卒 業 論 文

一点物売買プラットフォームの動作検証を目的とした社会的シナリオに則ったデータ生成エージェントの開発

2025 年度

指導教員 植松 幸生 准教授

(6322087)

三笠 悠太郎

東京理科大学 創域理工学部 情報計算科学科

提出日： 2026 年 2 月 15 日

一点物売買プラットフォームの動作検証を目的とした社会的シナリオに則ったデータ生成エージェントの開発

要旨

近年，クリエイターエコノミーの発展に伴い，作品単体の価値だけでなく，その背景にあるストーリーやクリエイターとファンが形成するコミュニティの文脈を統合して流通させる Web プラットフォームの重要性が高まっている．このようなプラットフォームにおいて，ユーザー間の社会的相互作用や共同制作プロジェクトといった関係性を記録・表現する必要があるため，そのデータベースは社会的シナリオを反映した複雑なリレーション構造となる傾向にある．このようなプラットフォームの開発初期段階において，UI 表示や検索システムの有用性を検証するためには，サービス固有のデータ構造と整合性を保った大量のテストデータが不可欠である．しかしながら，社会的シナリオを反映したデータを手動で大量に用意することは困難であり，従来のランダムなダミーデータ生成手法では，リレーションの整合性やコンテンツの意味的な文脈を再現できないという課題があった．

本研究では，大規模言語モデル（LLM）を用いて，ユーザの社会的背景や文脈を記述した社会的シナリオを構築し，それに基づいてサービス固有のリレーションを持つデータ群を生成するエージェントを開発した．本エージェントは，LangGraph を用いたステートマシンとして設計されており，データベースを活用した記憶機構を実装している．これにより，過去に生成したデータと重複しない多様なコンセプトを立案し，そのコンセプトに基づいて相互に整合性の取れたユーザー，プロジェクト，作品群をひとまとまりの単位として, 本プラットフォーム上のデータ構造に特化した，自律的にデータを生成・投入する能力を持つデータ生成エージェントを開発した．

提案手法の有効性を検証するため，実験環境として一点物取引に特化した Web プラットフォームを開発し，google, Open AI の４つの基盤モデルに対して本エージェントを適用して 1000 件のデータ生成実験を行った．評価の結果，過去の生成履歴を参照する記憶機構を導入することで, Gemini 3 Flash にて生成される語彙数が 39.0% 増加し, GPT-5.2 においては全生成データのコサイン類似度分布が効果量 $d = -0.51$ で低下した．これらの結果は，記憶機構が生成内容の重複や特定のパターンへの収束を抑制し，生成されるデータセット全体の多様性を向上させることを示している．本研究は，LLM を用いた社

会的シナリオ生成が，開発初期段階においても実運用環境に近い整合性と多様性を持つテストデータの生成に有効であることを実証した．

Title
— Subtitle

Summary

The author shall describe his/her thesis in English text. 150 words.

Contents

1　序論

1.1　背景

1.1.1　クリエイターエコノミーの拡大と作品流通の変容

近年，インターネット技術の発展とソーシャルネットワーキングサービスの普及により，個人が自身の創作物を発信・販売し，収益を得るクリエイターエコノミーが急速に拡大している [1]．動画投稿，ライブ配信，ハンドメイド作品の販売，デジタルコンテンツの提供など，その形態は多岐にわたり，従来は企業や専門家が独占していた生産・流通・販売のプロセスが個人レベルまで民主化された．

特に，物理的な創作物を制作するクリエイターにとって，Mercari [2] や minne [3] といった CtoC マーケットプレイスの普及，あるいは BASE [4] のように個人が手軽に独自のオンラインストアを開設できるサービスの登場は，販路開拓のハードルを劇的に引き下げた．中でも，イラストや同人誌などのサブカルチャー領域に特化した BOOTH [5] は，物理的な作品とデジタルコンテンツを並列して扱うことができ，本研究が対象とするような「作品の文脈」を重視するクリエイター層から厚い支持を得ている．これにより，趣味の延長線上で制作を行っていた個人が，容易に売り手として市場に参加できる環境が整った．

しかし，これらの既存プラットフォームの多くは，既製品や大量生産品，あるいは不用品の流通に最適化されて設計されている側面が強い．商品自体は代替可能な商品として扱われ，その背後にある制作者の思想，制作過程の試行錯誤，作品が生まれた文脈は，商品説明欄の補助的なテキスト情報として扱われるに留まる．

一点ものや極めて少数の生産数しか持たない創作物において，その価値の源泉は物理的な素材や機能性よりも，この物語にこそ存在する．既存のプラットフォームでは，こうした作品固有の物語性や文脈を十分に表現・保存することが難しく，結果として作品が単なる商品として消費され，制作者のブランディングや持続的なファンコミュニティの形成に繋がりにくいという課題がある．

1.1.2　物理的な展示・即売会における制約とアーカイブの欠如

デジタルプラットフォームの対極として，学園祭，展示会，即売会といった物理的な場での作品発表も依然として重要な役割を果たしている．これらの場は，制作者と鑑賞者が直接交流し，作品の熱量を共有できる貴重な機会である．

しかし，物理的なイベントには一過性という避けられない制約が存在する．第一に，展示情報のアーカイブ化の問題である．展示会のために制作された解説パネルやキャプションボードは，イベント終了とともに廃棄または散逸することが多く，そこに記された詳細な作品情報はデジタル空間に蓄積されない．結果として，イベントに参加できなかった潜在的なファンには作品の魅力が届かず，制作者のポートフォリオとしても十分に機能しない場合が多い．

第二に，金銭授受と販売機会の損失である．学園祭や小規模な展示会では，販売機能を持たず展示のみとするケースや，販売する場合でも現金取引のみの対応となるケースが多い．また，展示されている作品そのものを購入したいという需要があっても，会期終了後の引き渡し手続きや配送手配の煩雑さから，その場での成約に至らないことも頻繁に発生する．このように，物理的な展示空間とデジタル上の決済・物流システムが分断されていることにより，多くの経済的機会が損失しているのが現状である．

1.1.3 一点もの取引における価格決定と信頼性の課題

一点もの，あるいは極めて供給量の少ない作品の流通においては，価格決定メカニズムと取引の信頼性に関して特有の課題が存在する．

まず，価格決定においては，需要が供給を大幅に上回る場合，定価販売形式では早い者勝ちとなり，転売目的の購入や，bot による自動購入などの問題を引き起こしやすい．これに対し，オークション形式は需要に応じた適正価格を発見する有効な手段であるが，既存のオークションサイトは，前述の通り不用品処分の文脈が強く，クリエイターが自身の作品を発表・販売する場としてのブランディング機能に欠ける場合がある．

1.1.4 共同体による創作活動と収益分配の複雑性

創作活動は必ずしも個人だけで完結するものではない．サークル活動，合同誌，アンソロジー企画，YouTuber 同士のコラボレーションなど，複数のクリエイターが関与するプロジェクト単位での活動も一般的である．しかし，既存の CtoC プラットフォームの多くは一人の売り手対一人の買い手という単純な取引モデルを前提としている．そのため，一つの作品またはイベントから得られた収益を，関与した複数のメンバーに分配する場合，代表者が一度全額を受け取り，その後銀行振込などで個別に再分配するという，極めてアナログで煩雑な事務作業が必要となる．これは，企画主催者の負担を増大させ，継続的な共同制作活動を阻害する要因となっている．

1.2 開発上の課題：複雑な文脈を持つシステムの検証難易度

前述したような作品の物語や複雑な権利・収益構造を持つプラットフォームを新規に開発する際，最大の障壁となるのが検証用データの不足である．

一般的な EC サイト（商品と価格のみの管理）であれば，ランダムな文字列やダミー画像を用いた機械的なデータ生成（Faker.js [6] 等）で十分に機能検証が可能である．しかし，本研究が対象とするような文脈を重視するシステムにおいては，以下の理由により，既存のテストデータ生成手法が通用しない．

1. **リレーションの整合性**: プロジェクト主催者と参加メンバー，そして出品された作品の間には，矛盾のない論理的な関係性が求められる．ランダム生成では，「部外者が他人のプロジェクトの作品を管理している」といった矛盾が頻発し，権限管理ロジックのテストが成立しない．
2. **検索・推薦アルゴリズムの検証**: 文脈に基づく検索システムを評価するには，データ自体に意味のあるテキスト（制作背景や動機）が含まれている必要がある．Lorem Ipsum のような無意味なテキストでは，検索精度の検証が不可能である．

したがって，開発者は「手動で時間をかけて整合性のあるデータを作る」か，「質の低いデータで妥協して検証を行う」かの二者択一を迫られており，これが開発効率と品質向上のボトルネックとなっていた．

1.3 本研究の目的

本研究の目的は，一点物売買プラットフォームの開発と，それを支える新たなテストデータ生成手法の確立の 2 点にある．

1.3.1 一点物作品のナラティブを保存するプラットフォームの開発

第一に，既存の E コマースでは埋没しがちな作品の文脈を構造化データとして保存・活用できる，新たな CtoC プラットフォームを構築する（第 4 章にて詳述）．これは本研究において，提案するデータ生成手法の有効性を検証するための実験環境としての役割も担う．

1.3.2 LLM を用いた社会的シナリオ生成によるシステム検証

第二の，そして本論文の主眼となる目的は，上記のような複雑なリレーションを持つ Web サービスの開発において，大規模言語モデル（LLM）を活用した社会的シナリオに基づくテストデータ生成手法を提案・実証することである．

本研究では，LLM に「架空のユーザーの人格」や「プロジェクトの動機」といった社会的背景（社会的シナリオ）を与え，自律的にデータベースへの登録を行わせるエージェントシステムを構築する．さらに，エージェントに記憶機構（RAG）を組み込むことで，既存データとの衝突を回避し，多様なシナリオを生成できるかを実験的に検証する．これにより，学習データが存在しない環境下においても，あたかも人間が実際に活動しているかのような，高密度で整合性の取れたテストデータを大量生成できることを示す．

2 関連研究

本章では，ソフトウェアテストデータの生成手法，大規模言語モデル（LLM）のデータ生成への応用に関する先行研究を概観し，本研究の位置づけを明確にする．

2.1 ソフトウェアテストデータの生成手法

Web アプリケーションの開発現場において，動作検証や負荷試験のためのテストデータ生成は不可欠な工程であり，主に構文的整合性やコード網羅率の確保に主眼が置かれてきた [7].

代表的なアプローチとして，Faker.js [6] や JavaFaker [8] に代表されるライブラリを用いたランダム生成手法が存在する．これらは，正規表現や定義済みの辞書に基づき，氏名，住所，メールアドレスといった形式が決まったデータを大量に生成することに長けている．また，Mockaroo [9] のようなデータ生成サービスでは，GUI 上でスキーマ定義を行うことで，ある程度の外部キー制約を満たした CSV や SQL データを生成可能である．さらに，近年の Web 開発フレームワーク（例：Prisma [10] などの ORM）には，開発初期のシードデータ（初期データ）を投入する機能が標準で備わっていることが多いが，これらも基本的には開発者が手動で作成した固定データか，あるいは前述の Faker 等で生成したランダムデータを投入するための機構に留まる．

しかし，これらの手法は，本来アプリケーションのログイン機能や権限管理といった基本機能の動作検証を目的としたものであり，データ間の複雑な意味的整合性や文脈を再現

することを主眼としていない．例えば，「居住地が日本であるにも関わらず住所がニューヨークに設定されている」といった整合性の欠如や，「ユーザー間のメッセージ交換に見られる時系列的な因果関係」を含む複雑なシナリオを再現することは，単純なランダムサンプリングでは原理的に困難である．本研究が対象とする CtoC プラットフォームのような，ユーザー間の相互作用や作品の背景ストーリーが重要となるシステムの検証において，これらの既存手法は不十分であると言える．

2.2 統計的・機械学習的なデータ合成

統計モデルや機械学習を用いて，既存データセットの分布を学習し，類似した特性を持つデータを新たに生成する手法も広く研究されている．代表的な手法として，条件付き GAN を用いた CTGAN [11] や，リレーショナルデータベース全体の生成に特化したベンチマークである SyntheRela [12] において評価されている SDV（Synthetic Data Vault）などが挙げられる．

これらの手法は，プライバシー保護（匿名化）やデータ拡張（アップサンプリング）において強力なツールとなるが，適用するには「学習元となる大量の既存データ」が存在することが前提となる．したがって，新規サービスの立ち上げ時のような「学習データが全く存在しない状態」においては適用することができない．本研究は，学習データが存在しないフェーズにおいて，LLM の知識と創造性を利用してゼロからデータを生成するアプローチをとるため，これらの統計的手法とは適用領域が根本的に異なる．

2.3 LLM を用いたリレーショナルデータの構築・操作

近年，LLM の高度な言語理解能力を活かし，複雑なスキーマ構造を持つリレーショナルデータを扱う研究が進展している．

Li ら [13] の OmniSQL は，Text-to-SQL モデルの学習用データ不足を解消するために，Web 上の表データから多様なデータベーススキーマと SQL クエリを自動合成するフレームワークを提案した．また，Sadia ら [14] の SQUiD は，非構造化テキストからリレーショナルデータベースを構築するタスク（Text2R）において，LLM に直接 SQL を書かせるのではなく，スキーマ設計とデータ投入のプロセスを分離することで整合性を高める手法を示した．さらに，Kim ら [15] の ReFuGe は，予測タスクの精度向上のために，LLM エージェントを用いてリレーショナルデータベースから有効な特徴量を自動生成す

る枠組みを提案している．

これらの研究は，LLM が複雑なリレーション構造を理解・操作できることを示しているが，その主目的はあくまで「学習データの作成」や「既存情報の抽出・分析」にある．本研究は，これらの研究と技術基盤（LLM エージェント，プロセス分解アプローチ）を共有しつつ，目的を「架空の社会的シナリオに基づく新規データの創造」に置く点で独自性がある．

2.4 社会シミュレーションと生成的エージェント

LLM を用いて人間らしい振る舞いをシミュレートする研究として，Park ら [16] の Generative Agents が挙げられる．彼らは，記憶と計画を持つ複数のエージェントを仮想空間内で生活させ，エージェント同士の創発的な社会的相互作用（例：パーティーの企画）が生まれることを実証した．

この「記憶に基づく一貫した行動生成」という概念は本研究の基礎となっているが，Generative Agents の出力は自然言語のログ（非構造化データ）であり，そのままではリレーショナルデータベースの厳密な制約（外部キーやデータ型）を満たすテストデータとして利用できない．本研究の貢献は，Generative Agents のような「人間らしい文脈や動機」を維持しつつ，それを SQUiD のような「構造的整合性」を持つリレーショナルデータとして出力するためのシステムアーキテクチャを提案する点にある．

3 提案手法：社会的シナリオに基づくデータ生成エージェント

本章では，一点物売買プラットフォームのような複雑なリレーション構造を持つシステムに対し，整合性と多様性を両立したテストデータを自律的に生成するエージェントシステムについて述べる．

3.1 アプローチの概要

従来，ウェブサービスの開発におけるテストデータ生成には，Faker.js 等のライブラリを用いたランダム生成や，固定のダミーデータセットの流用が一般的であった．しかし，これらの手法では「ユーザー A がプロジェクト B を立ち上げ，その文脈に沿った作品 C を制作し，ユーザー D がそれを購入する」といった，エンティティ間の意味的な整合性（社会的シナリオ）を再現することは困難である．本研究では，大規模言語モデル（LLM）

を中核に据え，データの生成プロセスを「脚本生成」「演出」「設計」という映画制作の工程に着想を得た段階的な生成プロセスに分解することで，この課題を解決する．提案するエージェントは，単に単一のテーブルデータを埋めるのではなく，「一貫した社会的文脈」を生成単位とする．これにより，生成されたデータ群はデータベースのリレーション制約（外部キー制約）を満たすだけでなく，人間が読んでも不自然さを感じない「物語」としての整合性を保持する．

3.2 システムアーキテクチャ

本エージェントは，LangGraph [17] を用いたステートマシンとして実装されている．LangGraph は，LLM エージェントの制御フローをグラフ構造（ノードとエッジ）として定義するフレームワークであり，Yao ら [18] が提案した ReAct パターンなどの複雑な推論プロセスを容易に実装可能にする．本研究では，各ノードを特定の役割（Director Agent, Scenarist Agent, Designer Agent, Saver Agent）を持つ専門家として定義し，共有された状態を更新しながらパイプライン処理を行うアーキテクチャを採用した．図 1 にシステム全体の構成を示す．

3.2.1 ステートマシンの構成

エージェントの処理フローは以下の 4 つの主要ノードによって構成される．

1. **Director Agent**: 生成プロセス全体の指揮を執り，データの分布（カテゴリバランス）を管理する．過去の生成ログを参照し，現在不足しているカテゴリや構造タイプを特定する．本ノードは LLM を使用せず，あらかじめ定義された確率モデル（Distribution Matrix）に基づき，次に生成すべき「枠組み（例：ガジェットカテゴリ・組織型構造）」を決定し，下流の Designer Agent へ指示を発行する．
2. **Designer Agent**: Director Agent から渡された抽象的な枠組みに基づき，具体的な「シナリオのコンセプト」を立案する．この段階で記憶機構（RAG）を活用し，過去に生成された類似シナリオと重複しないよう，ユニークなテーマやトーン（例：「深夜ラジオのリスナー交流」や「退廃的なサイバーパンク」）を決定する．コードリスト 1 に，本エージェントのシステムプロンプトを示す．

```
1        あなたはテストデータ生成のための「構成作家」です。
2        指定された「カテゴリ」と「構造タイプ」に基づいて、具体的でユニークなシナリオのコンセプトを立案してください。
3
```

```
    ## プラットフォームの定義
    このプラットフォームは、現実世界のユーザーが、自分の制作した作品（模型、手芸、イラストなど）やコレクションを展示・売買するためのサービス（）です。CtoCArtSquare
    作中の登場人物や兵器そのものを生成するのではなく、「それを趣味として楽しむ人々」のシナリオを作成してください。

    ## 指示
    - 既存のシナリオとは異なる視点、設定、トーンで企画してください。
    - ニッチな設定や、エッジの効いた設定を歓迎します。

```

Listing 1　Designer Agent（コンセプト立案）のシステムプロンプト

3. **Scenarist Agent**: Designer Agent が作成したコンセプト指示書に基づき，具体的なデータ（User, Project, Item）を生成する．最新の LLM が持つ Structured Output 機能を利用し，自然言語による物語の記述と，データベーススキーマ（JSON Schema）へのマッピングを同時に実行する．これにより，文脈的な整合性と技術的な整合性（外部キー制約等）を同時に満たすデータを生成する．コードリスト 2 に，本エージェントのシステムプロンプトを示す．

```
    あなたは架空のクリエイタープラットフォーム(ArtSquare)のデータを生成する「脚本家」です。
    与えられた「カテゴリ」と「社会構造(Structure Type)」に基づいて、
    整合性の取れた一つの小さな社会（シナリオクラスター）を生成してください。

    ## 重要な前提
    - はの創作・ホビー・コレクションのプラットフォームです。ArtSquareCtoC
    - 生成されるは、現実的に個人が売買・展示できるもの（模型、手芸、同人誌、素材、データなど）に限られます。Item

    ## 制約事項
    - User, Project, Item の各データを作成すること。
    - tempId を使用して、生成データ内で矛盾のないリレーションを構築すること。
    - は形式で、読み応えのある記事のように書くこと。DescriptionMarkdown
    - や説明文の口調は、設定された「ペルソナ」や「カテゴリの文化」に合わせること。Bio

```

Listing 2　Scenarist Agent（データ実装）のシステムプロンプト

4. **Saver Agent**: 生成された構造化データをデータベースに保存すると同時に，その内容をベクトル化（Embedding）し，エージェントの長期記憶に格納する．これにより，次回の Designer Agent の推論時に参照可能な状態とする．

3.3　社会的シナリオのデータモデル

本エージェントが生成するデータの最小単位である「シナリオクラスター」は，以下の 3 層構造を持つ．この構造は，実験対象とするプラットフォーム（第 4 章で詳述）のデー

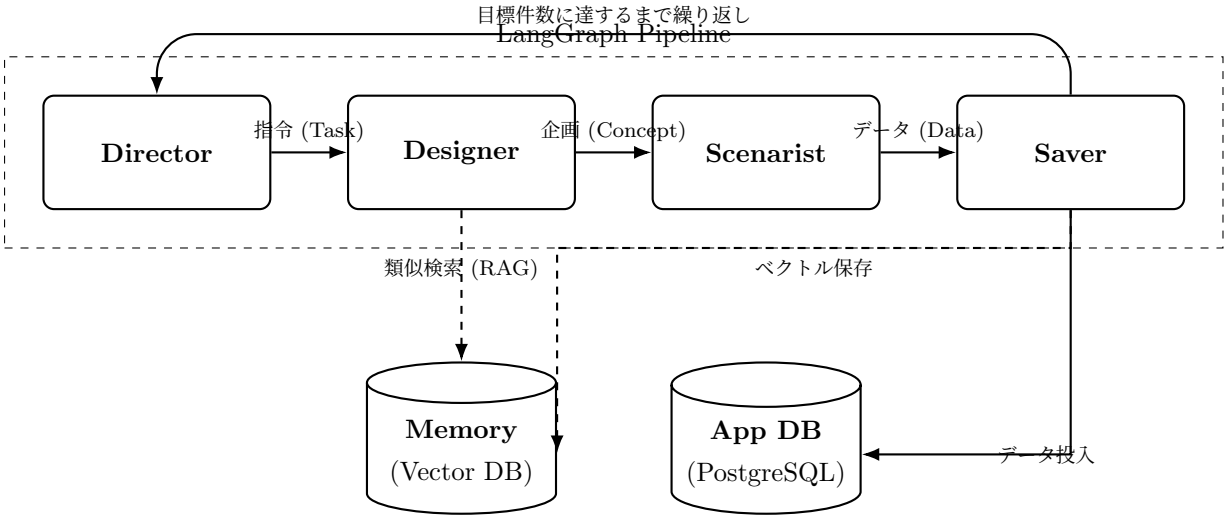

Figure1　社会的シナリオ生成エージェントのシステムアーキテクチャ

タモデルと 1 対 1 で対応している．

3.3.1　Concept

すべてのデータの起点となる「テーマ」である．ここには，活動のジャンル（例：Handcraft, Gundam），動機（例：リサイクル, 教育, 偏愛），および雰囲気（例：Cyberpunk, Nostalgic）が含まれる．RAG による重複排除は，主にこの Concept レベルで行われる．

3.3.2　Community

Concept を実現するために集まった人々の定義である．

- **Organizer**: プロジェクトの主催者．強い動機とリーダーシップを持つ．
- **Collaborator**: 協力者．技術提供や資材提供など，主催者を補完する役割を持つ．

LLM は，これらの登場人物に対し，性格特性や背景ストーリーを付与し，User プロフィールの自己紹介文などを生成する．

3.3.3　Output

コミュニティ活動の結果として生み出された「作品」である．各作品には，以下の要素が付与される．

- **Narrative**: その作品がなぜ作られたかという背景．
- **Spec**: 素材，サイズ，重量などの物理的特性．

- **Log**: 制作過程の記録や，使用されたツールの履歴.

3.4 記憶機構と RAG

本研究の核となる技術的特徴は，エージェントが「過去に自分が何を作ったか」を記憶し，それを参照しながら次の生成を行う点にある．これにより，生成数が増えても「ネタ被り」を防ぎ，常に新しいパターンを模索し続けることが可能となる.

3.4.1 ベクトル検索による重複検知

記憶機構の実装には，軽量かつ高速なベクトル検索ライブラリである `sqlite-vec` を採用した．エージェントが新たなシナリオを生成する際，以下のプロセスを実行する.

1. **Director Agent**: 次に生成すべきカテゴリと構造を決定し，タスクを発行する.
2. **Designer Agent**: タスクに基づき，ベクトルデータベースから類似する過去のシナリオ（上位 N 件）を検索する.
3. 検索された既存シナリオを Prompt に含め，**「これらとは異なる，新規の観点を持つシナリオを作成せよ」**という制約のもと，新しいコンセプトを立案する.
4. **Scenarist Agent**: 確定したコンセプトに基づき，詳細なデータを実装する.

この RAG ループにより，エージェントは自己複製的な生成を回避し，探索空間を強制的に広げることが可能となる．実験結果（第 5 章）で示される通り，この機構は語彙数やアイデアの多様性に劇的な向上をもたらした.

3.5 プロンプトエンジニアリング

LLM に対して「リアルなデータ」を生成させるために，以下のプロンプト戦略を採用した.

3.5.1 Few-Shot Prompting with Anti-Patterns

良質な例だけでなく，避けるべき例を明示的に与えた．特に，「LLM が生成する傾向にある典型的なパターン」を禁止事項として列挙した.

- 禁止用語：「革新的な」「ソリューション」「多様なニーズ」「シームレスな」等の一般的なマーケティング用語.

- 推奨スタイル：個人の嗜好や生活様式に焦点を当てた記述.

3.5.2 Persona Adoption

各ノードにおいて，LLM に明確なペルソナを与えることで，出力のトーンを制御した．例えば Designer Agent には，「熟練したデータベース管理者であり，かつ特定のサブカルチャーに精通した言語感覚を持つ」といった複合的なペルソナを設定し，技術的な整合性と人間味のあるテキストの両立を図った．

4 実験環境：一点物売買プラットフォーム

本研究では，提案するデータ生成エージェントの有効性を検証するための実験環境として，一点物の取引に特化した Web プラットフォームを実際に開発した．本章では，このプラットフォームの概要，技術構成，およびテストデータに求められる要件について述べる．

4.1 プラットフォームの概要

開発したプラットフォームは，クリエイターが制作した「作品」を制作背景とともに販売・購入できる CtoC マーケットプレイスである．Amazon や Mercari 等の代表的なサービスが，商品をスペックと価格で比較される代替可能なモノとして扱うのに対し，本プラットフォームは，作品を代替不可能なアート作品として扱うことを主眼としている．

主な特徴は以下の通りである．

- **プロジェクトベースの活動**: ユーザーは単独で出品するだけでなく，プロジェクトを立ち上げ，複数人で協力して制作・販売を行うことができる．
- **制作ログの重視**: 作品の完成形だけでなく，使用した素材，制作中の失敗，試行錯誤の履歴をログとして記録し，作品価値の一部として提示する．
- **複雑な権利・収益分配**: 一つの作品に対し，素材提供者，制作者，プロジェクト主催者など複数のステークホルダーが存在し，売上は Stripe Connect を通じて自動的に分配される．

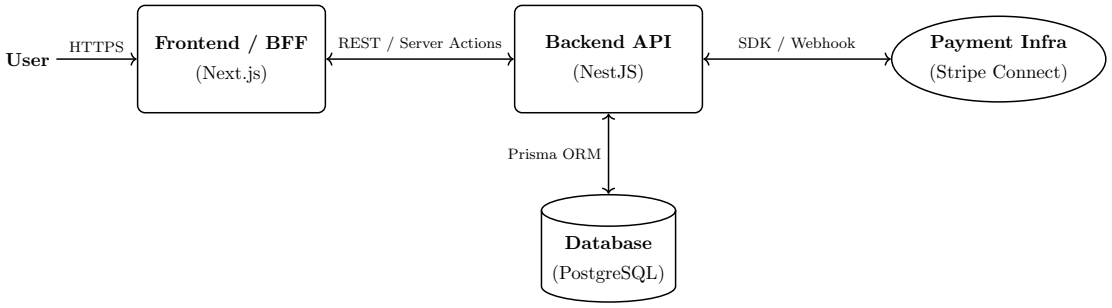

Figure2　Payment Platform 2 のシステム構成

4.2　システム構成

本プラットフォームは，拡張性と保守性を重視し，モダンな Web 技術スタックを用いて構築された（図 2）.

4.2.1　フロントエンド

ユーザーとの接点となるフロントエンドには，React ベースのフレームワークである Next.js (App Router) [19] を採用した．本システムでは，単なる画面描画だけでなく，サーバーサイドでのデータフェッチや認証セッション管理を行う BFF としての役割も担わせている.

■ディレクトリベース・ルーティングの採用　ソースコード構成（src/app 配下）において，ファイルシステムの階層構造をそのまま URL パスにマッピングするディレクトリベース・ルーティングを採用した．例えば，クリエイターのポートフォリオページは src/app/[userPubId]/page.tsx，作品詳細ページは src/app/[userPubId]/[itemPubId]/page.tsx として配置されている．このように動的パラメータをディレクトリ名に含めることで，https://domain.com/user123/item456 のような，直感的で可読性の高い URL 構造を容易に実現している.

■Server Actions とデータ取得の最適化　コンポーネント設計においては，React Server Components (RSC) を全面的に採用した．作品情報（Markdown テキスト）やユーザープロファイルなどの静的な初期データは，サーバーサイドでデータベース（またはバックエンド API）から直接取得し，HTML としてクライアントに送信される．また，フォーム送信や入札アクションなどの動的な操作には Server Actions (src/actions/) を用いることで，クライアントサイドに露出する API エンドポイントを隠蔽しつつ，型安全な関数呼

び出しとしてバックエンドとの通信を実装している．

■ 認証と認可　認証基盤には Node.js のライブラリである NextAuth.js (src/api/auth/[...nextauth]) を導入し，Google OAuth およびメールアドレス認証によるセキュアなログインフローを構築した．セッション情報は JWT（JSON Web Token）として暗号化され，HTTP Only Cookie を用いて管理される．

4.2.2 バックエンド

システムのコアロジックを担当するバックエンドには，Node.js フレームワークである NestJS [20] を採用した．このフレームワークは、複雑な CtoC 取引のステート管理や，金銭に関わる厳密なバリデーションを行うため，TypeScript による静的型付けと，依存性の注入（DI）によるモジュラーアーキテクチャを徹底している．

■モジュール分割による関心の分離　アプリケーションは機能単位でモジュールに分割されている．

User Module　アカウント管理，プロフィール更新
Auth Module　認証処理
Item Module　作品の CRUD，Markdown 解析，在庫管理
Order Module　注文作成，決済ステータスの遷移管理
Project Module　企画のガバナンス管理，収益分配計算

このようにドメインごとに責務を分離することで，コードの可読性を高めるとともに，特定の機能（例：オークション処理）に変更が生じた際の影響範囲を最小限に抑えている．

■非同期イベント処理　本システムでは決済処理を Stripe に委譲しているため，決済の成功・失敗や，オークションにおける与信枠の確保の結果は，すべて非同期の Webhook イベントとして通知される．これを受け取るための専用のエンドポイント (src/webhook) を実装し，Stripe からの署名検証を行った上で，Order データベースのステータスを安全に更新する仕組みを構築した．これにより，ネットワーク遅延や一時的な障害が発生しても，最終的なデータの整合性が保証される．

4.2.3 データベース

データの永続化には，リレーショナルデータベースである PostgreSQL [21] を採用した．複雑な多対多のリレーション（例：Project と Collaborators，Item と Bids）を矛盾なく管理し，ACID 特性（原子性，一貫性，独立性，永続性）を備えたトランザクション処理を実現するためである．

■Prisma ORM による型安全なデータアクセス　アプリケーション層とデータベース層の架け橋として Prisma [10] を使用している．schema.prisma ファイルに定義されたデータモデルに基づき，型定義ファイル（TypeScript 型）が自動生成される．これにより，バックエンドの開発において「存在しないカラムへのアクセス」や「型の不一致」といった単純なミスをコンパイル時に検出可能となり，開発効率と品質が大幅に向上した．

■スキーマ駆動開発　開発フローにおいては，まず schema.prisma でデータ構造（User, Item, Project 等のモデル定義）を記述し，それを元にマイグレーション（prisma migrate）を実行して DB スキーマを更新する手法をとった．これにより，仕様書に定義されたデータモデルが，実装コードとデータベースの実体に乖離することなく，常に同期された状態を維持している．

4.2.4 決済および送金基盤 (Stripe Connect)

CtoC プラットフォームにおける金銭取引の要となる決済インフラには，Stripe Connect [22] を採用した．本システムでは単なる決済代行だけでなく，売り手（クリエイター）への送金管理，本人確認，および複雑なオークション決済を実現するために，以下の高度な機能群を実装している．

■連結アカウントによるユーザー管理　プラットフォームに参加するクリエイター（売り手）は，Stripe の「連結アカウント」として管理される．User モデルの stripeAccountId カラムには，Stripe 側で発行された一意のアカウント ID（acct_...）が保存される．これにより，売上の入金先銀行口座の管理や，特定商取引法に基づく本人確認手続きといったセンシティブな業務を Stripe の堅牢なインフラにオフロードしつつ，プラットフォームとしてのコンプライアンス遵守を実現している．

■支払いと送金の分離　資金フローにおいては，買い手からの支払いを即時に売り手へ送金するのではなく，「支払いと送金別方式」を採用した．この方式では，買い手の決済は一

度プラットフォームのアカウントに対して行われ，その後，任意のタイミングで売り手の連結アカウントへ送金が実行される．これにより，以下の要件を満たしている．

エスクロー機能　商品の発送完了や受け取り評価が行われるまで，プラットフォーム上で売上を一時的に保持し，トラブル時の返金を容易にする．

収益分配の柔軟性　プロジェクト主催者への手数料やプラットフォーム手数料を差し引いた上で，正確な金額をクリエイターに分配する．

■オークションにおける与信枠の活用　イングリッシュオークションや抽選販売において，最大の課題である「当選後の未払い」を防ぐため，クレジットカードの「オーソリゼーション」機能を活用した決済フローを構築した．

与信枠確保　入札や抽選応募の時点で，PaymentIntent を作成し，capture_method: 'manual' オプションを用いてカードの与信枠のみを確保する．この段階では実際の請求は発生しない．

確定と解放　落札や当選が確定した時点で capture を実行し，売上を確定させる．一方，落選や高値更新が発生した場合は即座に cancel を実行し，与信枠を解放する．

この仕組みにより，ユーザーの資金拘束を必要最小限に抑えつつ，システム側は「支払い能力のある入札」のみを受け付けることが可能となり，取引の安全性を飛躍的に高めている．

4.3　データモデル設計

本システムのデータモデルは，User，Item，Project の 3 つのコアエンティティを中心に設計されている．主要なエンティティの関係性を figure 1 に示す．

4.3.1　User モデル

User モデルは，本プラットフォームにおけるすべての活動の基点となる「主体」を表すリソースである．単なる認証用のアカウント情報とは異なり，インターネット上におけるクリエイターとしてのアイデンティティと，経済活動を行うための法的人格としての側面を併せ持つ．

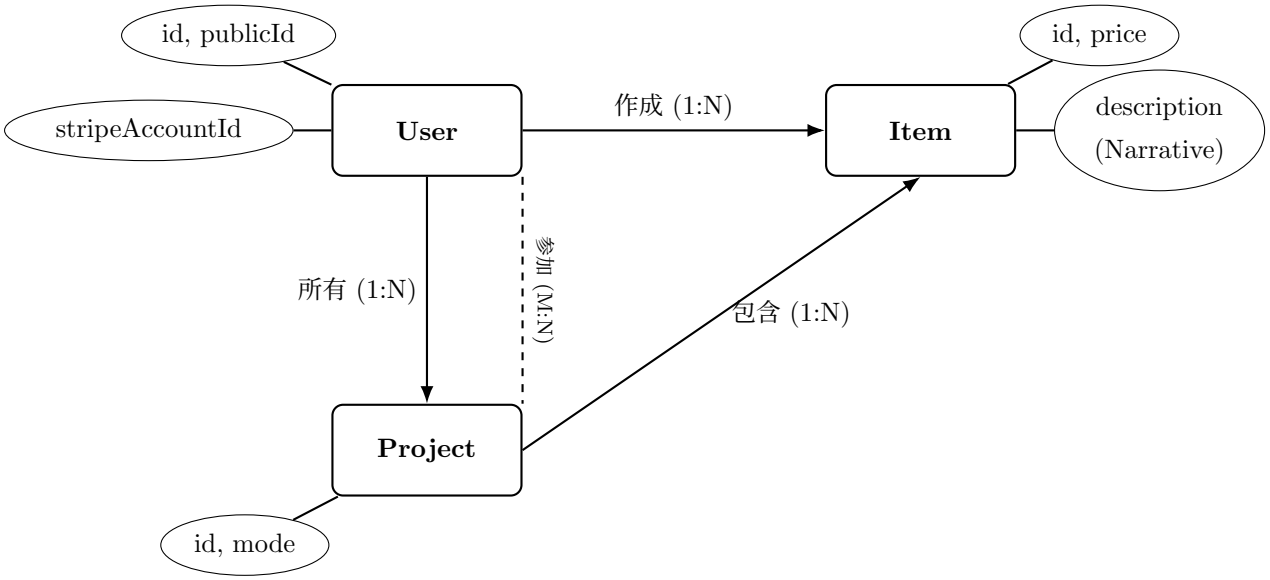

Figure3　User, Item, Project の ER 図（主要なリレーションのみ抜粋）

■公開識別子とポートフォリオ機能　各ユーザーには，データベース内部の主キー（id）とは別に，外部公開用の識別子として publicId（CUID）が付与される．これにより，https://domain/[userPubId] という予測不可能な固有 URL が生成される．この URL は，ユーザーの出品作品や参加プロジェクト，活動履歴を集約したポートフォリオページとして機能し，クリエイターのブランディングを支援する．

■経済的主体としての属性　CtoC 取引における売り手としての機能を果たすため，Stripe Connect のアカウント ID を stripeAccountId カラムに保持する．これは，本人確認が完了したユーザーにのみ発行される ID であり，このフィールドの有無によって，システムはユーザーが「出品可能」か「購入専用」かを判別する．また，買い手としての情報は stripeCustomerId に紐づけられ，クレジットカード情報等の機微な決済情報はすべて Stripe 側の Vault に保存される設計としている．

■多層的なリレーション　User は他のリソースと多層的な関係を持つ．自身が作成した作品（items）だけでなく，他者と共同制作した作品（contributingItems）や，主催する企画（ownedProjects），参加者として関わる企画（contributingProjects）へのリレーションを保持し，個人の活動の広がりをグラフ構造として表現する．

4.3.2 Item モデル

Item モデルは，取引の対象となる「作品」を管理するリソースである．既存の EC サイトにおける「商品」モデルと決定的に異なる点は，物理的なスペック情報よりも「ナラティブ」の記述と保存に重きを置いている点である．

■ナラティブの保存　作品の背景ストーリーや制作過程を記録するために，description フィールドには Markdown 形式のテキストデータが保存される．フロントエンド側ではこれがリッチテキストとしてレンダリングされ，画像や動画の埋め込みとともに，作品単体で完結した「記事」として表示される．これにより，物理的な展示会場において QR コード経由でアクセスされた際，鑑賞者に深い理解を促すキャプションボードとしての役割を果たす．

■販売ロジックの抽象化　一点物や限定品など，作品の性質に応じた最適な販売形態を選択可能にするため，salesMethod カラム（ENUM 型）を定義している．

AUCTION　イングリッシュオークション形式．currentPrice や endTime と連動し，入札を受け付ける．
LOTTERY　抽選販売形式．事前与信を用いた応募ロジックが適用される．
FIXED_PRICE　定額販売形式．即時購入が可能となる．

■ステータス管理　作品のライフサイクルを管理するために，status カラム（ENUM 型）を用いる．出品前の DRAFT，販売中の FOR_SALE，取引成立後の SOLD などの状態遷移を厳密に定義することで，不正な購入や二重決済を防ぐ排他制御を実現している．また，Stripe の商品マスタと同期するために stripeProductId および stripePriceId を保持し，決済システムとの整合性を保っている．

4.3.3 Project モデル

Project モデルは，複数の User と Item を束ねる「共同体」および「文脈」を提供するリソースである．これは単なるカテゴリ分類タグではなく，独自の経済圏を持つ「企画」そのものをデジタル上に実体化させたものである．

■ガバナンス構造の表現

Owner　プロジェクトの作成者であり，管理者権限を持つ．ownerId 外部キーによって User モデルと一対多で結ばれる．

Collaborators　企画に参加するクリエイター群．中間テーブルを介した多対多のリレーションによって定義される．この構造により，招待制のサークル活動や，主催者が権限を持つコンテストなど，多様な組織形態をシステム上で再現可能にしている．

■経済圏としての機能　Project は，紐づけられた作品群（items）の集合体として機能する．Item モデルは projectId を外部キーとして持ち，自身がどの文脈に属しているかを示す．このリレーションに基づき，システムは「企画単位での売上集計」や「主催者への手数料分配」といった高度な金銭処理を実行する．すなわち，Project モデルは単なるグルーピング機能ではなく，収益分配の計算単位としても機能する設計となっている．

4.4　Project 管理方式

本プラットフォームにおける Project は，単なる商品カテゴリや検索タグではなく，独自の規律と経済ルールを持つ「共同体」として定義される．現実世界の創作活動には，大学のサークル活動のような水平的な人間関係に基づくものから，コンテストのように主催者が強い権限を持つ垂直的な構造，あるいはインターネット上のミームのように不特定多数が自律的に参加するものまで，多種多様な形態が存在する．これらを単一のシステム仕様でカバーすることは困難であるため，本システムでは Project リソースに対して 3 つの異なるガバナンスモード（管理方式）を実装した．

4.4.1　Cooperative Mode

■意義と適用領域　Cooperative Mode は，既知のメンバー間の「信頼」を基盤とした小規模から中規模の共同体を再現するためのモードである．大学のサークル活動，グループ展，あるいは特定のクリエイター同士のコラボレーション企画などを想定している．ここでは，厳格な管理コストを削減し，メンバー間の自律的な調整を優先する運用がなされる．

■仕様詳細

参加権限　「招待制」を採用する．プロジェクトオーナーまたは既存メンバーからの招待リンク（署名付き URL）を受け取ったユーザーのみが，Collaborator として参加で

きる．

作品管理　参加メンバーは，自身の判断で作品（Item）を Project に紐づけ，即座に公開することができる．オーナーによる事前承認プロセスは存在しない．

収益構造　原則として，売上は各作品の制作者に直接帰属する．オーナーによる「場所代」の強制徴収機能は無効化されており，金銭的な搾取が発生しない構造となる。

4.4.2　Managed Mode

■意義と適用領域　Managed Mode は，明確な主催者が存在し，品質管理やレギュレーションの遵守が求められる階層型の共同体である．企業主催のコンテスト，有名インフルエンサーによる企画展，アンソロジーの編纂などを想定している．このモードの核心は，主催者に対して「管理権限」と引き換えに「経済的インセンティブ」を与える点にある．これにより，従来はボランティアに依存していた企画運営業務（キュレーション，広報，進行管理）を，収益を生む事業として成立させることを可能にする．

■仕様詳細

参加権限　「公募制」を採用する．ユーザーは Project に対して参加申請を行い，オーナーが管理画面でこれを「承認」または「拒否」することでメンバーシップが確定する．

作品管理　出品された作品は一時的に PENDING ステータスとなり，オーナーが内容を確認し承認するまで公開されない．また，オーナーは規約違反の作品を強制的にプロジェクトから除外する権限を持つ．

収益構造　Stripe Connect の ”Separate Charges and Transfers” API を活用し，「主催者手数料」 の設定を可能にしている．取引成立時，システムは売上総額からプラットフォーム手数料と主催者手数料を自動的に控除し，残額を作品制作者へ送金する．

4.4.3　Theme Mode

■意義と適用領域　Theme Mode は，中央集権的な管理者を排し，共通の文脈やテーマの下に不特定多数の主体が自律的に集合する「創発的な共同体」である．インターネット上で突発的に発生するトレンドや，特定のハッシュタグを介して形成される大規模な社会現象，あるいは季節的なイベントに伴う集合知的な制作活動をモデル化している．本モードの設計目的は，参加における許認可プロセスを完全に撤廃することで参入障壁を最小

化し，ネットワーク外部性を最大限に作用させることにある．これにより，短期間で幾何級数的な作品数の増加とトラフィックの凝集を実現する．

■仕様詳細

参加権限　「自由参加」を採用する．アカウントを持つ全ユーザーは，許可を得ることなく即座に Project に参加できる．

作品管理　紐づけられた作品は即時公開される．オーナー（発起人）は，通報対応を除き，他者の作品を削除する権限を持たない．

収益構造　主催者手数料の設定は不可である．売上はすべて作品制作者に還元される．これにより，発起人が「他人の作品で金儲けをしている」という批判を避けることができ，純粋なムーブメントとしての成長を促進する．

4.4.4　Project 管理方式の比較

各モードにおける権限と金銭的フローの差異を示す。

Table1　インダストリアルアートの授業

方式名	参加フロー	作品公開フロー	主催者権限	収益分配 (Organizer Fee)
Cooperative	招待のみ	即時公開	メンバー除名可	不可
Managed	申請 → 承認	投稿 → 承認	作品削除・否認可	可
Theme	自由参加	即時公開	管理不可	不可

4.5　統合型検索システム

本プラットフォームにおける検索機能は，単に特定の物品を発見するためのツールではなく，ユーザーを未知の作品やクリエイター，あるいはコミュニティへと導く「ディスカバリー」の中核機能として位置づけられる．ユーザーの探索意図は，「特定の作品が欲しい」だけでなく，「面白い企画はないか」や「特定の作家の活動を知りたい」など多岐にわたる．これら多様な検索意図を単一のインターフェースで充足させるため，User，Item，Project の 3 つの主要リソースを横断的に検索する「統合型オムニサーチシステム」を実装した．

4.5.1 設計思想: 探索的検索への対応

従来のEコマースサイトにおける検索システムは，SKU（最小管理単位）ベースの商品検索に特化しており，ユーザーが明確な購買目的を持っていることを前提とすることが多い．しかし，一点物や創作活動を扱う本プラットフォームにおいては，ユーザーは必ずしも具体的な商品名を検索するとは限らない．「折り紙」「ガレージキット」といったジャンル名や，「学園祭」といったイベント名から検索を開始し，そこから関連するクリエイターや作品群へと興味を広げていく「探索的検索」の行動様式が支配的である．したがって，検索システムは，単一のリソースリストを返すのではなく，システム内の多様なエンティティ（主体，文脈，作品）を網羅的に提示し，ユーザーの回遊を促す情報アーキテクチャを持つ必要がある．

4.5.2 アーキテクチャ: 並列クエリ実行モデル

オムニサーチのバックエンド処理においては，異なるデータ構造を持つ複数のテーブルに対する検索を効率的に処理するため，並列クエリ実行モデルを採用した．具体的には，ユーザーが入力した単一の検索クエリに対し，バックエンド（NestJS）は以下の処理を非同期かつ並列に実行する．

Project Search　プロジェクト名および概要文に対する全文検索またはベクトル検索．
User Search　ユーザー名およびプロフィール文に対する検索．
Item Search　作品名，説明文（Markdown），タグに対する検索．

これらのクエリは Promise.all 等を用いて並行してデータベース（PostgreSQL）に発行され，個別の検索結果が得られ次第，単一のレスポンスオブジェクトに統合されてクライアントへ返却される．これにより，リソースごとに個別の検索画面を遷移する必要をなくし，シームレスな検索体験を実現している．

4.5.3 インターフェース: 階層的情報の統合表示

検索結果の表示画面においては，情報の重要度とユーザーの認知負荷を考慮し，リソースタイプごとに優先順位を設けた階層的表示を採用した．

上部: 関連プロジェクト　検索結果の最上部には，クエリに関連性の高い Project を最大3件表示する．これは，個別の作品よりも「企画」や「イベント」という大きな文脈

を優先して提示することで，ユーザーをコミュニティ全体へと誘導するためである．

中部: **関連クリエイター**　次に，関連する User を最大 5 件表示する．これにより，特定の作家を探しているユーザーのナビゲーション意図に即座に応える．

下部: **関連作品**　最下部には，関連する Item をグリッド形式で表示する．ここでは無限スクロールを採用し，ユーザーが満足するまで作品を閲覧し続けられるよう設計している．

また，検索結果画面のサイドバーには，「ファセットナビゲーション」を配置した．ユーザーは，統合検索によって得られた広範な結果から，価格帯，カテゴリ，販売方式（オークション・定額など）といった属性に基づいて動的に絞り込みを行うことができる．これにより，探索的な「広げる検索」と，条件を特定する「絞る検索」の両立を図っている．

4.6　販売形態と価格決定アルゴリズム

本プラットフォームが扱う「一点物」や「希少在庫」の流通において，固定価格での販売は必ずしも最適解ではない．需要が供給を大幅に上回る場合，価格メカニズムが機能せず「早い者勝ち」となることで，適正価格の販売ではなくなるからである．この課題を解決し，作品を最も必要とするユーザーへ適正価格で配分するために，本システムでは以下の 4 つの販売アルゴリズムを実装した．また，全ての方式において，Stripe Connect API を用いたクレジットカードの与信枠確保を必須とすることで，CtoC 取引における最大の問題である支払い不履行を低減している。

4.6.1　イングリッシュ・オークション

在庫が単一（$N = 1$）である一点物の作品に対し，最も高い評価額を持つ購入希望者を決定するための標準的な競り上げ方式である．

■自動入札アルゴリズム　ユーザーの利便性とサーバー負荷の軽減を考慮し，Vickrey Auction の概念を取り入れた自動入札システムを採用した．ユーザーは「現在価格」ではなく，自身の「支払ってもよい最高額」を入力する．システムは，2 番目に高い入札額（または開始価格）に最小入札単位を加えた額を「現在価格」として自動的に算出・更新する．これにより，ユーザーは画面に張り付いて再入札を繰り返す必要がなくなり，合理的な入札行動が促進される．

■連続的な与信枠管理フロー　本システムでは，「最高入札者は常に支払い能力を保証されている」という状態を維持するため，最高入札者が入れ替わるたびに決済リソースの「ハンドオーバー」を行う．具体的なトランザクションフローは以下の通りである．

1. **新規与信**　新たな高値入札者が現れた際，システムはそのユーザーのクレジットカードに対し，入札額全額の与信枠確保（paymentIntents.create with capture_method: 'manual'）を試行する．これに失敗した場合，入札自体を却下する．
2. DB **更新**　与信確保に成功した場合のみ，データベース上の Item レコード（現在価格，リーダー ID）および Bid レコードを更新する．
3. **旧与信解放**　更新完了後，直前の最高入札者の与信枠（PaymentIntent）に対してキャンセル処理（paymentIntents.cancel）を非同期的に実行し，枠を即時解放する．

このプロセスにより，オークション終了時に「落札者が支払えない」というリスクを回避している。

4.6.2 複数点式オークション

在庫が複数（$N > 1$）存在するが，需要がそれを上回る場合（例：限定 5 個のガレージキット）に適用される，本システム独自の販売方式である．従来の単一価格オークションではなく，入札額がそのまま支払額となる Pay-as-Bid 方式 を採用した．

■当選者決定ロジック　入札額の高い順にソートを行い，上位 N 名を当選者として確定する．

$$B_1 \geq B_2 \geq \ldots \geq B_N \geq \ldots \geq B_M$$

ここで B_i は i 番目のユーザーの入札額であり，上位 N 名までの入札が有効となる．同額入札が存在する場合は，先着優先として順位付けを行う．

■Pay-as-Bid による収益最大化　当選者全員が B_N（ボーダーライン価格）を支払う単一価格方式ではなく，各当選者が自身の提示した B_i を支払う Pay-as-Bid 方式を採用した．これにより，作品に対し強い選好を持つ熱狂的なファン（B_1）からは高い収益を，学生などの価格に敏感な層（B_N）からは適正な収益を得ることが可能となる．結果として，同一の商品であっても購入者の支払意思額に応じた価格差別化が自然発生し，クリエイターへの還元額を最大化する．

■取引および決済フローの詳細　本方式では，複数の当選候補者が並存するため，イングリッシュ・オークションのような「入れ替え」ではなく「全件確保」に近い戦略をとる．

Step 1: **入札と与信確保**　ユーザーが入札を行う際，システムは即座に入札額全額の与信枠確保を実行する．この時点で決済は確定しないが，カードの利用枠は押さえられる．これにより，冷やかしや入札後の支払い拒否を未然に防ぐ．

Step 2: **ボーダーラインの可視化と競争**　オークション期間中，入札画面には「現在，上位 N 位に入るための最低価格（B_N）」がリアルタイムで表示される．ユーザーはこのボーダーラインを参考に，自身の予算内で入札額を調整する．

Step 3: **決着と一括決済**　決着と一括決済 (Settlement) 終了時刻（endTime）に到達した時点で，システムは全入札を確定させる．上位 N 名: 確保していた与信枠に対し capture を実行し，売上を確定させる．落選者 ($N+1$ 位以下): 確保していた与信枠に対し cancel (Void) を実行し，即座に解放する．

4.6.3 抽選販売

価格の高騰を抑制したい場合や，機会の平等を重視する場合に用いられる方式である．技術的な特異点は，応募プロセスにおける事前与信の導入にある．

■フローとステート管理

応募時　ユーザーが「応募」ボタンを押下した瞬間に，商品価格分の与信枠確保を実行する．確保に失敗した場合，応募は受理されない．

抽選時　擬似乱数生成器を用いて当選者を選出する．

確定時　当選者の取引ステータスを CAPTURED に更新し，決済を確定させる．同時に，落選者に対しては VOID 処理を行い，与信枠を即時解放する．

この仕組みにより，従来のアナログな抽選販売で頻発していた「当選連絡後の音信不通」や「支払い拒否」による再抽選の手間を完全にゼロにし，運用コストを最小化している．

4.6.4 定額販売

在庫が無制限，あるいは需給バランスが安定している作品向けの，一般的な EC と同様の販売方式である．即時決済（Capture）によって取引が成立する．

4.7 テストデータに対する要件と課題

このような「文脈重視型」かつ「複雑なリレーションを持つ」プラットフォームの開発において，テストデータの用意は大きな課題であった．

1. **リレーションの整合性**: ランダムなデータを生成すると，「ユーザーが未参加のプロジェクトに関連する作品を出品している」といった外部キー制約違反や論理矛盾が頻発する．これを防ぐためには，データベースの依存関係を深く理解した生成ロジックが必要となる．
2. **コンテンツのリアリティ**: UI の検証（例えば，長い説明文がレイアウト崩れを起こさないか，検索機能が適切にキーワードを拾うか）を行うためには，「Lorem Ipsum」に代表される無意味なダミーテキストではなく，ドメイン固有の用語や適切な長さを持つ「リアルなテキスト」が必要である．特に本プラットフォームでは「制作ログ」や「物語」が主要なコンテンツであるため，その質が UX 評価に直結する．
3. **多様性の確保**: 検索アルゴリズムやレコメンデーション機能の性能を評価するためには，データの分布に偏りがあってはならない．特定のカテゴリ（例：アクセサリー）だけにデータが集中すると，システムの性能を正しく評価できない．

第 3 章で提案したデータ生成エージェントは，これらの課題，特に「整合性」と「多様性」の両立という課題を解決するために設計されたものであり，本プラットフォームはその有効性を検証するための最適なテストベッドであると言える．

5 評価実験

本章では，提案手法である「RAG を用いたデータ生成エージェント」の有効性を検証するために実施した比較実験の結果について述べる．実験の目的は，以下の仮説を定量および定性の両面から検証することである．

> **仮説**: 過去の生成履歴を参照（RAG）することで，エージェントは「類似内容の重複生成（モード崩壊)」を回避し，結果としてより多様かつ具体的で品質の高いデータを生成できる．

5.1 実験設定

以下の条件において，それぞれ 1000 件のシナリオデータ（User-Project-Item のセット）を連続生成し，評価用データセットとした．

Table2　実験条件とパラメータ

実験 ID	基盤モデル (LLM)	RAG (メモリ参照)	データ数
A-1	Gemini 3 Pro	**ON**	1000
A-2	Gemini 3 Pro	OFF	1000
B-1	Gemini 3 Flash	**ON**	1000
B-2	Gemini 3 Flash	OFF	1000
C-1	GPT-5.2	**ON**	1000
C-2	GPT-5.2	OFF	1000
D-1	GPT-5 Mini	**ON**	1000
D-2	GPT-5 Mini	OFF	1000

すべての条件において，生成カテゴリは `PLASTIC_MODEL`, `HANDCRAFT`, `GADGET` の 3 つを均等に生成するよう指示し，創造性を最大化するために Temperature パラメータは `0.9` に設定した．

5.2 生成データの実例

本実験により生成されたデータの一例として，「夜間大学・無線同好会」というテーマに基づいたシナリオをコードリスト 3 に示す．この例では，3 名のユーザー（霧島えみ，矢木拓真，水瀬ナオ）と，彼らが共有するプロジェクト（都市電波ノイズ可視化マップ），および制作された作品（Item）が生成されている．特に注目すべきは，‘User‘の ‘bio‘や ‘Item‘の ‘description‘における記述の具体性である．単に「アンテナを作る」だけでなく，「法規リテラシー」や「共同校正」，「可視化は派手さよりも比較できることが重要」といった，具体的かつ専門的な文脈（社会的シナリオ）が一貫して記述されている．

```
1 {
2   "theme": 夜間大学の「電波観測・自作アンテナ同好会」が運営する、都市電波ノイズの“見える化”ガジェット共同制作組織"",
3   "category": "GADGET",
4   "structureType": "COMMUNITY",
5   "users": [
```

```
    {
      "name": 霧島えみ" (LEADER)",
      "bio": 夜間大学の社会人学生。"と自作アンテナを社会に接続したい。「都市のノイズは見えない公害」として可視化を推進。SDR",
      "role": "LEADER"
    },
    {
      "name": 矢木拓真" (MEMBER)",
      "bio": 金属加工屋。アンテナの「曲げ」と「固定」の沼担当。現場で使える形に落とすのが得意。"",
      "role": "MEMBER"
    },
    {
      "name": 水瀬ナオ" (MEMBER)",
      "bio": "DSP組込み担当。ノイズの“気配”を/に落とす役。法規とリテラシーの説明も担当。UI",
      "role": "MEMBER"
    }
  ],
  "project": {
    "name": 都市電波ノイズ可視化マップ"",
    "description": 街の中で増え続けるノイズを測って・比べて・説明できる形にする。法規・電波リテラシーを重視。"",
    "mode": "THEME"
  },
  "items": [
    {
      "name": ウォーターフォール"パックUI",
      "description": 夜間観測用テーマ＋ログ出力設定。可視化は派手さよりも「比較できる」ことが重要。"",
      "authorId": "u_minase"
    },
```

```
33      {
34        "name": 小型"八木アンテナUHF",
35        "description": 方向でノイズを読む。組立治具データ付き。ノイズフロアの持ち上がりを見比べる。"",
36        "authorId": "u_yagi"
37      }
38    ]
39  }
```

Listing 3　生成されたシナリオデータの例（夜間大学・無線同好会）

5.3　評価 1: 語彙数の増加推移

生成されたデータセットの多様性を測る指標として，テキストに含まれる「ユニーク単語数」の増加推移を解析した．もしエージェントが類似した表現ばかり繰り返していれば，語彙数の増加は早期に飽和するはずである．逆に，多様なテーマを生成し続けていれば，語彙数は線形に増加し続ける．

5.3.1　解析手法

各シナリオの `theme` フィールドを対象に，形態素解析（分かち書き）を行い，自立語（名詞・動詞・形容詞等）の累積ユニーク数をカウントした．

5.3.2　結果と考察

図 4 に，生成ステップ数に伴う語彙数の増加推移を示す．

実験の結果，1000 件生成時点での総語彙数は以下の通りとなった．

- **GPT-5.2**: RAG ON で 4,425 語に対し，OFF では 3,368 語 (**+31.4%**)
- **Gemini 3 Flash**: RAG ON で 3,041 語に対し，OFF では 2,187 語 (**+39.0%**)

すべてのモデルにおいて，RAG を有効にした場合の方が圧倒的に多くの語彙を使用していることが確認された．特にグラフの傾きに注目すると，RAG OFF 群（図中暖色系）は後半にかけて飽和傾向（傾きが緩やかになる）が見られるのに対し，RAG ON 群（図中寒色系）は直線的な成長を維持している．これは，エージェントが記憶を参照し，「未出の単語」や「未開拓のトピック」を能動的に選択していることを強く示唆している．

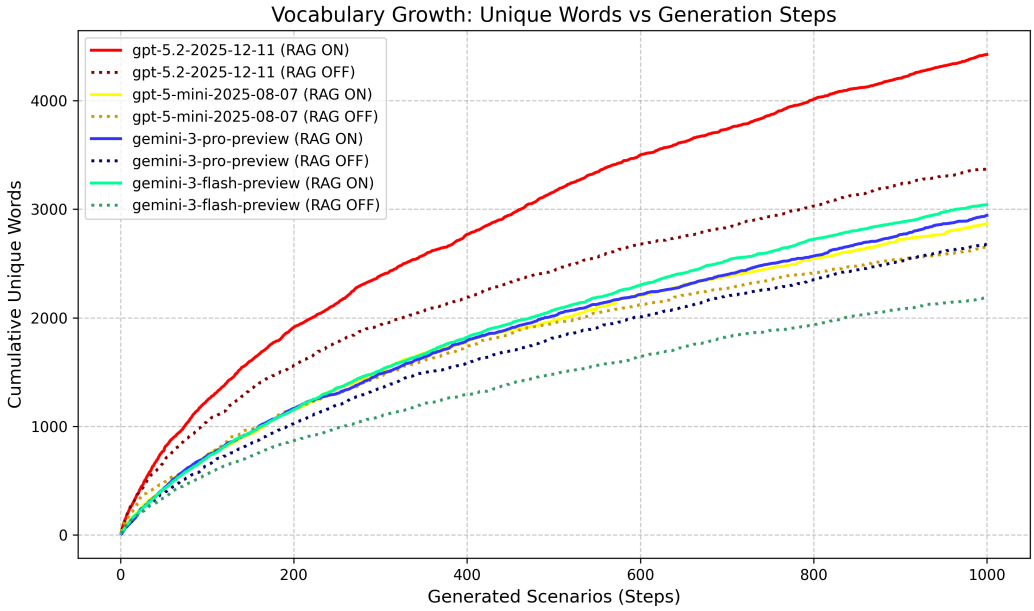

Figure4　生成数に伴うユニーク語彙数の増加推移

5.4　評価 2: ベクトル類似度による重複検知

次に，生成されたデータ同士がどの程度似通っているか（類似性の程度）を検証した．

5.4.1　解析手法

各シナリオに含まれるすべてのエンティティ（Theme, User, Project, Item）のテキストプロパティ（名前，説明文，属性など）を連結し，単一の文字列として Embedding モデル（`Google GenAI Embedding: embedding-001`）によりベクトル化した．なお，本実験では 768 次元の埋め込みベクトルを使用し，全ペア（${}_{1000}C_2$ 通り）のコサイン類似度分布を計算した．

5.4.2　結果と考察

図 5 にコサイン類似度の分布を示す．また，表 3 に各モデルにおける類似度の変化と効果量を示す．

実験の結果，Gemini 3 Pro を除くすべてのモデルにおいて，RAG を有効にすることで類似度の分布が有意に低下（左方シフト）することが確認された．特に GPT-5.2 においては，効果量 $d = -0.51$ という中程度の効果が認められ，RAG が「既存のシナリオとの衝突」を強力に回避していることが示された．Gemini 3 Pro において変化が見られな

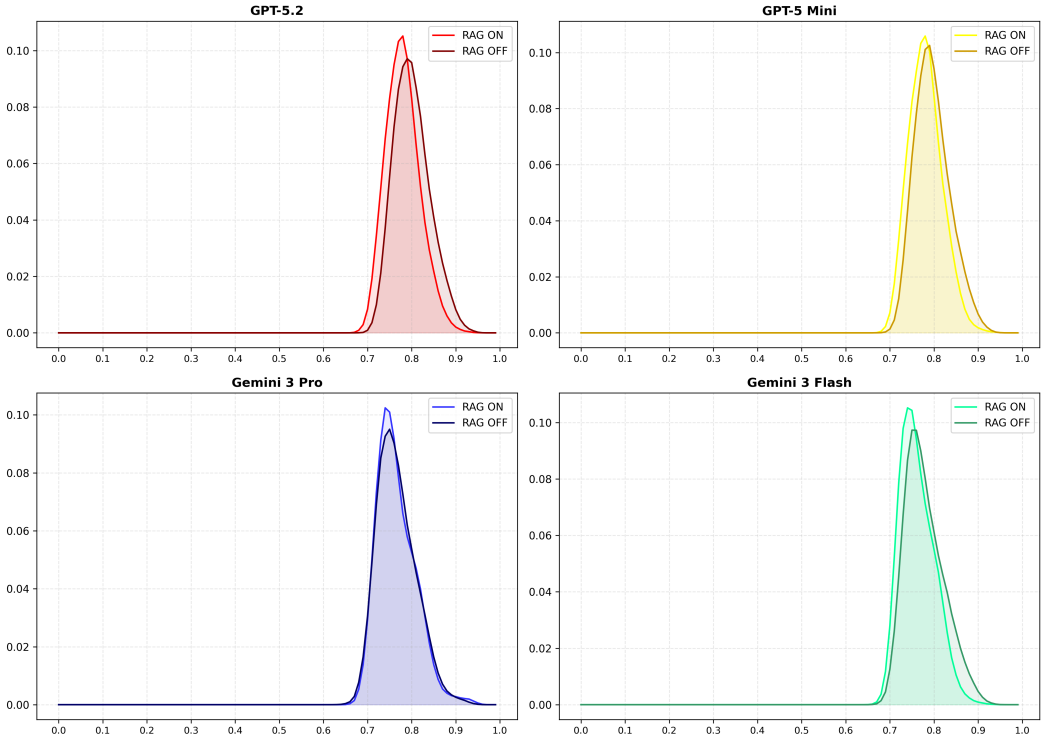

Figure5　生成データ間のコサイン類似度分布（全プロパティ結合 Embedding）

Table3　RAG 有無による平均類似度の変化と効果量

モデル	平均類似度 (OFF → ON)	差分	Cohen's d
GPT-5.2	0.806 → 0.785	**-0.021**	**-0.51**
GPT-5 Mini	0.802 → 0.786	-0.016	-0.40
Gemini 3 Flash	0.783 → 0.767	-0.016	-0.39
Gemini 3 Pro	0.771 → 0.770	-0.001	-0.01

かった点については，元々のベースモデルが持つ多様性が非常に高く（OFF 時ですでに 0.771 と低い），RAG による補正の余地が少なかったためと推察される．

5.5　評価 3: LLM Judge による定性評価

多様性を追求した結果，データの品質（整合性やリアリティ）が損なわれては意味がない．そこで，第三者 LLM を用いた定性評価を実施した．20 件ずつランダムに抽出したシナリオに対し，LLM Judge が以下の 3 観点で評価を行った．

5.5.1　評価基準

各シナリオに対し，以下の 3 つの観点を 5 段階でスコアリングした．

1. **整合性**: 設定の矛盾がないか.
2. **具体性**: 固有名詞，数値，専門用語が含まれているか.
3. **人間らしさ**: AI 特有の機械的な記述ではなく，熱量や生活感があるか.

5.5.2 結果概要

全 4 モデルにおける RAG の有無によるスコア比較を表 4 に示す.

Table4　LLM Judge による定性評価結果（各条件 20 件抽出，5 段階評価）

モデル	条件	整合性 (Coherence)	具体性 (Specificity)	人間らしさ (Human-like)
GPT-5 Mini	OFF	5.00	4.75	4.15
	ON	**5.00**	**4.95**	**4.30**
Gemini 3 Flash	OFF	5.00	5.00	4.75
	ON	**5.00**	**5.00**	**4.95**
GPT-5.2	OFF	5.00	5.00	5.00
	ON	5.00	5.00	4.95
Gemini 3 Pro	OFF	5.00	5.00	5.00
	ON	4.95	5.00	5.00

結果として，すべてのモデルにおいて「整合性」はほぼ満点（5.00）を記録し，RAG の有無にかかわらず，提案手法（社会的シナリオを経由する生成パイプライン）自体が高い整合性を保証していることが確認された．また,「具体性」や「人間らしさ」についても，条件間での有意な差は見られず，RAG を適用しても生成データの品質が損なわれないことが示された.

5.5.3 定性的考察：具体性と専門性

生成されたシナリオの内容を精査すると，RAG の有無にかかわらず,「専門的な技法」や「強いこだわり」が反映された高品質な記述が多数確認された．以下に，LLM Judge によって抽出された代表的な事例を示す.

事例 1: **専門的具体性の向上**

テーマ: 自然風化・天然ウェザリング至上主義「プラスチック盆栽」学会

評価コメント: 「具体的なキット名（RG ザク、サザビー Ver.Ka)、設置環境（室外機の上)、劣化メカニズム（加水分解、紫外線退色）が極めて詳細かつ論理的に描写されています。塗装＝欺瞞という独自の哲学にも熱量を感じます。」

事例 2: **独自の視点と人間らしさ**

テーマ: 巨大人型兵器の足元にある「都市インフラ」景観保存委員会

評価コメント: 「『ロボット本体には関心がない』という特異なテーマが一貫しており、0.2mm 真鍮線や座屈、応力集中といった専門用語が多用され解像度が高い。AI 特有の不自然さを感じさせない，強いこだわりが反映された内容である.」

事例 3: **現代的な風刺とユーモア**

テーマ: デジタル世界の『不快な UI』を物理ガジェットとして再現する

評価コメント: 「Cookie 同意バナーや reCAPTCHA といったデジタルストレスを物理化するという発想が秀逸。『トロールの快感』といったブラックユーモアを含む記述には、AI 生成とは思えない強い人間味が感じられます。」

記憶機構は過去のシナリオとの矛盾を回避するために実装した機構であり，生成される個々のデータのクオリティにはほとんど影響しないと考えられる.

5.6 評価 4 数値的類似度と意味的多様性の関係

前節までの評価により, RAG の導入がコサイン類似度を低下させ, 語彙数を増加させることが確認された．しかし，Embedding の数値的な類似度が低いことが，必ずしも「人間にとって意味のある多様性」につながっているとは限らない．そこで本節では，GPT-5.2 (RAG ON) および Gemini 3 Pro (RAG ON) において生成されたシナリオペアを類似度のレンジ別に抽出し，その内容を比較することで，数値的な類似度がシナリオの構造や意味の多様性をどのように反映しているかを定性的に検証する.

5.6.1 GPT-5.2

実際の生成テキスト（ユーザー設定および代表アイテム）の比較を図 6 および図 7 に示す.

高類似度ペア（図 6）では,「週末のブラックアウトごっこ」と「マンションでのベランダ発電」という異なるテーマを扱っているが,「安全規格や監査プロトコルへの執着」「配電・ヒューズ・保護回路といった泥臭い実務」を重視するコミュニティの姿勢が酷似している.

一方，低類似度ペア（図 7）では，どちらも「ログを取る」という行為を行っているが，その目的が対照的である．シナリオ C は「泡の揺らぎや熱の余韻」という**感性的・詩的**

シナリオ A: 停電ごっこ運用委員会
User: 黒田イサム（終末論系ラジオ番組の常連投稿者。口は悪いが安全規格にはうるさい「停電ごっこ運用委員会」委員長。週末に“世界が終わった体”で暮らすのが趣味。オフグリッド DIY はロマン、BMS は現実。イベントでは必ずブレーカを落とす前にチェックリストを読ませるタイプ…）
Item: 停電ごっこスターター木箱（配電・照明・最低限通信）
Description:
週末の模擬ブラックアウト（停電ごっこ）で、最初に必要なものを「木箱」にまとめたスターターセット。見た目は終末、中身は現実。

- 5V/12V の簡易配電／低消費電力の LED 照明／最低限の連絡手段
- **内容物**: ヒューズ付き配電ブロック（12V 系）、USB 5V 取り出し（過電流保護あり）、電圧表示（夜間視認優先の控えめ輝度）
- **注意**: バッテリーは付属しません（電池は各人の責任、でも事故は全員の責任）。自作バッテリー管理 BMS のチェック項目は同梱の手順書に従ってください。

シナリオ B: ベランダ・マイクログリッド協議会
User: 神名（築 14 年・400 戸規模マンションの管理組合で設備系の議題を担当。ベランダ太陽光と超小型蓄電で“家庭内で完結する省電力ガジェット”を作るのが趣味。DIY は好きだが、共用部ルールと安全監査はもっと大事派。電気工事士ではないので、手を出せる範囲は徹底的に線引きする…）
Item: 窓際ベランダ太陽光→ USB 微量充電キット（匿名レビュー付き）
Description:
ベランダ太陽光を“置くだけ”で始めるための、超小型構成の部材キット。マンションの共用部ルールを守るため、手すり外側への張り出しや外壁固定は前提にしていません。

- **できること**: 晴天の短時間で、USB 機器へ“ちょっと足す”程度の給電。
- **同梱部材**: 小型ソーラーパネル、低電流向け充電制御モジュール、小容量セル＋保護基板（構成は監査プロトコルに準拠）、ヒューズ相当（過電流保護）。
- **騒音・発火リスク監査**: 充電電流は低めに設定。連続充電時の温度ログを添付。

Figure6　GPT-5.2 高類似度ペア (High: 0.89)

シナリオ C: コインランドリー常連『回転する時間』
User: ミナ（終電後の街で、コインランドリーだけが明るい。私はそこで回っているものを見ています。洗濯槽の周期、泡の立ち上がり、乾燥の熱が糸に残す癖。服がきれいになる場所というより、都市の生活が一度ほどけて、また巻き取られていく小さな舞台…）
Item: 泡軌道（Foam Orbit）ピアス／イヤリング | 一点物
Description:
ガラス越しに見える泡は、いつも同じようで、同じではない。このピアスは、深夜のコインランドリーで観察した“泡の帯”を、輪郭として固定する試みです。

- **素材**: リサイクル繊維（古着コットン）、熱で変化する糸。
- **回転パターン採取**: 洗濯槽の窓を 12 分割して泡の濃淡を記録。泡が厚く残る区間は編み密度を上げ、途切れが出る区間は糸を飛ばす。
- **同梱ログ**: 観察時刻（02:06 – 02:33）、場所、所感（泡が一度厚くなってから急に薄くなる）を記した紙片を封入。

シナリオ D: ガンプラ静音研究コミュニティ
User: マキ（平日 22 時以降しか机に座れない社会人モデラー。防音室なし・ワンルーム・隣室あり。「音を出さない工夫＝制作の自由度」と考えていて、静音工具の比較、サイレント改造の手順、作業音ログを地道に残すタイプ…）
Item: RX 系 クリック関節の“カチ音”低減チューニング
Description:
深夜にポージング確認してると、関節の「カチッ」が響く。保持力を落としすぎず、音を丸めるための調整。

- **手順**: 分解前に布を敷く（落下音対策）。クリック部の当たり面を超微量面取り。受け側に薄いテープを点貼り。
- **作業音ログ**: 時間帯（23:40〜00:25）、音の種類（クリック音・高音）、対策（布二重敷き）、体感（「カチッ」→「コクッ」に変化）。

Figure7　GPT-5.2 低類似度ペア (Low: 0.74)

な現象を記録して作品に昇華させているのに対し，シナリオ D は「dB や振動」という**物理的・工学的な現象**を記録して騒音対策という実利に変えている．

5.6.2　Gemini 3 Pro

Gemini 3 Pro (RAG ON) の比較事例を図 8 および図 9 に示す．

シナリオ E: AnatomyModeler_K
User: AnatomyModeler_K（架空の『第 3 工科大学・機械構造学科』で教鞭を執っているという設定のモデラー。模型を「玩具」ではなく「工業製品のサンプル」または「解剖学標本」として捉え、徹底的な内部構造の再現と、それを可視化するためのカットモデル制作を行う…）
Item: 1/100 MS 脚部アクチュエータ構造断面
Description:
汎用人型機動兵器の膝関節周辺を対象としたカットモデル。

- **制作のポイント**: 設定画には存在するがキットで省略された「流体パルス伝達パイプ」をスクラッチビルド。
- **切断面の処理**: 装甲の切断面は「赤色」で塗装し、教育用・解説用のカットモデルであることを強調（実車カタログ等の技法）。
- **解剖学的視点**: 膝装甲の連動スライドギミックと油圧ピストンの同期を観察できるよう、メインアーマーを正中線でカット。

シナリオ F: Dr. Cross-Section / 断面解剖研究所
User: Dr. Cross-Section（「装甲は嘘をつくが、フレームは真実を語る」。既成のキットを外科手術の如く切断し、内部メカニズムを露出させる『カットモデル』専攻。ウェザリングはノイズ。清潔な手術室で鋼鉄の巨人の内臓を愛でたい…）
Item: 【RX 系】正中矢状断面：コアブロックシステムの機能的配置
Description:
1/100 スケールキットを使用し、エッチングノコによる手作業で正中線から完全に二分割した作品。

- **解剖学的ポイント**: コアブロック周辺の変形ヒンジ構造と、パイロットシートの衝撃吸収ダンパーをプラ板で新造。
- **塗装**: 露出したフレーム断面はシルバー、切断面のエッジを蛍光レッドでハイライトし、「切断面であること」を強調。博物館展示風のフィニッシュ。

Figure8　Gemini 3 Pro 高類似度ペア (High: 0.95)

Gemini 3 Pro の高類似度ペア（図 8）は，類似度 0.95 という高い値を示した．両者とも「架空メカの解剖」「切断面の赤色塗装」というポイントが重複している．

シナリオ G: CyberMonk_Logic（サイバー仏教）
User: CyberMonk_Logic（元組み込みエンジニア、現・在宅僧侶。テクノロジーによる功徳（Merit）の最大化と自動化を研究する「Cyber-Buddhism」の実践者。はんだ付けは動禅であり、回路設計はマンダラ構築...）
Item: PCB Talisman: 厄除け回路基板 v2.0
Description:
従来の紙製のお守りを、FR-4 ガラスエポキシ基板で再定義した PCB Talisman。

- **技術仕様**: 4 層基板の内層に般若心経のバイナリデータを銅箔パターンとして埋め込み、通電によりデジタルな結界を展開。
- **実装**: 中央の MCU が乱数生成により 24 時間体制で吉凶をシミュレートし、最適な運気を LED インジケータで可視化。

シナリオ H: 特定外来生物資源化プロジェクト
User: 郷田博士（元環境保全研究所の生物学者。日本の在来種を脅かす「特定外来生物」の駆除活動に参加する中で、奪った命を廃棄することへの葛藤から、それらを「ジビエレザー」として昇華させる道を選んだ...）
Item: 【特定外来生物】ヌートリアの撥水ファーポーチ
Description:
河川の土手を掘り崩す大型齧歯類「ヌートリア」の毛皮を使用したポーチ。

- **生態学的特徴**: 水辺で生活するため、驚異的な撥水性と保温性を持つ。ビーバーに似た濃密な手触り。
- **制作背景**: 兵庫県の河川敷で行った個体数調整（駆除）の際に捕獲した個体。肉はコンポスト化し、最も美しい冬毛の部分をなめし加工。命の温かみを道具に変換。

Figure9　Gemini 3 Pro 低類似度ペア (Low: 0.73)

低類似度ペア（図 9）では，「デジタルによる精神の自動化（サイバー仏教）」と「狩猟による生命の資源化（外来生物）」という，テクノロジーと自然の対極的な世界観が提示されている．

5.7　まとめ

以上の実験結果より，RAG を用いた記憶機構は，単にデータの重複を防ぐだけでなく，エージェントに多様なデータの生成を促し，結果として高品質かつ高密度なテストデータを生成させるための補助となることが実証された．

6　結論

6.1　まとめ

本研究では，一点物プラットフォームの開発において，社会的シナリオやそれを反映したスキーマを持ったテストデータ生成の課題を解決するために，RAG を搭載した LLM エージェントシステムを提案した．また，その有効性を検証するためのテストベッドとして，実際にモダンな Web プラットフォームを構築した．

従来の統計的手法やランダム生成では再現が困難であった「社会的シナリオ」に基づく整合性の高いリレーショナルデータを，LangGraph を用いたステートマシンによって自律的に生成する手法を確立した．

提案システムの有効性を検証するために実施した比較実験により，以下の知見が得られた．

1. **多様性の確保と衝突回避**: 履歴をデータベースに登録し，それを参照しながらデータを生成することで，生成されるテキストの 1000 件時点でのユニーク語彙数が

Gemini 3 Flash において 39.0% 増加した。また、生成された社会シナリオおよび各エンティティのテキストをエンベディングし、全ペアのコサイン類似度分布を分析した結果、GPT-5.2 において効果量 $d = -0.51$ という結果が得られた．これにより，記憶機構が生成内容の重複を抑制し，生成されるデータ全体の多様性を向上させることが示された．

2. **探索空間の拡張による具体性の向上**: 定性評価の結果,「夜間大学の無線同好会」の事例に見られるように，User の経歴から Item の技術的仕様に至るまで，整合性の取れたデータが構築されていることが確認された．

6.2 今後の展望

今後の課題として，まず本エージェントの実践的な有用性のさらなる検証が挙げられる．本研究では，文脈を持つテストデータの生成までを主眼としたが，本来の目的はこのデータを用いてアプリケーションの高度な機能を開発することにある．特に，文脈を考慮した検索システム，複雑なリレーションを可視化する検索 UI，あるいはユーザーの行動文脈に基づいたレコメンデーションシステムといった機能は，意味のあるデータが存在しなければ開発も評価も難しい．今後は，本エージェントによって生成された大量のシナリオデータを活用し，これらの機能を実際に開発・実装することで，本手法が実際の Web サービス開発プロセスにおいて具体的にどのように貢献できるかを確認していく必要がある．

さらに，本手法の一般化も重要な課題である．現在は一点物売買プラットフォームという特定のドメインに特化したデータモデルを用いているが，エージェントのアーキテクチャ自体は汎用的なものである．今後は，このフレームワークを他の Web サービスや，異なるドメインを持つアプリケーションにも適用できるよう一般化し，広範な開発現場で利用可能なツールとして昇華させることを目指す．

謝辞

本研究を遂行するにあたり，多大なるご指導とご助言を賜りました，指導教員の植松幸生准教授に深く感謝いたします．また，日頃より議論を交わし，有益な示唆を頂いた研究室の皆様に心より感謝申し上げます．

参考文献

[1] Eric Sheridan. *The creator economy could approach half-a-trillion dollars by 2027*. Goldman Sachs Research. Accessed: 2026-02-12. 2023. URL: https://www.goldmansachs.com/intelligence/pages/the-creator-economy-could-approach-half-a-trillion-dollars-by-2027.html.

[2] Inc. Mercari. *Mercari: Your Marketplace*. https://jp.mercari.com/. Accessed: 2026-02-12. 2024.

[3] Inc. GMO Pepabo. *minne* — **ハンドメイド・手作り・クラフト作品のマーケット**. https://minne.com/. Accessed: 2026-02-12. 2024.

[4] Inc. BASE. *BASE (**ベイス**)* — **ネットショップを無料で簡単に作成**. https://thebase.com/. Accessed: 2026-02-12. 2024.

[5] pixiv Inc. *BOOTH* - **創作物の総合マーケット**. https://booth.pm/ja. Accessed: 2026-02-12. 2024.

[6] Marak Squires et al. *Faker.js: generate massive amounts of fake data in the browser and node.js*. https://github.com/faker-js/faker. 2022.

[7] J Edvardsson. "A survey on automatic test data generation". In: *Proceedings of the Second Conference on Computer Science and Engineering in Linköping* (1999), pp. 21–28.

[8] DiUS. *JavaFaker*. https://github.com/DiUS/java-faker. Accessed: 2026-02-02. 2024.

[9] *Mockaroo: Random Data Generator and API Mocking Tool*. https://www.mockaroo.com/. Accessed: 2026-02-02. 2024.

[10] Prisma Data. *Prisma: Next-generation ORM for Node.js and TypeScript*. https://www.prisma.io/. Accessed: 2026-02-02. 2024.

[11] Lei Xu et al. "Modeling tabular data using conditional gan". In: *Advances in Neural Information Processing Systems*. Vol. 32. 2019.

[12] Martin Jurkovič, Valter Hudovernik, and Erik Štrumbelj. "SyntheRela: A Benchmark For Synthetic Relational Database Generation". In: *ICLR 2025 Workshop on SynthData*. 2025.

[13] Haoyang Li et al. "OmniSQL: Synthesizing High-quality Text-to-SQL Data at Scale". In: *Proceedings of the VLDB Endowment*. 2025.

[14] Mushtari Sadia et al. "SQUiD: Synthesizing Relational Databases from Unstructured Text". In: *arXiv preprint arXiv:2505.19025* (2025).

[15] Kyungho Kim et al. "ReFuGe: Feature Generation for Prediction Tasks on Relational Databases with LLM Agents". In: *Proceedings of the ACM Web Conference 2026*. 2026.

[16] Joon Sung Park et al. "Generative agents: Interactive simulacra of human behavior". In: *Proceedings of the 36th Annual ACM Symposium on User Interface Software and Technology*. 2023, pp. 1–22.

[17] Harrison Chase. *LangChain*. https://github.com/langchain-ai/langchain. Accessed: 2026-02-02. 2022.

[18] Shunyu Yao et al. "ReAct: Synergizing Reasoning and Acting in Language Models". In: *International Conference on Learning Representations (ICLR)*. 2023.

[19] Vercel. *Next.js by Vercel - The React Framework*. https://nextjs.org/. Accessed: 2026-02-02. 2024.

[20] Kamil Mysliwiec et al. *NestJS - A progressive Node.js framework*. https://nestjs.com/. Accessed: 2026-02-02. 2024.

[21] PostgreSQL Global Development Group. *PostgreSQL: The World's Most Advanced Open Source Relational Database*. https://www.postgresql.org/. Accessed: 2026-02-02. 2024.

[22] Stripe. *Stripe API Reference*. https://stripe.com/docs/api. Accessed: 2026-02-02. 2024.