要求間の関連性の自動推測を目指した。

4-2 LLM を用いたゴールモデル構築エディタ

本調査研究では、大規模言語モデル (LLM) を活用した自動ゴールモデリングエディタを設計・実装した。本エディタは、ユーザレビューや仕様記述などの自然言語文書から KAOS ゴールモデルを自動生成することで、ソフトウェア開発時の要求定義作業を支援したり、ユーザ要求を把握・可視化することを目的としている。本手法の主要な特徴は次に挙げる 3 点である。(1) **自然言語からの自動ゴール抽出**：ユーザ入力 (例：レビュー文、要望記述) から、目標とされる意図や振る舞いを LLM が抽出し、ゴール候補を生成する。(2) **KAOS メタモデルに準拠した構造化**：抽出されたゴール、エージェント、ドメインプロパティなどの要素は、KAOS の構文規則に則ってモデル内に自動的に配置される。(3) **インタラクティブなプロンプト設計支援**：利用者の入力やモデリング文脈に応じて、適切なプロンプトを内部的に設計・調整し、出力の品質と整合性を確保する。

本研究におけるゴールモデルの自動構築手法は、以下の 3 つのステップにより構成される。

1. LLM による要求の抽出
2. ゴールモデルの再構造化
3. 要求の分割

入力として与えられる自然言語で記述された文書はまず前処理され、入力文の意味構造を捉えるために複数のプロンプトとともに LLM に送信される。得られた出力は、専用の構文変換モジュールにより KAOS モデルの構成要素にマッピングされ、エディタ上に視覚的に表示される。さらに、開発者が手動で加えた修正も LLM へのフィードバックとして再利用され、モデリング支援が継続的に強化される。

本手法は、従来の手法に比べて、自然言語で書かれた曖昧な文からも有効なゴールを抽出できる点、ゴールと手段の関係性を含む階層的構造を自動で生成できる点、モデル構築の負担を軽減し、反復的な要求検討を促進する点において優位性を持つと考えられ、本エディタは初期の要求定義・分析のための支援ツールとして機能することが期待できる。

4-3 システム構成

本節では、提案手法を実現するために開発したゴールモデルエディタの実装について説明する。本エディタ (図 3) は、自然言語入力から KAOS ゴールモデルを自動生成・可視化するインタラクティブなモデリングツールであり、バックエンドで大規模言語モデル (LLM) を統合している。本エディタは、入力インタフェース、プロンプト生成モジュール、LLM 呼び出しエンジン、KAOS モデルレンダリングエンジンの 4 つのコンポーネントから構成される。入力インタフェースはユーザが自然言語のテキスト (レビュー、要望、仕様記述など) を入力するためのエディタ部であり、プロンプト生成モジュールは入力文の文脈や構造を解析し、LLM への問い合わせ用プロンプトを動的に生成するモジュールである。LLM 呼び出しエンジンは OpenAI GPT-4 API と連携し、プロンプトに基づいて KAOS モデル要素 (ゴール、エージェント、ドメイン仮定など) を抽出する役割を担い、KAOS モデルレンダリングエンジンは LLM 呼び出しエンジンの実行結果に含まれる要素群を KAOS メタモデルに基づいて整理・配置し、Graphviz を用いて視覚的に表示する機能を有する。これらの機能

を持つエディタを用いることで、ユーザは生成されたモデルを GUI 上で編集することも可能であり、LLM による再提案やラベル修正を対話的に実行できる。

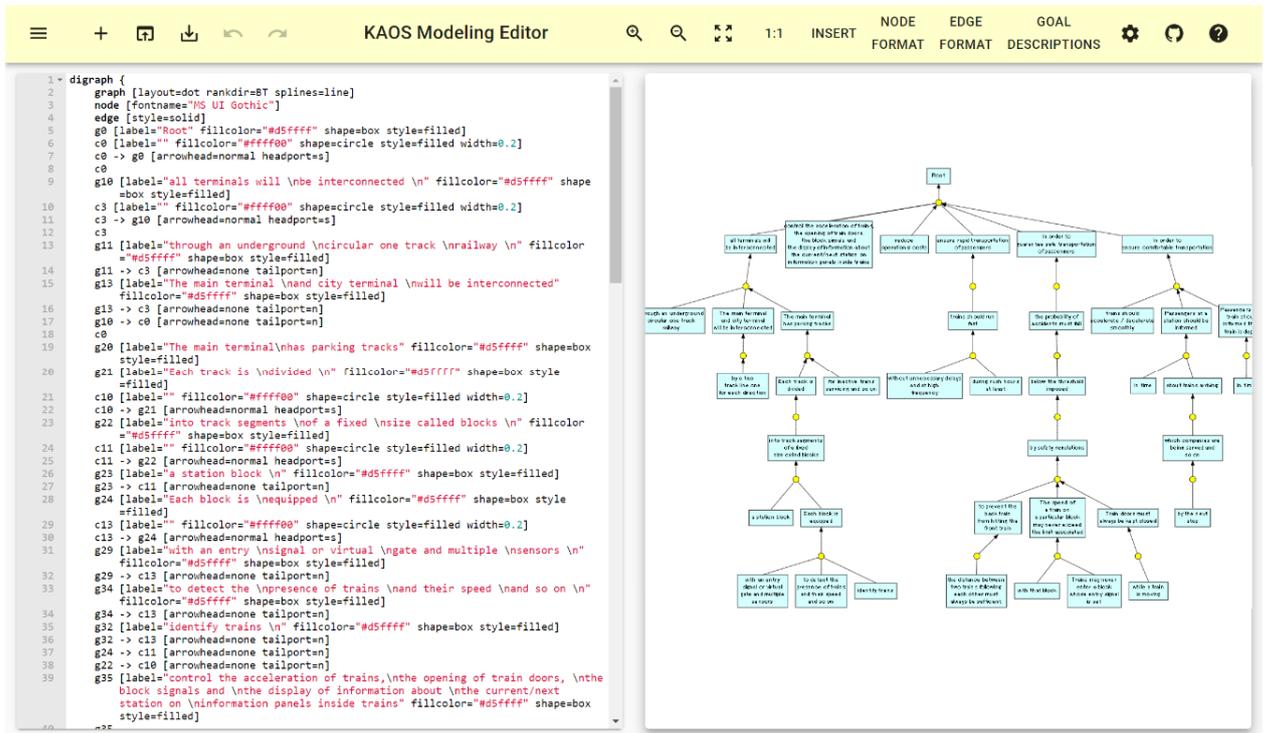


図 3：ゴールモデル生成エディタ

提案手法は LLM を利用してゴールモデルを生成する。LLM に対しては、モデル要素ごとに設計されたプロンプトを用いて、ゴールモデルの構成要素となる出力を得る。図 4 は、ゴールモデル生成時に用いるプロンプトである。

The following text is a software requirements description. Please extract only the sentences relevant to the goals that the software should meet. Next, extract the goals that the software should meet from those sentences, using the wording and expressions from the original text as much as possible, and assign numbers to them. Divide each goal into sub-goals as much as possible and conclude each goal as a single sentence.

Furthermore, if one goal is necessary to achieve another goal, please make it explicit. In such cases, assign numbers like A.1 for the sub-goals required to achieve goal A. For example, if the second goal requires two sub-goals, assign 2.1 and 2.2 to them. Please perform this task until goals and their sub-goals form as deep a tree structure as possible.

Please never include 'and' and 'or' in the goals. If you must include commas, 'and,' or 'or' in a goal, please break down that goal into another goal with the comma, ' and,' or 'or' as the object. Do not extract goals that are not present in the original text. The output is the goals and subgoals only, in plain text, with no indent tab and bold letters. No preface or additional text should be included.

図 4：ゴールモデル生成時用プロンプト

このプロンプトは、自然言語文書と共に LLM に与えられ、結果として得られた各要素は、Python で実装した構文変換モジュールにより KAOS メタモデルにマッピングされ、最終的に階層構造と依存関係を可視化するゴールモデルが生成される。LLM によりゴールモデルを生成した後に、提案エディタは、自然言語処理を用いてゴールモデルの再構造化を行う。この再構造化アルゴリズムを図 5 に示す。

```
subgoals = root ゴール. 子ゴール集合
再構造化済のゴール集合 = {root ゴール}
for goal in subgoals :
    単語を分析 (goal)
    親候補 = 再構造化済のゴール集合の内,
              goal と共通する単語数が最大のものの集合
    if len(親候補) == 1 :
        親候補 [0].addSubgoal (goal)
        再構造化済のゴール集合.add(親候補 [0])
    else :
        topGoal = 親候補を深さ順に並べ替えた内
                  最も深さが小さいゴール
        topGoal.addSubgoal (goal)
        再構造化済のゴール集合.add(topGoal)
```

図 5 : ゴールの再構造化アルゴリズム

4-4 評価

本研究では、ゴールモデルエディタの有効性を検証するために実施した予備的な評価実験について述べる。本評価では、図書館管理システム、電車制御システム、会議日程調整システムの 3 つの要求記述を対象として、生成されたゴールモデルの品質を定性的に分析し、LLM ベースの自動モデリングの実用可能性を確認した。各要求記述に対して、提案するエディタを用いてゴールモデルを生成した。

生成されたゴールモデルに対し、LLM による自動抽出結果とユーザの編集操作の関係を分析した。具体的には、過不足なく要求を抽出できているか、ゴールモデル再構造化の精度が高いか、要求の分割の精度が高いかの 3 種類の評価を実施した。結果として、ゴール・ソフトゴール・エージェント・ドメイン仮定といった構成要素が一通り抽出され、ゴールモデルとして基本的な構造が形成されることが確認された。また、ユーザによる軽微な手動修正により、モデルの粒度調整や誤認識の補正が行えることから、本ツールが対話的なモデリング支援に有効であることも確認された。一方で、LLM の出力結果には不完全な構成要素や文脈誤認による誤抽出も一部含まれており、完全自動化には限界があることも確認された。そのため、ユーザのフィードバックを活用した再抽出・修正の仕組みが今後の改良課題である。しかしながら、総じて、提案エディタは、初期段階のゴールモデリング支援として有用であり、要求分析における LLM 活用の可能性を示す成果が得られた。

4-5 関連研究

本節では、ゴールモデリングに関する既存研究、および自然言語処理技術を用いた要求抽出・モデル生成の分野における関連研究について概観する。

ゴールモデリング支援ツール：KAOS や i*, UML Use Case などの要求モデリング手法において、モデリング作業を支援するツールは多数提案されている。たとえば、OpenOME や KAOSTool などは、ユーザが手動で要素を定義・編集することを支援する GUI ベースのエディタである。しかしながら、これらはいくまで手動操作に依存しており、自然言語からの自動的な要素抽出や構造化には対応していない。一方で、半自動的にゴールモデルを構築するアプローチとして、シナリオ記述からのモデル変換や、事前定義されたルールベースによる要素抽出、対話に基づいたゴールモデル構築[10]が提案されているが、汎用性や柔軟性に課題がある。

自然言語による要求抽出：自然言語からの要求抽出に関する研究は近年活発に行われており、特にユーザレビューや仕様文書から有用な情報を抽出する手法が注目されている。LDA や TextRank, BERT などを用いた手法では、要求の要約や分類、キーワード抽出が可能であるが、抽出結果は単なるリストにとどまり、要求間の意味的関係や階層性を考慮することは難しい。また、テンプレートベースやルールベースの変換アプローチ[11]も提案されているが、ドメイン依存性が高く、柔軟な適用には限界がある。

LLM を活用したモデリング支援：近年の大規模言語モデル (LLM) の進展により、自然言語からの構造的情報抽出やコード生成が大きく進展している。要求工学の分野でも、LLM を用いた要件分析や仕様記述の支援が模索され始めている。例えば、ChatGPT や GPT-4 を用いて、ユーザストーリーの生成や要件の分類を支援する試みが報告されているが、ゴール指向モデル (KAOS) との統合的なアプローチは限定的である。また、LLM の出力をそのままモデルとして取り込むのではなく、構文規則に準拠した構造として再構築する枠組みはほとんど存在しない。

以上を踏まえると、本研究は以下の2点において独自性を有する。ゴール・ソフトゴール・エージェント・ドメイン仮定といった構成要素を、LLM を介して自動的に抽出・構造化する点で、自然言語からゴールモデルへの一貫した変換が可能である。また、開発者の操作を取り入れつつ、反復的にモデルを改善可能なユーザインタフェースを提供する点で、インタラクティブなモデリング支援を実現している。このように、本研究は従来の自動抽出手法と手動モデリングツールの中間に位置し、LLM の柔軟性と KAOS の構造化性を両立する新たな支援環境として位置づけられる。

5 まとめ

本調査研究は、ユーザの潜在的な要求や嗜好を可視化・解析することを目指したものであり、ユーザ要求の構造的表現が可能なゴールモデルを自動生成する手法について、その実現可能性を2つの自動生成手法を設計・実装することにより調査した。具体的には、(i) ユーザレビューに基づいたゴールモデル生成法と、(ii) 要求記述からのゴールモデル生成法を提案し、有効性を評価した。前者は、自然言語で書かれたユーザレビューをクラスタリング技術により構造化し、要求モデルとして可視化するという、要求工学と自然言語処理、深層学習を融合した手法である。この生成法は、ゴールモデルの生成に、LDA、距離ベースのクラスタリング技術、LLM を利用している点の特徴である。後者は、自然言語からゴールモデルへの一貫した変換を実現するゴールモデル生成法と洗練支援手法である。その特徴は、ゴールのみならず、ソフトゴールやエージェント、ドメイン仮定といったゴールモデルに関連する構成要素を、LLM を介して自動的に抽出・構造化する点と、反復的にモデルを改善可能なエディタを提供することにより、対話的なゴールモデリング支援を実現している点にある。このアプローチは、自動抽出手法と手動モデリングツールの中間に位置し、LLM の柔軟性とゴールモデルの形式的構造を両立する構築支援技術として位置づけられる。これらの2つのゴールモデル生成法の発展・融合により、ユーザの潜在的な要求・嗜好の構造化・可視化が期待できる。

謝辞

本研究をご支援下さった電気通信普及財団様，大阪大学大学院情報科学研究科 土屋 達弘教授，ならびに同研究を積極的に推進してくれた，大阪大学大学院情報科学研究科 博士後期課程 任 帥才氏，博士前期課程 渡辺 啓太郎氏に心より感謝致します。

【参考文献】

- [1] A. Dardenne, A. van Lamsweerde, and S. Fickas. Goal-directed requirements acquisition. *Science of Computer Programming*, 20(1-2):3-50, Apr. 1993.
- [2] E. S. Yu. Towards modelling and reasoning support for early-phase requirements engineering. In *Proceedings of ISRE'97: 3rd IEEE International Symposium on Requirements Engineering*, pages 226-235. IEEE, 1997.
- [3] J. Mylopoulos, L. Chung, and B. Nixon. Representing and using nonfunctional requirements: A process-oriented approach. *IEEE Transactions on software engineering*, 18(6):483-497, 1992.
- [4] H. Kaiya, H. Horai, and M. Saeki. Agora: Attributed goal-oriented requirements analysis method. In *Proceedings IEEE joint international conference on requirements engineering*, pages 13-22. Ieee, 2002.
- [5] C. H. Papadimitriou, P. Raghavan, H. Tamaki, and S. Vempala. Latent semantic indexing: A probabilistic analysis. *Journal of Computer and System Sciences*, 61(2):217-235, 2000.
- [6] D. M. Blei, A. Y. Ng, and M. I. Jordan. Latent dirichlet allocation. *Journal of machine Learning research*, 3(Jan):993-1022, 2003.
- [7] T. Hofmann. Probabilistic latent semantic indexing. In *Proceedings of the 22nd annual international ACM SIGIR conference on Research and development in information retrieval*, pages 50-57, 1999.
- [8] W. Li and A. McCallum. Pachinko allocation: Dag-structured mixture models of topic correlations. In *Proceedings of the 23rd international conference on Machine learning*, pages 577-584, 2006.
- [9] A. Abdelrazek, Y. Eid, E. Gawish, W. Medhat, and A. Hassan. Topic modeling algorithms and applications: A survey. *Information Systems*, 112:102131, 2023.
- [10] H. Nakagawa, H. Shimada, and T. Tsuchiya, "Interactive goal model construction based on a flow of questions," *IEICE Transactions on Information and Systems*, vol. E103.D, no.6, pp.1309-1318, 2020.
- [11] H. Shimada, H. Nakagawa, and T. Tsuchiya, "Constructing a goal model from requirements descriptions based on extraction rules," *Requirements Engineering for Internet of Things*, eds. by M. Kamalrudin, S. Ahmad, and N. Ikram, pp.175-188, Springer Singapore, Singapore, 2018.

〈発表資料〉

題名	掲載誌・学会名等	発表年月
Efficient Code Reachability Analysis and Visualization Using Probabilistic Model Checking	International Journal of Software Engineering and Knowledge Engineering	2025年6月 (掲載決定)
後付けUIにおける機能の整合性検証	ヒューマンインタフェース学会論文誌, Vol. 27, No. 2, pp. 97-112	2025年5月
性格特性に応じたユーザモデル半自動生成手法	日本ソフトウェア科学会 学会誌『コンピュータソフトウェア』	2025年9月 (掲載決定)

An XAI-Based Meta-parameter Tuning for Time-series Forecasting	Annals of Mathematics and Artificial Intelligence, Springer	2025年3月
v-KAOS Editor: LLMを活用した自動ゴールモデリングエディタ	電子情報通信学会論文誌, Vol. J107-D, No. 10, pp. 454-462	2024年10月
Detection and Classification of Concept Drift in Streaming Process Discovery	The 29th International Conference on Knowledge-Based and Intelligent Information & Engineering Systems (KES 2025)	2025年9月 (発表決定)
Requirements Analysis of a High-Precision Search Function for Reusable RPA Bot Development	The 29th International Conference on Knowledge-Based and Intelligent Information & Engineering Systems (KES 2025)	2025年9月 (発表決定)
Exhaustive Model Identification on Process Mining	The 20th International Conference on Evaluation of Novel Approaches to Software Engineering (ENASE 2025), pp. 449-456	2025年4月
Facility Layout Generation using Hierarchical Reinforcement Learning	The 17th International Conference on Agents and Artificial Intelligence (ICAART 2025), Vol. 3, pp. 150-157	2025年2月
Towards Log-based Execution Status Estimation Using Graph Neural Networks	The 31st Asia-Pacific Software Engineering Conference (APSEC 2024), pp. 492-496	2024年12月
Code Reachability Visualization Based on Probabilistic Model Checking	The 36th International Conference on Software Engineering & Knowledge Engineering (SEKE 2024)	2024年10月 (Best paper award 授賞)
Review-Based Bot Smell Classification in Robotic Process Automation	The 28th International Conference on Knowledge-Based and Intelligent Information & Engineering Systems (KES 2024)	2024年9月
Combining Prompts with Examples to Enhance LLM-Based Requirement Elicitation	The IEEE 48th Computers, Software, and Applications Conference (COMPSAC 2024)	2024年7月
Self-adaptive System Implementation Framework Considering Execution Time Uncertainty	The IEEE 48th Computers, Software, and Applications Conference (COMPSAC 2024)	2024年7月