

# 大規模言語モデルを用いた 1DCAE のモデリング作業の支援

池田 孟<sup>\*1</sup>, 石橋 祐太郎<sup>\*2</sup>, 金丸 正寛<sup>\*3</sup>, 小林 孝<sup>\*1</sup>

## Making research paper (About the use of the JSME specification template file)

Taro KIKAI<sup>\*1</sup>, Sakura GIJYUTSU<sup>\*2</sup>, Hanako TOKYO<sup>\*3</sup> and Hanako TOKYO<sup>\*1</sup>

<sup>\*1</sup> Japan Society of Mechanical Engineering  
35Shinanomachi, Shinjuku-ku Tokyo 160-0016, Japan  
<sup>\*2,\*3,\*1</sup> Department of Mechanical Engineering, Kikai University  
1 Kogakumachi, Shinjuku-ku, Tokyo 160-0001, Japan

Received: XXXX; Revised: XXXX; Accepted: XXXX

### Abstract

In complex system design, where mechanical, electrical, and control subsystems interact, 1D CAE is increasingly adopted to optimize products quickly by verifying requirements through early-stage physical simulations on virtual prototypes. The Modelica language has become the standard for modeling these systems due to its object-oriented, declarative nature and ability to handle multi-domain interactions. It supports hierarchical modeling using extensive libraries like the Modelica Standard Library, which includes components for thermal, electromagnetic, and control domains. Recent advancements in generative AI, particularly Large Language Models (LLMs), have enabled new approaches in design engineering, such as automating design system handling and managing design knowledge. This study presents a framework that uses Retrieval-Augmented Generation (RAG) to search closed corporate Modelica repositories and employs three collaborative agents—a planner, coder, and tester—to automatically assemble libraries. Experiments using OpenAI-4o demonstrated that this agent-based workflow could generate candidate code for tasks that RAG alone could not handle. The results also showed improved code generation performance when working with proprietary internal libraries, highlighting the potential of combining LLMs with structured agent collaboration in engineering design automation.

**Keywords** : Term1, Term2, Term3, Term4, ...(Show five to ten key words)

## 1. 緒 言

複雑なシステムの設計においては、機械、電気、制御の各サブシステムが相互に干渉する複雑な製品を短期間で最適化するため、設計初期から物理シミュレーションを活用して仮想プロトタイプ上で要求充足性を検証する 1DCAE が主流となりつつある(日本機械学会, 2020)。物理シミュレーションモデルの記述には、オブジェクト指向かつ宣言型で多領域連成を自然に扱える Modelica 言語が事実上の標準となっている。Modelica Standard Libraryをはじめ、熱流体、電磁気、制御系など多様なドメインのサブモデルがライブラリとして公開されており、ユーザは既存コンポーネントをインポートして階層的にモデルを構築できる。近年、設計工学では、生成 AI を用いて設計知識を管理し、設計システムの取り扱いを自動化する試みが行われてきた(藤田喜久雄, 2023)。大規模言語モデル Large Language Model (LLM)を中心とした生成 AI の性能は大きく向上し、さまざまな設計問題への応用が検討されてされている。本研究は、企業内でクローズされた 1DCAE のライブラリの作業支援・読解支援に LLM を適用した。企業内でクローズされた Modelica のライブラリを対象に RAG で関連ドキュメントを検索し、プランナー、コーダ、テスターの三つのエージェントが協調してライブラリを自動組立するフレームワークを構築した事例を報告する。OpenAI 4o-mini を用いた実験では RAG だけのシステムでは、生成できなかったコード

作成タスクにおいて候補コードが生成できるようになることを確認できた。企業内のクローズなライブラリの操作においてもエージェントワークフローによってコード生成能力が向上することがわかった。

## 2. 先行研究

### 2.1 LLM の設計の展開事例

2024 年以降に設計問題の LLM を適用した事例を紹介する。Picard らは、概念設計から製造・検査に至る四つの設計フェーズを対象として、マルチモーダル Vision-Language Model (VLM) である GPT-4V と LLaVA 1.6 34B の有効性を体系的に評価した(Picard et al., 2024)。同研究は VLM が生成した回答根拠を設計図や仕様書にトレースする手法を提案し、設計者が結果を検証しやすい解釈性向上の方向性を示した。

トヨタ自動車は、熟練エンジニアの暗黙知を LLM にリンクさせて次世代車両の開発速度を高める目的で、マルチエージェント型の設計支援基盤を構築している(Onishi, 2024)。同社では、パワートレイン設計部門において GenAI エージェントが資料検索、要件整理、設計レビューの各フェーズを自律分担し、意思決定を加速する事例が報告された。三菱電機は、設計・生産準備を含む ECM (Engineering Chain Management) 領域に生成 AI を組み込む社内プラットフォームを構築し、2017 年以降の AI 活用技術を集約したソリューション群を展開している(玉谷基亮, 2024)。同社は技術文書のレイアウト解析と差分照合を行う自然言語処理 AI、FMEA/DRBFM データを参照して新規設計変更リスクを自動抽出するリスク抽出 AI、CAE 解析結果を学習したサロゲートモデル AIなどを設計フローに導入し、設計パラメーター探索の高速化と知見の再利用を図った。さらに、デジタルツインと生成 AI エージェントを連携させ、IoT で収集した実機データと仮想空間のシミュレーション結果を横断的に解析しながら設計条件を最適化する将来像を提案している。

企業内では、社内でクローズされた情報を検索し設計活動に用いる。このように機密情報を RAG などの検索システムを構築し、知識管理に LLM を適用している。

### 2.2 IDCAE のモデリング作業の支援

物理シミュレーションや制御シミュレーションには、さまざまな専門ライブラリを用いてモデリング作業をおこなう。

このようなモデリング作業は、モデル対象への理解、プログラム言語と専門ライブラリの仕様、数値計算などの知識や経験など高度なドメイン知識と経験が必要になる。このようなモデリングとシミュレーションを実行可能なエンジニアの養成するためには、多くの時間とコストが必要になる。物理シミュレーション言語 Modelica や数値流体計算ライブラリ OpenFOAM によるモデリング作業とシミュレーション実行は、研究開発でよく使われる専門ライブラリであるが、ライブラリの複雑さと、モデル対象の理解に専門性が必要なため、高度な専門知識をもつエンジニアや研究者を必要とする。このようなシミュレーションのモデリングへの LLM の適用は、依然に難しい。最近の先行研究を紹介する。

近年の大規模言語モデル LLM はコード生成能力の向上により、Modelica コンポーネント自動生成への応用が進む。ベンチマーク「ModiGen」では、LLM が指標で 0.33 以上を達成する事例が報告されており、複雑なモデル構造でも一定の妥当性を保持する段階に到達した(Xiang et al., 2025)。OpenFOAM を用いた CFD シミュレーションの自動化を目的として、GPT-4o および CoT 機能搭載モデルを基盤とするエージェント「OpenFOAMGPT」が提案されている。本手法では、OpenFOAM チュートリアルケースから得た領域知識を検索拡張生成 (RAG) パイプラインに組み込み、入力ファイルの自動生成・実行・エラー訂正を反復的に行う構造を採用している。評価では、RAG を用いた少数ショットプロンプトにより、境界条件やメッシュ解像度の変更に対応できることが示された。一方、精度担保のための人手監視やモデル性能の時間変動が課題として指摘されている。「MetaOpenFOAM」は、Architect, InputWriter, Runner, Reviewer の四つのエージェントが連携し、ユーザーの自然言語要求から入力ファイル生成、シミュレーション実行、エラー分析・修正までを自動化する反復ループを構築する。

このような専門ライブラリの適用が LLM にとっていまだに難しい原因をまとめる。

- 高度なドメイン知識の不足 LLM はインターネット上の広範囲のデータを用いて訓練されているが、Modelica の特定のライブラリや OpenFOAM のファイル構造やパラメータ設定や各種コマンドなどの情報をインターネット上の汎用的な訓練データから学習することは、難しい。
- 信頼性と正確性の向上 シミュレーション作業のセットアップは、一般的に厳密な正確性が要求される。モデリング作業は、複数のセットアップ作業を正確に実行する必要があるが、すべての作業を正確にじっこうすることは現在の LLM にとっては、難しい。
- 企業内の情報の機密性 企業の研究開発に用いる専門ライブラリは、機密性が高く、インターネットには公開されていない。LLM の訓練データにはいっさい含まれていないため、LLM に専門ライブラリの情報を与えるために、RAG によって検索結果を LLM に与えることが一般的である。専門ライブラリの活用は、一企業内だけにとどまることがほとんどのため、ファインチューニングに必要な訓練データを用意することも難しい。

### 3. 企業でクローズされた Modelica ライブラリの読解支援

本稿では、空調開発におけるモデルベースのシミュレーションに生成 AI の支援を適用した例を報告する。当社（三菱電機）でクローズされた情報で作成された物理モデルを活用し、

#### 3.1 Modelica 言語の特徴

Modelica 言語は、オブジェクト指向の記述構文を採用した非因果的・宣言型のモデリング言語であり、複雑な物理システムの記述およびシミュレーションに適している。機械、電気、熱、流体、制御などの異なるドメインに属する物理現象を統一的に扱うことが可能であり、マルチドメインシステムの統合的設計に広く用いられている。Modelica では、物理法則を微分代数方程式（DAE）の形で直接記述することができる。この特徴により、従来の因果的（入力・出力を明示する）記述方式とは異なり、より自然な形でのモデリングが可能となる。ユーザは既存のモデルをライブラリと利用することができる。ユーザはそれらを組み合わせることで高い再利用性と保守性をもったモデルを構築できる。

#### 3.2 Modelica 言語による冷凍サイクルシミュレーション

本研究で対象としたライブラリについて説明する。三菱電機では、冷凍サイクルの物理シミュレーションに海外研究所 Mitsubishi Electric Research Laboratory (MERL) で開発した Modelica ライブラリを使用している。本稿では、このライブラリを MERL HVAC Library と呼ぶことにする。このライブラリは、エアコンなどの空気調和機の冷凍サイクルを動的な挙動をシミュレーションすることができる。冷凍サイクルを構成する熱交換器、圧縮機、膨張弁などの内部を流れる冷媒の圧力、温度、比エンタルピーといったパラメータを計算することができる。

冷媒の挙動は、非線形性が強く、動的なシミュレーションが困難である。MERL HVAC Library では、冷媒流れの差分化方法、計算スキームを選択できる (Laughman, )。

2025 年 2 月現在では、MERL HVAC Library は 8 種類のパッケージで校正されている。現在は、パッケージの全ファイル数は 3,112 個ある。各ファイルに定義されたクラスとメソッドの継承関係が複雑であることと、冷凍サイクルの物理モデリング手法が多岐にわたることがライブラリの理解を難しくしている。下記に MERL HVAC Library によるモデリングの一例としてルームエアコンのモデルを示す。一部のモデルはダイアグラム表示によってクラスの継承関係の可視化が可能であるが、ほとんどのモデルはダイアグラム表示の実装中であり、クラス継承関係はコードから読み解く必要がある。

#### 3.3 ライブラリの内部構造の理解

##### 3.3.1 シミュレーションコードの実装

モデル作成時には、標準ライブラリ（Modelica Standard Library）との接続性を意識しつつ、AixLib 内のクラス構造や継承関係を理解する必要がある。例えば、建築ゾーンのモデル AixLib.Building.LowOrder は、熱容量、外皮性能、内部発熱などをパラメータとして受け取り、建物の動的応答を表現できるように設計されている。そのため、コンポーネントの選定とパラメータ設定を適切に行うことが求められる。

さらに、AixLib は多くのサブモデルを再帰的に参照する構造となっており、Modelica のオブジェクト指向的な特徴（継承・再定義・階層化など）を十分に理解しておく必要がある。モデルの拡張やカスタマイズを行う場合には、replaceable や extends などのキーワードを用いて、既存のテンプレート構造を再利用しつつ、新たな仕様に適合させる。

シミュレーション実行時には、熱的・流体的計算の複雑性により収束性や計算時間が問題となる場合があるため、ステップサイズ・ソルバ・パラメータのスケーリングに留意する。また、AixLib はモデル記述の一貫性を保つために物理単位や制約条件が厳格に定義されているため、整合性の確認が重要である。

### 3.3.2 シミュレーションのポスト処理

Modelica 言語において、AixLib などの大規模ライブラリを利用するには、以下のような手順で開発を進める。まず初めに、使用する Modelica 開発環境を整備し、対象とするライブラリ（AixLib）の依存関係を含めて正しくインポートする。本稿では Dymola を使用している。

AixLib はシステム全体の熱的振る舞いをモデリング可能なコンポーネント群を備えており、特に建築熱負荷・HVAC・制御アルゴリズムに関するモデルが豊富に実装されている。

以上のように、複雑な Modelica ライブラリを活用するには、(i) 開発環境と依存関係の適切な管理、(ii) ライブラリ内部構造の理解、(iii) Modelica 言語の文法と設計思想の把握、(iv) 数値計算の安定性に関する知識、が不可欠である。

## 4. Modelica ライブラリの設計における役割

前節の MERL HVAC Library には、

よるモデリングの一例としてルームエアコンのモデルを示す。一部のモデルはダイアグラム表示によってクラスの継承関係の可視化が可能であるが、ほとんどのモデルはダイアグラム表示の実装中であり、クラス継承関係はコードから読み解く必要がある。

このような Modelica のライブラリは、モデルの忠実度（Fidelity）の根拠となる理論と実験が集積されている。このライブラリは、シミュレーションの実務が繰り返されるたびに、拡張されていき、複雑度を増していく。藤田ら(藤田喜久雄, 2023)が指摘するように、ライブラリそのものが設計根拠の集積である設計カタログとなっている。専門性の狭い分野のため、一般書籍や Review 誌などのまとまった情報を入手することも容易ではなく、ライブラリの学習の妨げになっている。

企業内で構築されているライブラリについては、上記と同様に、高度な理論

### 4.1 シミュレーションの方法

ここで MERL HVAC Library などの物理シミュレーションライブラリを用いてシミュレーションモデルを構築する際のエンジニアの作業プロセスを書く。ルームエアコンを例にとると、熱交換器、ファン、圧縮機、膨張弁などの該当するクラスを探索し、各クラスを継承したモデルを接続すれば良いのであるが、

また、数値差分法のアプローチを選択する。

#### 4.1.1 モデルの理論の調査

公開情報は、博士論文や最新の論文などを調べる。社外秘密の情報

### 4.2 大規模言語モデルの応用

本稿で構築した2つのシステムについて説明する。

#### 4.2.1 RAG システム

図1に示す RAG システムは4層構造で設計した。

- (1) データ：MERL HVC Library コード，ドキュメント，論文を Markdown へ変換し，Sentence-BERT で 768 次元ベクトル化して FAISS に格納する．
- (2) 検索層：ユーザ質問を同一埋め込み空間で検索し，上位  $k$  件を取得．複数チャンクを階層マージしてプロンプト長を最適化する．
- (3) 生成層：OpenAI GPT-4o-mini をベースとする LLM に対し，役割指示+検索結果+会話履歴から構成されるテンプレートを送信する．
- (4) 出力層：回答と同時に引用情報を提示し，ユーザはソースコード断片へ直接ジャンプできる．

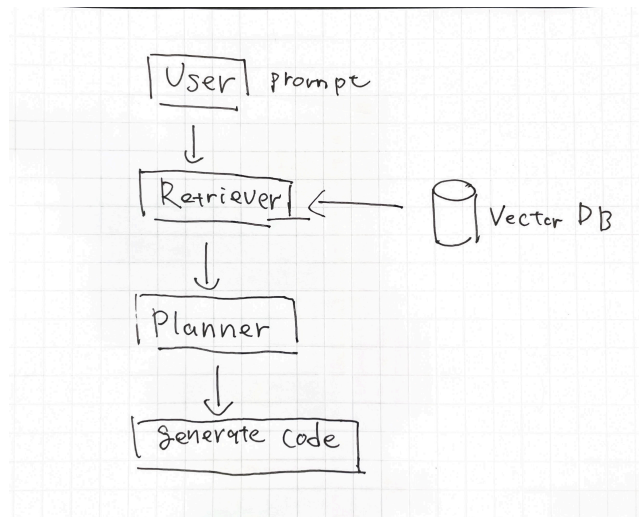


Fig. 1 Workflow chart of our agent.

#### 4.2.2 Modelica コード生成エージェント

図2にエージェントのワークフローを示す．核心は，Plan-Execute-Critique ループにより「検索→提案→コーディング」を自律実行する点である．

- (1) RAG: 前節の RAG システムを用いて，ユーザからのプロンプトに関連するドキュメントを検索する．
- (2) プランナー：RAG システムが出力した検索結果からモデル構造の手順書（pseudo-plan）を生成する．
- (3) コーダー：手順書に従いコードスニペットを生成する．

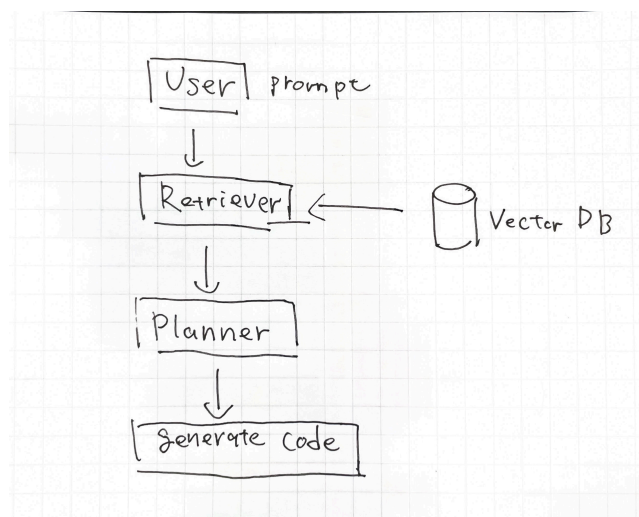


Fig. 2 Workflow chart of our agent.

## 5. 結果

### 5.1 単体システムの回答

一般知識で回答できる問題は、熱力学の知識に関する問題など、設計エンジニアにとって難易度の高い問題も回答することができる。ただし、

#### 5.1.1 公知の情報で回答可能な可能な可能質問

### 5.2 RAG システムの回答

#### 5.2.1 文書データベースの情報が回答に必要な質問

prompt

#### 5.2.2 コード作成の指示

### 5.3 Modelica コード生成エージェント

RAG システムでは、無回答であったコーディング作業にも

## 6. 結 言

本稿では、大規模言語モデルの物理シミュレーションへの作業支援について調べた。現状では、複雑なプロセスが必要な実用ライブラリのコーディング作業は完全には自動化できないが、知識探索やコードの作成方針を探索には実用できることを明らかにした。

## 文 献

日本機械学会, 設計のための 1DCAE 概念と実現技術, 日本機械学会 (2020), p. 256, 初版発行日: 2020 年 11 月 2 日.  
玉谷基亮, 三菱電機グループの持続的なものづくりを支える AI ソリューション群, 三菱電機技報, Vol. 98, No. 9 (2024), pp. 3–6.  
藤田喜久雄, 設計論: 製品設計からシステムズイノベーションへ, コロナ社 (2023), p. 494, 初版第 1 刷発行日: 2023 年 7 月 7 日.

## References

Chen, Y., Zhang, L., Zhu, X., Zhou, H. and Ren, Z., OptMetaOpenFOAM: Large Language Model Driven Chain of Thought for Sensitivity Analysis and Parameter Optimization based on CFD (2025).  
Chen, Y., Zhu, X., Zhou, H. and Ren, Z., MetaOpenFOAM: an LLM-based multi-agent framework for CFD (2024).  
Laughman, C. R., A Comparison of Transient Heat-Pump Cycle Simulations with Homogeneous and Heterogeneous Flow Models.  
Onishi, K., Lessons from toyota for building durable multi-agent copilots, Presented at Microsoft Ignite 2024 (2024), Session BRK117.  
P, Ey, S., Eep, Xu, R., Wang, W. and Chu, X., OpenFOAMGPT: a RAG-Augmented LLM Agent for OpenFOAM-Based Computational Fluid Dynamics, available from <<http://arxiv.org/abs/2501.06327>>.  
Picard, C., Edwards, K. M., Doris, A. C., Man, B., On, Giannone, G., Alam, M. F. and Ahmed, F., From Concept to Manufacturing: Evaluating Vision-Language Models for Engineering Design (2024).  
Xiang, J., Ye, T., Liu, P., Zhang, Y. and Wang, W., Modigen: a large language model-based workflow for multi-task modelica code generation (2025).