卒業論文

LLM エージェントによる
状況依存型 IoT サービス連携

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　4 回生
2600220350-1

本多 快

2025 年度（秋学期）卒業研究 3（CH）
令和 8 年 1 月 30 日

LLM エージェントによる状況依存型 IoT サービス連携

本多 快

内容梗概

近年，住宅やオフィスなどに多数の IoT デバイスが導入され，温度や湿度などの環境情報を取得するセンサと空調機器や照明などの環境に作用するアクチュエータが日常空間に広く配置されている．これに伴い，自然言語による指示によって状況に応じた最適な機器制御を実現するスマートホームサービスへの需要が高まっている．しかしながら，現状のスマートホームサービスの多くはルールベース制御という事前に定義された制御に依存しており，新規の IoT デバイスの追加やデバイスの故障による欠損値など想定外の状況に対応できない．さらに，ユーザの自然言語による要求との条件照合も困難である．

そこで，本研究では，大規模言語モデル（LLM）を用いて，ユーザの自然言語による曖昧な指示から，利用可能な IoT センサによる観測値を総合的に解釈し，実世界の状況を推定した上で，その状況に応じた IoT サービスを合成するエージェントを提案する．本エージェントは，環境状態推定，意図推論，アクチュエータ選択，デバイス間の物理的・論理的競合検証という段階的推論フローによって構成される．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．

環境状態の推定

ユーザの「部屋を快適にして」のような曖昧な自然言語指示では複数の状況が想定され，問題となる環境要因を特定できない．状況を特定するためにセンサの観測値を用いても，観測値の組み合わせをそのまま状態にすると状態爆発が生じる．そこで，環境を抽象化して状態を定義し，利用可能な異なるセンサ群から多様な観測データを収集し，LLM に現在の環境状態を推定させる必要がある．

物理的整合性の検証

IoT サービスは実空間内のアクチュエータを制御対象とするため，「冷房 ON かつ窓開放」や「加湿と除湿の同時実行」といった物理的・論理的に矛盾する操作が生じる可能性がある．合成されたサービスがこのような競合を含まないよう，物理的・論理的な整合性を自動的に検証し修正する仕組みが求められる．

一つ目の課題に対しては，環境状態を同定するために，室内環境品質（Indoor Environment Quality: IEQ）の主要 4 因子に基づき，温熱環境（thermal），空気

質（air quality），照明・視環境（lightnig），音環境（Acoustics）の 4 つの状態変数から構成される抽象的な環境状態を定義した．この定義に従い，LLM は Web of Things Thing Description (WoT TD)に則った記述方式によるメタデータの付与されたセンサリストと観測値から各環境状態を推定し，現在の環境状態を同定する．この機構により，現在の環境状態とユーザの目的状態との差分を状態変数の差として表現できるため，後続のユーザの意図推論に利用可能となる．

二つ目の課題に対しては，推論された意図，WoT TD の記述方式によるメタデータの付与されたアクチュエータリストを LLM に提示し，意図を実現するアクチュエータを選択する．選択時に排他関係や物理干渉を競合理由として出力させ，得られた競合理由を入力として制約を満たす修正後のデバイス設定を選択し，矛盾を回避するフローを組み込んだ．

提案手法の有用性を示すために，提案手法を組み込んだ段階的推論型 LLM エージェントによる IoT デバイス制御システムのプロトタイプをクラウド上に実装し，代表的なシナリオにより動作確認を行った．さらに，環境観測データを模擬的に生成し，提案手法による環境状態の同定精度を評価した．

本研究の貢献は以下の通りである．

環境状態の推定

各センサの観測値を適正値とその上下のレンジに区分けし，5 種の連続値センサを 3 値，1 種の離散センサを 2 値とし 486 パターンの観測値データを一様分布に基づいて網羅的に生成した．生成したデータに対する各状態変数の正解ラベルは評価対象の LLM とは異なる上位の LLM の多数決により決定し，提案手法の環境状態同定精度を評価した．その結果，適合率，再現率，F 値がそれぞれ 0.6362，0.6280，0.6319 となり，WoT によるメタデータにより精度がそれぞれ 1.99%，1.57%，1.77%向上した．

物理的整合性の検証

「冷房 ON かつ窓開放」や「加湿と除湿の同時実行」などの競合パターンを整理し，LLM によるアクチュエータ集合の物理的・論理的整合性検証ステップを導入するとともに，検証結果に基づき運転順序やモードを自動調整して矛盾のないデバイス設定を生成するフローを提示し，実空間における IoT サービス合成の安全性と一貫性を確保するための設計指針を与えた．

Situated IoT Service Orchestration Using LLM Agent

Kai Honda

Abstract

In recent years, many IoT devices have been introduced into homes and offices. This has led to an increasing demand for smart home services that can control devices in response to contextual commands in natural language. However, many current smart home services rely on predefined controls, making them unable to respond to unexpected situations and difficult to match with user requests in natural language.

In this study, we propose an agent that uses a large-scale language model (LLM) to interpret IoT sensor observations based on ambiguous user commands, infer real-world conditions, and synthesize IoT services appropriate to those conditions. This agent is composed of a step-by-step inference flow: environmental information acquisition, environmental state estimation, intention inference, actuator selection, and physical and logical conflict verification between devices. To realize this method, the following two challenges must be addressed:

Environmental State Estimation

A ambiguous natural language command, such as "Make the room comfortable," leads to multiple possible situations, making it difficult to identify environmental factors. Even if sensor observations are used to identify a situation, simply treating the combination of observations as a state results in state explosion. Therefore, it is necessary to abstractly define the environmental state, collect observation data from a group of different sensors, and have the LLM estimate the current environmental state.

Verifying Physical Consistency

When controlling a real space, physically and logically inconsistent operations may occur. A mechanism is needed to verify and correct physical and logical consistency to prevent such conflicts.

To address the first issue, we defined the environmental state as consisting of four state variables based on the four major factors of indoor environmental quality: thermal, air quality, lightning, and acoustics. Based on this definition, the LLM estimates and identifies each environmental state from a sensor list and

observations annotated with metadata in a description format conforming to the Web of Things Thing Description (WoT TD). This mechanism allows the difference between the current environmental state and the user's desired state to be expressed as the difference in state variables, which can be used to subsequently infer the user's intention.To address the second issue, the LLM is presented with the inferred intent and a list of actuators with metadata based on the WoT TD description format, and selects the actuator that will realize the intent. During selection, exclusion and physical interference are output as conflict reasons, and a flow is incorporated to avoid inconsistencies.

To demonstrate the usefulness of the proposed method, we confirmed the operation of a prototype IoT device control system using a stepwise inference-type LLM agent incorporating the proposed method. Furthermore, we simulated generated observation data and evaluated the accuracy of environmental state identification using the proposed method. The contributions of this research are as follows:

Environmental State Estimation

The observed values of each sensor were divided into an optimum value and its upper and lower ranges. Five types of continuous-value sensors were assigned ternary values, and one type of discrete sensor was assigned binary values. 486 patterns of observed value data were generated based on a uniform distribution. The correct label for each state variable for the generated data was determined by majority vote of a higher-ranking LLM different from the LLM being evaluated, and the accuracy of environmental state identification using the proposed method was evaluated. The resulting precision, recall, and F-measure were 0.6362, 0.6280, and 0.6319, respectively, demonstrating that the WoT metadata improved accuracy by 1.99%, 1.57%, and 1.77%, respectively.

Physical Consistency Verification

We introduced a step for verifying the physical and logical consistency of actuator sets using LLM, and based on the verification results, we presented a flow for generating consistent device configurations, providing design guidelines for ensuring consistency of IoT service composition in the real world.

LLM エージェントによる状況依存型 IoT サービス連携

目次

第 1 章 はじめに 1

第 2 章 関連研究 5

2.1 LLM を用いた相互運用型 IoT サービス基盤……5

2.2 実世界センサデータに基づく LLM 推論の強化……5

2.3 本研究の位置付け ……6

第 3 章 エージェントの段階的推論プロセス 7

3.1 環境状態 ……7

3.2 プロセスの概要……8

3.3 プロンプト設計方針……9

3.4 プロンプトの実装 ……9

3.4.1 環境状態の同定……9

3.4.2 意図推論 ……10

3.4.3 アクチュエータ選択 ……10

3.4.4 デバイス間整合性検証……10

3.4.5 デバイス設定の決定 ……10

第 4 章 状況依存型 IoT サービス制御のプロトタイプ 11

4.1 アーキテクチャ ……11

4.2 データストアとデバイス記述 ……12

4.3 室内コンテキスト収集……14

4.4 外部コンテキスト収集……14

4.5 LLM 推論実行基盤……15

第 5 章 評価 16

5.1 環境状態同定評価のための観測値データ生成……16

5.2 各センサ観測値の快適域定義 ……16

5.2.1 センサ値ベクトルの網羅生成 ……17

5.2.2 正解ラベルの決定……17

5.3 状態同定精度……18

5.3.1 入力提示条件の構成……18
5.3.2 評価指標と結果……18
5.4 シナリオに基づく動作検証……19
5.4.1 シナリオ設計方針……19
5.5 シナリオ毎の動作検証結果……20

第 6 章 考察 **23**
6.1 状態同定精度に関する考察……23
6.2 シナリオに基づく動作検証に関する考察……24

第 7 章 おわりに **26**

謝辞 **28**

参考文献 **29**

付録 **30**
A.1 プロンプト実装……30
A.2 センサ一覧……42
A.3 アクチュエータ一覧……47

第1章 はじめに

近年，住宅やオフィス，医療施設などの様々な実空間に多数の IoT（Internet of Things）デバイスが導入されつつある．温度，湿度，CO_2濃度，空気質，騒音，照度，窓やドアの開閉状態といった環境情報を取得するセンサと，エアコンや床暖房などの空調機器，照明，電動カーテン，自動窓，空気清浄機，換気扇といったアクチュエータが日常空間に広く配置されるようになっている．これらのデバイスを，ネットワークを介して連携させることで，利用者が個々の機器を逐一操作することなく，室内環境を自動的に調整するスマートホームやスマートビルディングの実現が期待されている．このような環境においては，利用者が細かな設定値や操作手順を指定するのではなく，「部屋を快適にして」「寝る前に過ごしやすい環境にしてほしい」といった直感的かつ曖昧な自然言語指示のみで意図を伝えられることが望ましい．しかしながら，「部屋を快適にして」という一文には，「暑さを解消したい」「乾燥を改善したい」「空気を入れ替えたい」「明るさを調整したい」など，複数の解釈が含まれ得る．したがって，システム側には，利用者が置かれた状況に応じて指示の意味を適切に解釈し，必要な観測と制御を選択する能力が求められる．

一方，現状のスマートホームサービスの多くは，IFTTT に代表されるルールベース制御や，あらかじめ定義したシーンに基づくシーン制御に依存している．これらの方式では，開発者または利用者が事前に条件と動作の組を定義し，生活スタイルや季節変化，デバイス構成の変更に応じてルールを手動で保守する必要がある．また，この種の制御は，温度や湿度など限られた条件に基づく単純な分岐にとどまりやすく，時刻帯，外気条件，居室の利用状況といった広い意味での状況コンテキストを統合的に考慮することが難しい．この課題に対し，近年は大規模言語モデル（Large Language Model ; LLM）の自然言語理解能力と推論能力に着目し，自然言語指示とセンサデータを統合して制御方針を生成する研究が進んでいる．

しかしながら，実空間における IoT サービスでは，センサ観測値を与えれば LLM が自動的に空間状態を正しく理解できるとは限らない．LLM は数値表現の取り扱いに不安定さが残り，温度や湿度，CO_2 濃度のような連続値を入力しても，大小関係や変化量の解釈を誤り，状態推定が正確でない可能性がある．その結果，要求解釈や後段の制御方針が，実際の環境状態と乖離する危険が生じる．

すなわち問題の本質は，数値観測を根拠として状態を構成する過程が LLM 内部で十分に保証されない点にある．

したがって，数値観測から空間状態を安定して認識するための表現設計や推論手順を，推論プロセスに明示的に組み込む必要がある．さらに，IoT サービスは物理世界の機器を制御対象とするため，「冷房 ON と窓開放」や「加湿と除湿の同時実行」といった物理的・論理的に矛盾する操作が生じ得る．これらの競合を適切に検出・回避しなければ，快適性や省エネルギー性を損なうだけでなく，利用者の不信感や機器の無駄な稼働につながる．このような点を踏まえると，曖昧な自然言語指示から一貫して安全な制御計画を生成する枠組みの設計は，スマートホームの信頼性を確保する上で重要な課題である．また，Web of Things（WoT）では，Thing Description（TD）[1]を用いてデバイスのプロパティやアクション，通信エンドポイントを機械可読な形式で記述する標準化が進められている．TD により，メーカーや機器種別の異なるデバイスを共通の記述形式で扱うことが可能となるが，TD を LLM の推論過程に組み込み，状況に応じたセンサ選択からアクチュエータ構成，整合性検証までを一貫して扱う枠組みは十分に検討されていない．

本研究は，事前に条件と動作を固定したルールベース制御において，新規 IoT デバイスの追加やデバイス故障に伴う欠損値の発生といった想定外の状況に対応できない点，ならびにユーザの曖昧な自然言語要求をルール条件へ適切に対応付けることが困難である点を主要な課題として位置付ける．その上で，これらの課題に対し，LLM の自然言語理解力と推論能力を活用して，曖昧な自然言語指示と実世界の状況コンテキストを統合的に解釈し，状況に応じた IoT 連携制御を合成できる LLM ベースの IoT 連携制御方式を確立することを目的とする．具体的には，LLM エージェントと WoT Thing Description を用いて，（i）センサ観測値に基づく環境状態の推定，（ii）状況コンテキストに基づく意図推論，（iii）意図に基づくアクチュエータ選択，（iv）デバイス間の物理的・論理的整合性検証，（v）具体的な制御パラメータ決定を一貫して扱う枠組みを設計し，スマートホームを想定したシナリオにおいてその有用性を示す．本手法の実現にあたり，本研究では以下の二つの課題に取り組む．

[1] https://www.w3.org/TR/wot-thing-description11/

環境状態の推定

曖昧な自然言語指示は複数の解釈を含み得るため，意図決定に必要な状況を特定できない．この不確実性に対処するには，利用可能なセンサ群から観測データを収集した上で，観測値をそのまま入力するのではなく，後続の推論で一貫して参照できる抽象的な状態表現へ正規化し，現在の環境状態を同定する必要がある．特に，実空間の観測値は連続的に変動し得るため，数値列を直接扱う推論では判断の揺らぎが生じ得る．そこで，本研究では，室内環境品質（Indoor Environment Quality：IEQ）の主要因子に基づく状態変数を導入し，観測値を状態ラベルへ写像することで，意図推論に供する環境状態表現を安定化する．

物理的整合性検証

IoT サービスは実空間内のアクチュエータを制御対象とするため，複数機器の同時制御により物理的・論理的な矛盾が生じ得る．合成されたサービスがこのような衝突を含まないよう，自動的に検証し，必要に応じて修正する仕組みが求められる．

本研究では，推論を単一プロンプトで完結させるのではなく，推論を複数段階に分解した段階的推論フローにより LLM エージェントを構成する．具体的には，センサ選択，状況理解の検証，意図推論，アクチュエータ選択，整合性検証，設定決定という段階を明示的に分離し，各段階の判断根拠を中間表現として保持する．これにより，曖昧な自然言語指示に伴う不確実性を段階的に低減するとともに，実空間制御に必要な安全性と説明可能性を確保することを狙う．

本論文は，全 6 章から構成される．第 1 章では，スマートホームにおけるルールベース制御の限界として，新規 IoT デバイスの追加やデバイス故障に伴う欠損値など想定外状況への対応困難性，ならびに自然言語要求との条件照合の困難性を整理し，本研究の目的と貢献を述べた．第 2 章では，LLM を用いた相互運用型 IoT サービス基盤と，実世界センサデータに基づく LLM 推論強化に関する関連研究を概観し，本研究の位置付けと解くべき技術課題を明確化する．第 3 章では，提案エージェントの段階的推論プロセスを示し，環境状態の表現として室内環境品質（IEQ）に基づく状態変数を定義した上で，プロンプト設計方針と各段階における推論方法を述べる．第 4 章では，状況依存型 IoT サービス制御を実現するプロトタイプの設計と実装を述べ，アーキテクチャ，データストアとデバイス記述，室内・外部コンテキスト収集，および LLM 推論実行基盤を示す．

第 5 章では，提案手法の有効性を検証する評価を述べ，環境状態同定評価のための観測値データ生成，状態同定精度の定量評価，およびシナリオに基づく動作検証を行う．第 6 章では，評価結果に基づく考察を行い，第 7 章において，提案手法の有効性と限界，ならびに今後の課題を議論する．

第2章 関連研究

本章では，曖昧な自然言語指示と IoT 環境を統合する研究を対象として，本研究と関連の深い先行研究を整理する．特に，（i）LLM を用いた相互運用型 IoT サービス基盤に関する研究，および（ii）実世界センサデータを用いた LLM 推論能力の強化に関する研究の二つの方向性に着目し，それぞれの到達点と課題を明確にした上で，本研究の位置付けを示す．

2.1 LLM を用いた相互運用型 IoT サービス基盤

まず，相互運用性を重視した LLM 統合 IoT サービス基盤の研究では，異種デバイス／Web サービスを統一インタフェースで扱い，イベント処理を起点として，IoT データをテンプレート化したプロンプトへ変換し，LLM 出力をアクチュエータ実行へ接続する枠組みが提案されている[1]．本研究も，多様なセンサ／アクチュエータを扱うため，デバイス記述や取得データを整理し，LLM が参照可能な形で提示する点で関連する．一方で同研究は，主として「相互運用な実行基盤の設計」に焦点があり，入力はイベントとして与えられることを前提に，LLM の入出力を実行系へ統合する設計である．これに対し本研究は，「曖昧な指示」から環境状態の同定・意図推論を経て制御方針を導出する過程を中核に据える点が異なる．さらにスマートホーム制御では複数機器の同時制御により物理的・論理的競合が生じ得るため，本研究は出力集合の整合性検証と修正を独立ステップとして組み込み，安全かつ一貫した制御設定を生成することを目的としている[2]．

2.2 実世界センサデータに基づく LLM 推論の強化

次に，実世界のセンサデータに基づく LLM 推論能力の強化を目的とする研究では，数値時系列などの IoT データを LLM が扱いやすい表現に変換し，検索拡張（Retrieval-Augmented Generation ; RAG）や推論誘導（CoT）を組み合わせて，分類・推定等の IoT 推論タスクを改善する枠組みが提案されている[3]．本研究も，センサ値に基づく状況理解や推論根拠の明示を重視する点で関連する．しかしながら，同研究の主対象は，センサデータからラベルや状態を推定する「推論性能の向上」であり，推定結果を用いて「どのアクチュエータをどの順序・

どのパラメータで動作させるか」というサービス合成・制御計画の導出や，競合を回避する整合性検証までは主眼としていない．本研究は，曖昧指示に対して制御計画を合成する意思決定問題を扱い，推論を段階化して中間表現を保持しながら不確実性を低減し，最終的に矛盾のない設定へ落とし込む点に特徴がある．

2.3 本研究の位置付け

以上の関連研究を踏まえると，既存研究は，（i）相互運用な IoT 実行基盤の設計，あるいは（ii）センサデータに基づく LLM 推論性能の向上，のいずれかに主眼を置いていることが分かる．一方で，「曖昧な自然言語指示」を起点として，要求解釈に必要な観測の選択から意図推論，アクチュエータ選択，物理的整合性検証，制御パラメータ決定までを一貫して扱う枠組みは十分に検討されていない．本研究は，このギャップに着目し，曖昧な自然言語指示に対して，段階的推論により不確実性を低減しつつ，実空間で安全に実行可能な IoT サービスを動的に合成する LLM エージェントを提案する．この点において，本研究は既存の相互運用基盤研究およびセンサ推論強化研究を補完する位置付けにある．

第3章 エージェントの段階的推論プロセス

3.1 環境状態

本研究の段階的推論プロセスでは，曖昧な自然言語指示から実空間制御に必要な判断を導出するために，推論の核となる「環境状態」を明示的に定義する．自然言語指示は，「部屋を快適にして」のように目的が抽象的であり，そのままでは「何を根拠に」「どの方向へ」制御すべきかが確定しない．また，実空間では温度，湿度，照度など単位やスケールが異なる数値観測が同時に存在し，観測値を生の数値として推論へ投入すると，観測の意味付けが推論途中で混線し，後段の意図推論やアクチュエータ選択が不安定になりやすい．したがって，段階的推論を成立させるためには，(i) 多様な観測値を同一の観点で整理し，(ii) 後続ステップが参照可能な形式で状態を表現し，(iii) 推論の根拠を一貫して追跡可能にする中間表現としての環境状態が必要である．本研究では，環境状態を導入することで，「観測値」から「意味づけされた状態」への写像を推論フローの初期段階として固定し，以降の推論が環境状態の定義に基づいて段階的に進行する構造を実現する．

ここで，本研究における環境状態の枠組みとして用いるのが，Indoor Environmental Quality（IEQ）[2]である．IEQ とは，室内環境が居住者の快適性，健康，および活動性に与える影響を総合的に評価する概念であり，建築環境工学や室内環境評価の分野において広く用いられている．一般に IEQ は，温熱環境，室内空気質，照明・視環境，音環境の 4 つの要素から構成され，住宅やオフィス等の一般的な室内空間における評価軸として整理されている．本研究では，IEQ をセンサによって観測される多様な物理量を整理するための中間表現の枠組みとして位置付ける．すなわち，観測値を IEQ の 4 領域へ対応付けた上で，それぞれを low，middle，high の 3 値ラベルへ写像し，このラベル集合を環境状態として定義する．この環境状態定義により，後続の意図推論およびアクチュエータ選択は，「数値そのもの」ではなく「数値から解釈された状態」を参照して推論を行えるため，推論の前提が明確化される．

[2] https://build-up.ec.europa.eu/en/resources-and-tools/articles/acceptable-indoor-environmental-quality-and-energy-efficiency

さらに，環境状態は推論フロー全体にわたり引き継がれるため，「どの観測値を根拠として，どの領域がどのラベルに写像されたか」を一貫して追跡できる．その結果，本研究の段階的推論プロセスは，曖昧な自然言語指示に伴う不確実性を段階的に低減しつつ，実空間制御に求められる説明可能性と安全性を両立する基盤として機能する．

3.2 プロセスの概要

本研究が対象とするスマートホーム環境では，室内状態だけでなく，時刻や外気条件といった外部要因が制御方針に影響し，さらに複数アクチュエータの同時関与によって物理的・論理的な衝突が生じ得る．このような状況において，「部屋を快適にして」のような曖昧な自然言語指示を，指示文のみから直接制御パラメータへ写像すると，(i) 意図推論が未観測または未整理の状態に依存して不安定化すること，(ii) デバイス間の排他関係や物理干渉を見落とすこと，の双方が起こり得る．そこで本研究では，「状態の確定」「意図の具体化」「手段の選択」「安全性の担保」を明確に分離し，段階的に接続する推論プロセスを設計する．提案プロセスの概要を図 1 に示す．プロセスは，ユーザによる曖昧な自然言語による要求を受けて，環境状態の同定，意図推論，アクチュエータ選択，整合性検証，制御パラメータ決定という複数段階に推論を分解し，各段階で参照する情報を明示的に限定しながら逐次実行する．このように，単一の推論で制御計画を一括生成するのではなく，段階ごとに役割を明確に分離したプロンプトを用いることで，判断過程の可視化と推論の混線抑制を実現する．

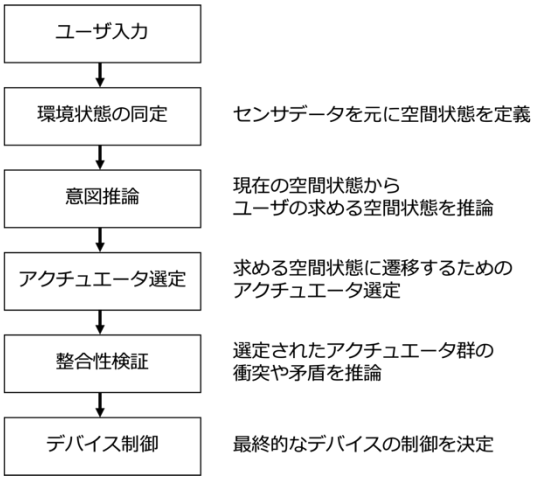

図 1：本研究における段階的推論プロセス

3.3 プロンプト設計方針

本研究では，大規模言語モデルに対して段階的な推論過程を誘導するため，CoT の考え方を参考にプロンプト設計を行った．本研究が対象とする課題は，曖昧な自然言語指示を，実世界の観測値やデバイス制約を考慮しながら制御計画へ変換することであり，単一の推論過程では，推論の混線や根拠不明瞭化が生じやすい．そこで，本研究では，CoT の考え方に基づき，推論過程を環境状態の同定，意図推論，アクチュエータ選択，整合性検証，制御パラメータ決定という複数の役割を持つステップへ明示的に分離した．各ステップでは，当該判断に必要な情報のみを入力として与え，それ以外の情報は参照できないように制限している．このように推論を段階化することで，単一プロンプトで一括生成する場合に生じやすい，（i）複数判断基準の混在による不安定化，（ii）どの情報を根拠として判断したのかが不明確になる問題を抑制する．また，各ステップの出力を次段の入力として逐次接続することで，判断過程を追跡可能な形で保持でき，後述する評価や分析においても，各判断の妥当性を段階ごとに検証可能となる．

一方で，CoT における詳細な思考過程の文章出力は，冗長化や出力の揺らぎを招くことが指摘されている．特に，本研究のように複数ステップを連鎖させる場合，非構造的な自然言語出力は後続処理の不安定化を引き起こす．そこで本研究では，内部では段階的な推論を促しつつ，外部出力は各ステップの決定結果のみを構造化形式で返すという方針を採用した．具体的には，各ステップの出力を JSON 形式に限定し，「選択したセンサ集合」「IEQ に基づく環境ラベル」「推定された意図」「選択されたアクチュエータ ID」「最終的な制御パラメータ」といった決定結果のみを明示的に出力させている．この設計により，後続ステップへの情報受け渡しが安定化し，段階的推論全体をプログラム的に制御・検証することが可能となる．以上の設計方針により，本研究は CoT の考え方を活かしつつ，実世界制御に適した安定性・再現性・説明可能性を備えた段階的推論プロセスを実現する．

3.4 プロンプトの実装

3.4.1 環境状態の同定

本ステップの目的は，室内環境を IEQ の 4 領域（温熱環境，室内空気質，照明・視環境，音環境）として評価し，各領域を High／Middle／Low の 3 値ラベルへ写像することである．入力として，センサ観測値とそのメタデータを与え明

示された情報のみを根拠に判断させる．出力として，温熱環境，室内空気質，照明・視環境，音環境の 4 領域に対して 3 値ラベルを返す．

3.4.2 意図推論

本ステップの目的は，ユーザの曖昧な自然言語指示を，IEQ の 4 領域に対する目標環境状態として定式化することである．入力として，ユーザ要求文と，前段で推定した現在の環境状態ラベルを与える．出力として，温熱環境，室内空気質，照明・視環境，音環境の 4 領域に対して，目標ラベルを返す．これにより，次ステップのアクチュエータ選択は「現在環境状態と目標環境状態の差分」を参照しながら制御候補を決定できる．

3.4.3 アクチュエータ選択

本ステップの目的は，目標環境状態を実現するために必要なアクチュエータを，利用可能なデバイス集合から選択することである．入力として，目標環境ラベル，現在環境ラベル，外部コンテキスト，およびアクチュエータリストとそのメタデータを与える．出力として選択されたアクチュエータを返す．

3.4.4 デバイス間整合性検証

本ステップの目的は，選択された複数アクチュエータを同時に動作させる際に生じ得る論理的・物理的な競合を検出し，後段の設定生成における制約として提供することである．入力として，選択したアクチュエータとそのメタデータを与える．出力として，アクチュエータ間で起こり得る競合を検出し，返す．この結果は次段の制御パラメータ決定に引き継がれ，競合回避のための設定調整に利用される．

3.4.5 デバイス設定の決定

本ステップの目的は，現在ラベルと理想ラベルの差分に基づいて目標環境状態を具体化し，選択アクチュエータの操作パラメータを決定することである．入力として，現在環境ラベル，理想環境ラベル，外部コンテキスト（時刻・天気），選択アクチュエータの詳細，および整合性検証結果を与え，整合性検証で競合が報告された場合には，その内容を制約として参照し，競合を回避するようにパラメータを調整する．出力はとして， 目標とする温度，湿度，CO_2，CO，照度，騒音を返し，目標を達成するためのアクチュエータのパラメータ設定を返す．

第4章 状況依存型 IoT サービス制御のプロトタイプ

4.1 アーキテクチャ

本研究で提案する LLM エージェントのシステム構成を図 2 に示す．本システムは，(1) センサデータ取得，(2) 推論システム，(3) デバイス実行の 3 要素から構成される．ユーザは曖昧な自然言語指示を入力し，その入力を起点としてセンサデータ取得部が室内および外部の状況情報を収集する．収集された情報は推論システム部へ渡され，段階的推論フローに従って制御対象デバイスおよび制御パラメータからなる制御計画が生成される．生成された制御計画に基づき，デバイス実行部が対象デバイスの操作を行う．

本研究が対象とするスマートホーム環境では，室内状態や外部環境が刻々と変化し，複数のアクチュエータが同時に関与し得る．したがって，事前に定義された固定ルールに基づくのではなく，ユーザ要求が与えられた時点の状況を観測し，状況依存で制御計画を生成する構成が必要である．本システムはこの要件を満たすため，観測値，デバイス機能情報，外部コンテキストを一貫した形式で収集し，LLM が参照可能な入力として整形した上で段階的推論を実行する設計を採用する．

また，本研究では，状況情報の収集と推論実行を実環境で継続運用可能な形で実装し，デバイス追加や観測頻度の変動に追従できる構成を要する．そのため，クラウド基盤として AWS を採用し，マネージドサービスを用いて運用負荷を抑制しつつ, 信頼性, セキュリティ, 性能効率などの設計原則に沿った構成をとる．ただし，本研究で実装したプロトタイプでは，センサ観測値の取得を実デバイスからは行っていない．代わりに，作成した室内環境データセットに含まれる観測値を，「取得済みデータ」として 後述の AWS DynamoDB に事前投入することで，推論時には DynamoDB を参照先として扱う構成を再現した．これにより，推論プロセス（IEQ 推定，意図推定，アクチュエータ選択，整合性検証，制御パラメータ生成）は，運用時と同様に DynamoDB 上のコンテキストを参照して実行される．一方で，アクチュエータ制御については実機へのコマンド送信・動作確認は行っておらず，本稿で示す制御結果は，デバイス定義に基づいて生成された制御パラメータとして出力する段階までを対象とする．

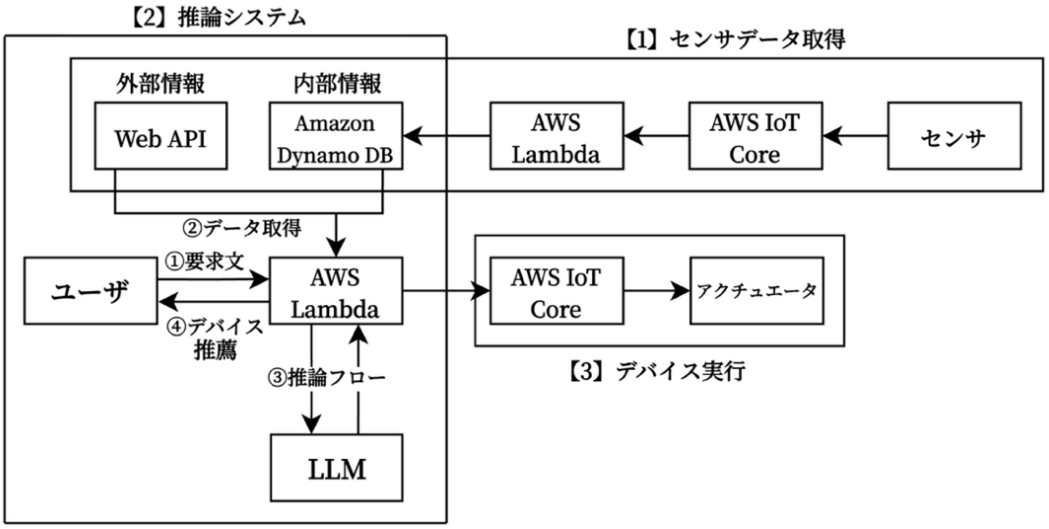

図 2：システムアーキテクチャ

4.2 データストアとデバイス記述

本システムでは，室内コンテキストおよびデバイス情報を管理するデータストアとして Amazon DynamoDB[3]を用いる．DynamoDB はフルマネージドな NoSQL データベースとして提供されており，データ量やアクセス頻度の変動に対して運用負荷を抑えた拡張が可能である．室内のセンサ観測値は AWS IoT Core を介して収集され，IoT Core のルールアクションにより DynamoDB へ書き込まれる．IoT Core は MQTT メッセージの全部または一部を DynamoDB テーブルに書き込むアクションを提供している．これにより，センサ観測値をクラウド側へ集約し，推論要求が発生した時点で一貫した経路で参照可能となる．推論時には DynamoDB から，少なくとも以下の三種類の情報を取得する設計とする．本文の記述を簡潔化するため，表 1 に整理する．

IoT デバイスはメーカーや機器種別によって仕様が異なり，観測対象，単位，制御可能な操作，パラメータ制約などが統一されていない．この差異をそのまま扱うと，LLM に提示される情報が不統一となり，誤解釈や推論の揺らぎを招く．そこで本研究では，Web of Things（WoT）の Thing Description（TD）に基づく JSON 形式を用いてデバイス仕様を統一的に記述する．TD はデバイスのメタデ

[3] https://docs.aws.amazon.com/iot/latest/developerguide/dynamodb-rule-action.html

ータとインタフェースを機械可読に記述する標準であり，異種デバイスを共通形式で扱うための基盤となる．具体的には，センサについては図 3 のように観測可能なプロパティを properties として，アクチュエータについては図 4 のように実行可能な操作と引数を actions として記述する．また，推論過程における安定した参照を実現するため，デバイス識別には表示名ではなく一意な ID を用いる. これにより, 人間にとっての可読性と機械処理における一貫性を両立する.

表 1：提案システムが DynamoDB から取得するデータ種別と役割

種別	内容	目的
Sensor List	利用可能なセンサ一覧と機能記述	観測値の意味づけと参照単位の統一
Sensor Data	各センサの観測値	推論時点での状況推定
Actuator List	利用可能なアクチュエータ一覧と機能記述	制御可能操作の制約下での計画生成

```
{
  "@context": "https://www.w3.org/2022/wot/td/v1.1",
  "id": "urn:dev:ops:room-temperature-sensor-001",
  "title": "室温度センサ",
  "description": "室内温度を測定するセンサ",
  "security": ["nosec_sec"],
  "securityDefinitions": {
    "nosec_sec": { "scheme": "nosec" }
  },
  "properties": {
    "temperature": {
      "title": "室内温度",
      "description": "室内の温度（°C）",
      "type": "number",
      "unit": "degree celsius",
      "readOnly": true,
      "observable": true,
      "forms": [
        {
          "href": "https://example.com/temperature/properties/temperature",
          "contentType": "application/json",
          "op": ["readproperty"]
        }
      ]
    }
  }
}
```

図 3：WoT TD に基づくセンサ記述例

```
{
  "@context": "https://www.w3.org/2022/wot/td/v1.1",
  "id": "urn:dev:ops:air-conditioner-001",
  "title": "空気調和機",
  "description": "冷暖房および送風量を制御する装置",
  "security": ["nosec_sc"],
  "securityDefinitions": {
    "nosec_sc": { "scheme": "nosec" }
  },
  "actions": {
    "setMode": {
      "title": "モード設定",
      "description": "冷房または暖房モードを切り替える",
      "input": {
        "type": "string",
        "enum": ["cool", "heat"]
      },
      "forms": [
        {
          "href":
"https://example.com/aircon/actions/setMode",
          "contentType": "application/json",
          "op": ["invokeaction"]
        }
      ]
    },
    "setFanLevel": {
      "title": "風量設定",
      "description": "送風の強さを設定する",
      "input": {
        "type": "string",
        "enum": ["low", "medium", "high", "auto"]
      },
      "forms": [
        {
          "href":
"https://example.com/aircon/actions/setFanLevel",
          "contentType": "application/json",
          "op": ["invokeaction"]
        }
      ]
    }
  }
}
```

図 4：WoT TD に基づくアクチュエータ記述例

4.3 室内コンテキスト収集

本研究の段階的推論では，単なる数値としての観測値だけでなく，観測値の意味付け（観測対象, 単位, 解釈の前提）を同時に参照できることが重要である．そこで本システムでは，観測値に加えて Sensor List から取得したメタデータを束ねたセンサ束を生成し，環境理解および意図推論の段階で参照する設計とする．これにより，LLM は TD に基づいて，センサ値の意味を説明可能な形で扱うことができる．

4.4 外部コンテキスト収集

曖昧指示の解釈には，室内観測値だけでなく，時刻や外気条件といった外部要因が影響する．本研究における外部コンテキストとは，室内センサでは直接取得できないが，制御方針の決定に影響を与える情報を指す．具体的には，時刻情報および気象情報である．例えば，同じ室温であっても，夜間か日中か，外気温が高いか低いかによって，換気の可否や冷暖房の優先度など，アクチュエータの望ましい動作は変化し得る．そこで本システムでは，外部コンテキストとして時

刻情報と気象情報を取得し，アクチュエータ選択および制御決定の推論材料として統合する．時刻は World Time API[4]から取得し，気象は Open-Meteo の Weather Forecast API[5]から取得する．これにより，室内観測値のみでは判別しにくい制御方針の分岐条件を，外部要因として明示的に推論入力へ含めることが可能になる．

4.5 LLM 推論実行基盤

LLM による推論システムは AWS Lambda 上で動作し，段階的推論フローを実行する．推論処理の構成には LangChain を用いる．LangChain は，大規模言語モデルを用いた推論処理をパイプラインとして構成・管理するためのフレームワークであり，本研究では複数の推論ステップを順序立てて実行するために利用する．本研究では，単一プロンプトで推論を完結させるのではなく，推論を複数段階へ分解する段階的推論フローを採用し，各段階が必要最小限の情報を入力として受け取り，決定結果と根拠を出力する設計とする．これにより，推論過程を段階ごとに追跡可能とし，後段が前段の出力に基づいて推論する関係を明確化する．

[4] https://worldtimeapi.org/
[5] https://open-meteo.com/

第5章 評価

5.1 環境状態同定評価のための観測値データ生成

本研究の段階的推論プロセスにおける，環境状態の同定は，後続ステップが参照する環境状態（中間表現）を確定させる役割を担う．この段階でセンサ観測値の意味（観測対象，単位）に関する誤解が生じると，誤った前提が意図推論や制御計画生成へ伝播し，推論全体の安定性・安全性・説明可能性を損なう．したがって，本節では，環境状態の推定を後段から切り離し，環境状態同定を独立に評価するための観測値データを，再現可能な手続きとして生成し，評価用データとして扱う．

本評価では，この評価を多クラス分類問題として定式化する．入力は，温度，湿度，照度，騒音，CO_2，CO の 6 種センサ観測値と，必要に応じて観測対象・単位等を含むメタデータである．出力は IEQ の 4 領域（温熱，空気質，照明・視環境，音環境）について，それぞれ low／middle／high の 3 値ラベルである．

5.2 各センサ観測値の快適域定義

観測値データ生成の第一段階として，評価で扱う 6 種センサ（温度，湿度，照度，騒音，CO_2，CO）の各観測値を，low／middle／high の離散ラベルへ割り当てるためのレンジを定義する．ここでのラベルは快適性の良否を表すものとして，環境状態同定において，異種の数値観測を後段推論で参照可能な抽象表現へ正規化するための基準である．基本方針として，既存の基準・指針における「基準適合域」を middle，それより高い側を high，低い側を low として区分する．

温度および湿度については，建築物環境衛生管理基準[6]に基づき，室内温度を 18°C以上 28°C以下，相対湿度を 40％以上 70％以下の範囲として扱い，この範囲を middle とする． これより高温側または高湿側を high，低温側または低湿側を low としてレンジを構成する．

一酸化炭素（CO）については，「middle（適正域）」に相当する中間域が基準として明確に与えられないため，本研究では low／high の 2 値で定義する．具体的には，建築物環境衛生管理基準における CO の基準値（6 ppm 以下）を境界として，基準以下を low，基準超過を high とする．

[6] https://www.mhlw.go.jp/bunya/kenkou/seikatsu-eisei10/

CO_2は，室内空気質評価の指標[7]として広く用いられる濃度区分（1,000 ppm 以下，1,000–2,000 ppm，2,000 ppm 以上）を参照し，1,000–2,000 ppm を middle，1000 未満を low，2000 以上を high として定義する．

照度については，居室における一般的な活動を想定した代表レンジとして，おおむね 50 lx（全般照明の下限側）から 200 lx（団らん等の上限側）を middle の参照域として設定し，それより低い側を low，高い側を high とする．この値域は，IES の照度選定に基づく解説記事[8]に示される居室用途の目安に従う．

騒音については，睡眠・休息への影響を扱う WHO の騒音に関するファクトシート[9]に基づき，30–40 dB を middle の参照域として設定する.30 未満を low，40db 以上を high として区分する．

5.2.1 センサ値ベクトルの網羅生成

第 2 段階として，定義した 3 値（low／middle／high）の組合せを網羅的に列挙し，評価用の 6 次元センサ値ベクトルを生成する．温度・湿度・照度・騒音・CO_2の 5 種はそれぞれ 3 値を取り得るため 3^5 通りとなる．一方，CO については middle に相当する快適域を一意に定義できない前提に立ち，2 値（low／high）のみを用いる．この結果，全組合せは $3^5 \times 2 = 486$ パターンとなる．各パターンに対して，定義済みレンジに従いレンジ内の基準値を割り当て，6 次元の観測値ベクトルとして構成する．

5.2.2 正解ラベルの決定

生成した各入力に対し，上位の LLM を用いて IEQ で定義される 4 つ状態変数（温熱，空気質，照明・視環境，音環境）のラベル推定を実行する．推定は各ベクトルにつき 3 回繰り返し，多数を採用することで，領域ごとの正解ラベルを決定する．このように評価対象と異なるモデルを用いて擬似正解を構成することで，評価対象の出力揺らぎを正解側へ混入させない形で，分類性能を測定可能とする．ここで作成したデータを正解データとして，評価を行う．

---

[7] https://www.mhlw.go.jp/content/11130500/000771220.pdf
[8] https://www.elementzcorp.com/determining-adequate-lighting-levels-a-guide-based-on-the-ies-illuminance-selector-
[9] https://www.who.int/europe/news-room/fact-sheets

5.3 状態同定精度

5.3.1 入力提示条件の構成

第 3 段階として，上記で作成したセンサ値ベクトルに対して，LLM への入力提示条件を 2 通り作成する．

- WoT TD によるメタデータ記述あり：観測値に加えて，観測対象・単位等のメタデータを WoT TD に基づいて併記する．
- WoT TD によるメタデータ記述なし：観測値のみを提示し，メタデータを与えない．

この 2 条件の比較により，メタデータ付与が環境状態同定精度に与える影響を定量化する．

5.3.2 評価指標と結果

本研究では，提案手法による環境状態同定の性能を評価するため，温熱環境（Thermal），空気質（Air_quality），照明・視環境（Lighting），音環境（Acoustics）の各属性に対する 3 値分類（low／middle／high）の精度を測定した．評価指標には適合率，再現率，F1 値用いた．また，WoT TD によるメタデータ記述の有無が，状態同定性能（Precision／Recall／F1）に与える影響を表 2 に，WoT TD によるメタデータ記述の有無によって，どの状態変数で誤りが減少したかを変数別に可視化したもの表 3 に示す．

表 2：WoT TD によるメタデータ記述の有無による精度比較

	適合率	再現率	F1
WoT TD メタデータなし	0.6238	0.6183	0.6209
WoT TD メタデータあり	0.6362	0.6280	0.6319
差分	+0.0124	+0.0097	+0.0110
向上率	+1.99%	+1.57%	+1.77%

表 3：WoT TD による記述の有無による状態変数別の誤り件数

	WoT TD なし	WoT TD あり	差分
Thermal	54 件	41 件	−13
Air_quality	188 件	180 件	−8
Lighting	114 件	110 件	−4
Acoustics	277 件	271 件	−6

5.4 シナリオに基づく動作検証

本節では，前節までに評価した状態同定精度に加え，提案手法が実環境を想定した文脈下で妥当な動作を行うかを確認するため，シナリオに基づく動作検証を行う．数値的な分類精度のみでは，外部環境と室内環境が組み合わさった状況において，システムが一貫した判断や制御方針を生成できるかを十分に評価することはできない．そこで本研究では，外部コンテキスト（気象条件）と内部コンテキスト(室内センサ状態)を明示的に組み合わせたシナリオテストを設計し，プロトタイプの振る舞いを定性的に検証する．

5.4.1 シナリオ設計方針

本研究では，提案手法が外部環境および内部環境という複数の文脈情報を統合的に解釈し，ユーザの曖昧な要求に応じた適切な動作を生成できるかを検証するため，シナリオに基づく動作検証を行った．本節では，そのシナリオ設計方針について詳述する．

まず, 外部コンテキストについては, 実環境への適用可能性を担保するため，架空の数値や理想化された条件ではなく，実在する気象観測データを用いる方針を採用し, 表 4 に示す. 具体的には, 気象庁が公開する観測データ[10]を参照し，滋賀県彦根市における冬季および夏季の正午（12 時）の気象条件を外部コンテキストとして採用した．正午を選択した理由は，日射量や外気温が比較的安定しており，季節差が明確に現れる時刻であるためである．これにより，提案手法が季節的な外部環境差を考慮した判断を行えるかを検証可能とした．

次に，内部コンテキストについて，本研究で構築した室内環境データセットから，室内が高温であるパターンと低温であるパターンの 2 パターン選定した

[10] https://www.data.jma.go.jp/stats/etrn/

表 4：採用した外部コンテキスト情報

区分	日時	地点	外気温 (°C)	外気湿度 (%)	天気	風速 (km/h)	降水量 (mm)
冬季外部コンテキスト	2026-01-17 12:00	彦根	7.2	82	晴れ	2.4	0.0
夏季外部コンテキスト	2025-08-17 12:00	彦根	32.7	66	晴れ	2.5	0.0

表 5：採用した内部コンテキスト情報

パターン ID	室温 (°C)	室内湿度 (%)	CO_2 (ppm)	CO (ppm)	照度 (lx)	騒音 (dB)
484	29.29	77.1	2977	6.94	358	27.9
2	17.56	30.4	718	0.77	40	34.5

ものを表 5 に示す．これにより，内部環境の状態が快適性に与える影響を外部コンテキストとの組合せによる振る舞いの違いとして観察可能とした．

以上で定義した外部コンテキストと内部コンテキストを組み合わせることで，以下に示す 4 つのシナリオを構成した．

- **シナリオ 1**：冬季外部コンテキスト × 低温内部コンテキスト
- **シナリオ 2**：冬季外部コンテキスト × 高温内部コンテキスト
- **シナリオ 3**：夏季外部コンテキスト × 高温内部コンテキスト
- **シナリオ 4**：夏季外部コンテキスト × 低温内部コンテキスト

このように，外部環境と内部環境の組合せを網羅的に設計することで，同一のユーザ要求に対しても，文脈の違いに応じて異なる環境解釈・目標状態・制御方針が導出されるかを評価できる．

5.5 シナリオ毎の動作検証結果

今回のシナリオテストでは，ユーザの曖昧な要求として，「部屋を快適にして」を一貫して用いる．

以下に，段階的推論の各結果を表 6，表 7，表 8 に示す.

表 6：推論プロセスにおける環境状態の同定と意図推論結果

シナリオ	現在の環境状態 (Thermal/ IAQ/ Light/ Acoustic)	正解環境状態 (Thermal/ IAQ/ Light/ Acoustic)	目標環境状態 (Thermal/ IAQ/ Light/ Acoustic)
シナリオ1	Low/Low/Low/Low	Low/Low/Low/Middle	Middle/Low/Middle/Low
シナリオ2	High/High/Middle/Low	High/High/High/Low	Middle/Low/Middle/Middle
シナリオ3	High/High/Middle/Low	High/High/High/Low	Middle/Low/ Middle/Low
シナリオ4	Low/Middle/Low/Low	Low/Low/Low/Middle	Middle/Low/ Middle/Low

表 7：推論プロセスにおけるアクチュエータ選択と整合性検証結果

シナリオ	選択アクチュエータ	アクチュエータの競合
シナリオ1	・空気調和機 ・照明	
シナリオ2	・空気調和機 ・換気扇 ・自動開閉窓 ・空気清浄機 ・スピーカー	(1)空気調和機作動・窓の開放 (2)空気清浄機作動・窓の開放 (3)空気清浄機作動・換気扇作動
シナリオ3	・空気調和機 ・扇風機 ・空気清浄機 ・換気扇	(1)空気清浄機作動・換気扇作動
シナリオ4	・自動開閉窓 ・空気清浄機 ・照明	(1) 窓の開放・空気清浄機作動

表 8：推論プロセスにおけるデバイス制御結果

シナリオ	目標値 （温度，湿度， CO_2, CO， 照度，騒音）	最終制御パラメータ
シナリオ**1**	(21,45, 600,0.5, 300,40)	空気調和機：モード設定＝heat，風量＝high 照明：明るさ＝50
シナリオ**2**	(22.0, 45,500, 0,300,40)	空気調和機：モード設定＝cool，風量＝high 自動開閉窓：開閉状態＝close 空気清浄機：モード設定＝auto スピーカー：音量＝10
シナリオ**3**	(25,50, 600,0.1, 300,30)	空気調和機：モード＝cool，風量＝high 扇風機：速度＝3 空気清浄機：モード＝high
シナリオ**4**	(24,50, 600,0.5, 300,30)	自動開閉窓：開閉状態＝close 空気清浄機：モード＝high 照明：明るさ＝50

第6章 考察

6.1 状態同定精度に関する考察

本研究では，段階的推論プロセスの初期段階に位置付けられる環境状態同定を多クラス分類問題として定式化し，WoT TD によるメタデータ付与の有無が環境状態の同定精度に与える影響を評価した．その結果，適合率，再現率，F1 値のすべてにおいて，WoT TD によりメタデータを付与した条件で一貫した性能向上が確認された．具体的には，WoT TD によるメタデータなしの場合と比較して，適合率は 1.99%，再現率は 1.57%，F1 値は 1.77%向上した．さらに，状態変数別の誤り件数においても，Thermal，Air_quality，Lighting，Acoystics のすべての領域で誤分類数が減少している．また，状態変数別の誤り件数に着目すると，WoT TD によるメタデータを付与した条件では，温熱環境，空気質，照明・視環境，音環境のすべての領域において誤り件数が減少していることが確認できる．

これらの結果は，WoT TD によるメタデータが，環境状態同定において有効な補助情報として機能したことを示唆する．本研究の環境状態同定は，異種センサの数値観測値を IEQ が定義する 4 つの状態変数へ対応付け，さらに，low／middle／high という抽象ラベルへ正規化する処理である．この過程では，連続値である数値を状態境界に基づき安定して解釈することが難しい可能性がある．このような状況において，何を計測しているかのような観測対象や単位といった意味的制約を WoT TD により明示することで，LLM が観測値を状態変数へ写像する際の解釈の揺らぎが抑制され，分類境界付近に存在する観測値に対する判断の安定性が高まったと解釈できる．

以上より，WoT TD に基づくメタデータ付与は，LLM の数値理解を補助し，観測値から状態変数を決定する，本研究における環境状態表現の頑健性を高める効果を有することが示された．これは，後段推論が参照する中間表現としての環境状態表現を安定に構築するという本研究の目的に対して，重要な知見である．

6.2 シナリオに基づく動作検証に関する考察

数値的な分類精度の評価に加え，本研究では，外部コンテキストと内部コンテキストを組み合わせたシナリオテストを通じて，提案手法が文脈依存的に一貫した判断および制御方針を生成できるかを定性的に検証した．本節では，その結果について考察する．

まず，表 6 に示される環境状態同定（現在の環境状態）および意図推論（目標環境状態）の結果から，提案手法は，ユーザ入力「部屋を快適にして」という曖昧要求に対して，温熱環境を Middle に近づける，空気質は Low（汚染負荷が低い状態）を志向する，照明・視環境を Middle に近づけるという一貫した方針で目標状態を生成できていることが分かる．特に，シナリオ 1 およびシナリオ 3 といった対照的な条件でも，目標環境状態が Middle/Low/Middle/Low に収束しており，条件が異なっていても快適な環境状態へ遷移することが確認できる．

また，意図推論への影響という観点では，状態同定に誤差が含まれていても，目標は多くのシナリオで同様の快適化方向に定まっているため，少なくとも本表に示す範囲では，目標生成が大きく破綻していない．ただし，シナリオ 2 では目標の音環境が Middle となっており，現在（Low）および正解（Low）から敢えて変化させる方向が選択されている点に対して，これは，音環境における快適さを生活上の標準状態(Middle)へ寄せるよう一般化して解釈した結果と考えられる．しかしながら，実際には静かな状態(Low)を快適とみなす利用者も多いため，音環境の目標設定には，静穏志向か否かといったユーザ嗜好を導入する余地があると考えられる．以上より，提案手法はユーザの曖昧な自然言語要求に対して一貫した目標状態を生成できることが確認された．一方で，静穏志向の有無といったユーザ嗜好を導入し，状況だけでなく個人差を反映できる枠組みへ拡張することが課題として挙げられる．

次に，表 7 に示されるアクチュエータ選択結果に着目すると，環境状態同定および意図推論の出力に基づき，利用可能なデバイス群から論理的に妥当な組合せが選択されていることが分かる．同時に，空気調和機と自動開閉窓，空気清浄機と換気扇といった機能的に競合し得る組合せが検出されている．この点は，本研究の推論プロセスが，単にアクチュエータを列挙するのではなく，同時実行時の競合や相反のような制御の整合性を明示的に扱っていることを示す結果である．特にシナリオ 2 では，空気調和機・換気扇・自動開閉窓・空気清浄機・スピーカーが選択され，競合として（i）空気調和機作動と窓の開放，（ii）空気清

浄機作動と窓の開放，（iii）空気清浄機作動と換気扇作動，が検出されている．これらはいずれも，空気調和機による温熱制御，空気清浄機による室内循環浄化のような閉空間を前提とする制御と，外気導入・排気を伴う制御が同時に選択されることで生じる，効率低下や作用の相殺・冗長化に相当する．すなわち提案手法は，快適化の達成に向けて空気質改善手段を複数選択し得る一方で，その同時実行が目的に対して必ずしも整合的ではないことを競合として顕在化できていることがわかる．

さらに，表 8 に示されるデバイス制御結果から，提案手法は目標値を中間表現として設定した上で，利用可能なアクチュエータの操作パラメータ（モード設定，風量，明るさ等）へ具体的な制御に写像できていることが分かる．すなわち，本研究の推論プロセスは，環境状態同定・意図推論の結果を受けて，最終的にどのデバイスを，どの設定値で動かすかという実行可能な出力へ落とし込めていることを表している．特に，シナリオ 2 では，空気調和機の制御として，モード設定を「冷房」，風量を「強」に加え，自動開閉窓の開閉状態を「閉」，空気清浄機を「自動」，スピーカーを音量「10」としている．ここで窓が「閉」に設定されている点は重要であり，表 7 で検出された「空気調和機作動・窓開放」と「空気清浄機作動と窓・換気扇の作動」といった競合可能性に対して，競合を考慮した制御であること，また，最終制御が閉空間前提の手段を優先する方針を反映した結果として解釈できる．さらに，窓を閉めるという動作は，外部環境を踏まえ，外部の状況が室内へ流入して環境状態の悪化や制御方針の競合を招くことを回避するために，外部影響を遮断する処置として選択されたと考えられる．この点においては，外部環境が室内状態へ与える影響を見込み，その流入を遮断して室内環境の安定性および制御の整合性を維持する方針が選択されているため，外部環境を考慮した制御であると解釈できる．ただし，これは外部条件に応じて制御方針を能動的に分岐させる「適応」とは異なり，外部影響を抑制する「遮断」によるものである．すなわち，競合検出が単なる診断情報に留まらず，後段の制御パラメータ選択に影響していることを示している．

以上の結果から，本研究で提案した段階的推論手法は，数値的な環境状態同定の精度のみならず，実環境を想定した複合的な文脈下にプロトタイプのシナリオテストにおいても，ユーザ要求に対して一貫した環境解釈および制御方針を生成できることが確認された．これは，状況依存型 IoT サービスにおける LLM エージェントの実用性を評価する上で，有用性を示すものである．

第7章 おわりに

本研究では，ユーザの自然言語による曖昧な要求に対して，利用可能な IoT センサ観測値を総合的に解釈し，実世界の状況を推定した上で，状況に応じた IoT サービスを合成する LLM エージェントを提案した．従来のスマートホームで主流であるルールベース制御が，新規 IoT デバイス追加やデバイス故障に伴う欠損値などの想定外状況へ対応しにくい点，および自然言語要求との条件照合が困難である点を踏まえ，本研究では，環境状態同定，意図推論，アクチュエータ選択，デバイス間の競合検証，デバイス制御という段階的推論フローを構築した．提案手法の特徴は 2 点あり，1 つは，推論過程を複数段階に分解し，中間表現を明示することで，曖昧要求に対する判断根拠を整理可能にした点である．また，2 つ目として，デバイス記述に WoT（Web of Things）Thing Description（TD）による記述方式を用いることで，センサ名，単位，観測対象といったメタデータを LLM 推論へ入力できる構成とした点である．評価では，環境状態同定を IEQ に基づく 4 領域それぞれの 3 値分類として定式化し，WoT TD によるメタデータ付与の有無で性能比較を行った．その結果，メタデータ付与ありの条件が，適合率・再現率・F1 の各指標で一貫して上回り，誤り件数も全領域で減少した．

さらに，シナリオテストにより，提案した段階的推論手法は，環境状態同定の精度にとどまらず，ユーザ要求に対して一貫した環境解釈および制御方針を生成できることが確認された．一方で，本研究のプロトタイプは，推論時に DynamoDB から各種情報を取得する設計としつつも，現時点では実アクチュエータ制御を行わず，推論結果として制御パラメータを出力する段階に留まっている．したがって，今後は，設計したアーキテクチャに基づき，実センサ値の取得から推論，アクチュエータ実行までを一貫して接続した実運用下で評価を行う必要がある．加えて，環境条件やデバイス構成の多様性，通信遅延やセンサ欠損といった実環境特有の外乱，およびユーザ嗜好の差異を考慮しつつ，安全性・整合性を担保した制御へ発展させることが課題である．また，本研究では、環境状態推定において利用可能なセンサを固定的に用いており，ユーザ意図や状況に応じたセンサ選定機構は導入していない．

一方で、実環境ではセンサの取得コスト，ノイズ特性，欠損，冗長性が存在し，すべてのセンサを常時用いる設計が最適とは限らない．したがって今後は，ユー

ザの要求やタスクに応じて，状態推定に寄与するセンサ集合を決定するセンサ選定機構を実装し，その有効性を検証することは課題として挙げられる．

以上より，本研究は，状況依存型 IoT サービスにおける LLM エージェントを段階的推論として体系化し，メタデータに基づく環境状態同定の安定化と，文脈に応じた制御方針生成の可能性を示した．

謝辞

本研究を行うにあたり，熱心なご指導，ご助言を賜りました村上陽平教授に深く感謝申し上げます．また，普段からお世話になっている社会知能研究室の皆様にも心より感謝申し上げます．

参考文献

[1] Yokotsuji, R., Lin, D., Uwano, F.: LLM-based Interoperable IoT Service Platform, Proceedings of the 2024 IEEE/WIC International Conference on Web Intelligence and Intelligent Agent Technology (WI-IAT), pp. 1–8 (2024).

[2] Huang, B., Chaki, D., Bouguettaya, A., Lam, K.-Y.: A Survey on Conflict Detection in IoT-based Smart Homes, ACM Computing Surveys, 56, 5, Article 122, 40 pages (2023).

[3] An, T., Zhou, Y., Zou, H., Yang, J.: IoT-LLM: A Framework for Enhancing Large Language Model Reasoning from Real-world Sensor Data, arXiv preprint, arXiv:2410.02429 (2025).

付録

3.4 で説明したプロンプトの実装と 4.2 で説明した WoT TD に基づき記述したセンサ・アクチュエータ以下に示す.

A.1 プロンプト実装

環境状態の同定

ROLE
You are an expert in assessing indoor environmental quality (IEQ) in residential spaces.

TASK
Using only the information provided, evaluate the current indoor environment and classify the following four IEQ condition variables into one of three labels: "High," "Middle," or "Low." Explain the rationale for each classification.

INPUT
Selected Sensors (Values and Metadata):
{bundle_json}

DEFINITIONS
IEQ Condition Variables:

1. Thermal Conditions
2. Indoor Air Quality
3. Lighting/Visual Conditions
4. Acoustics

Label meanings (use these rules)

Thermal Conditions:
High: Thermal Conditions deviates to the upper side.

Middle: Thermal Conditions are within a comfortable range.
Low: Thermal Conditions deviates to the lower side.

Indoor Air Quality:
High: high pollution load, insufficient ventilation.
Middle: pollutants exist but within acceptable limits.
Low: low pollution load, air sufficiently renewed.
Here, "High/Low" refers to "high/low pollution load," not "good/poor air quality."

Lighting/Visual Conditions:
High: illuminance is high.
Middle: comfortable brightness for daily life.
Low: low illuminance.

Acoustics:
High: noisy, disrupts concentration.
Middle: normal noise for daily life.
Low: very quiet.

REASONING RULES
You MUST use Chain-of-Thought prompting.
Start each domain reasoning with the exact phrase: "Let's think step by step."
Produce explicit, short reasoning steps as strings.
For each IEQ variable, explain:

Which sensor values were used (sensor name + value + unit if available)

Why the final label was selected (based ONLY on the rules above)
Use ONLY the information explicitly provided in the sensor bundle.
Do NOT assume any missing values, thresholds, comfort ranges, occupants, activities, or external context not included in the bundle.
Do NOT suggest control actions or recommendations.

OUTPUT RULES
Output ONLY a single valid JSON object.
No markdown.
No extra text.
Output MUST be in English.

OUTPUT FORMAT
{

"cot_steps": {
"Thermal Conditions": ["<step>", "<step>", "<step>"],
"Indoor Air Quality": ["<step>", "<step>", "<step>"],
"Lighting/Visual Conditions": ["<step>", "<step>", "<step>"],
"Acoustics": ["<step>", "<step>", "<step>"]
}
"Corrent_Env": {
"Thermal Conditions": "High/Middle/Low",
"Indoor Air Quality": "High/Middle/Low",
"Lighting/Visual Conditions": "High/Middle/Low",
"Acoustics": "High/Middle/Low",
"Reason": {
"Thermal Conditions": "<Description>",
"Indoor Air Quality": "<Description>",
"Lighting/Visual Conditions": "<Description>",
"Acoustics": "<Description>"
}
},
}

意図推論

ROLE
You are an expert who comprehensively interprets ambiguous natural language requests and the current indoor environment, infers the user's true intention

based on IEQ (Indoor Environmental Quality) indicators, and defines the target ideal indoor condition.

TASK

1. Infer the user intent using ONLY the request text and the provided current environmental conditions labels.
2. Infer the ideal IEQ labels ("High"/"Middle"/"Low") that best satisfy the inferred intent, using ONLY the definitions.

INPUT
User Request:
{user_input}

Environment Understanding (Current IEQ State Label):
{env_info}

DEFINITIONS
IEQ Condition Variables:

1. Thermal Conditions
2. Indoor Air Quality
3. Lighting/Visual Conditions
4. Acoustics

Label meanings (use these rules)

Thermal Conditions:
High: Thermal Conditions deviates to the upper side.
Middle: Thermal Conditions are within a comfortable range.
Low: Thermal Conditions deviates to the lower side.

Indoor Air Quality:
High: high pollution load, insufficient ventilation.

Middle: pollutants exist but within acceptable limits.
Low: low pollution load, air sufficiently renewed.
Here, "High/Low" refers to "high/low pollution load," not "good/poor air quality."

Lighting/Visual Conditions:
High: illuminance is high.
Middle: comfortable brightness for daily life.
Low: low illuminance.

Acoustics:
High: noisy, disrupts concentration.
Middle: normal noise for daily life.
Low: very quiet.

REASONING RULES
You MUST use Chain-of-Thought prompting.
Start your reasoning with the exact phrase: "Let's think step by step."
Produce explicit, short reasoning steps as strings.
Use ONLY the user request and the provided current IEQ labels.
Do NOT fabricate missing sensors, missing values, or unprovided context.
Use ONLY the IEQ rules/definitions above.

OUTPUT RULES
Output ONLY a single valid JSON object.
No markdown.
No extra text.
Output MUST be in English.

OUTPUT FORMAT
{
"cot_steps": {
"intent_inference": ["<step>", "<step>", "<step>"],
"ideal_label_mapping": {

"Temperature": ["<step>", "<step>"],
"Indoor Air Quality": ["<step>", "<step>"],
"Lighting/Visual Conditions": ["<step>", "<step>"],
"Acoustics": ["<step>", "<step>"]
}
},
"ideal_environment": {
"Temperature": "High/Middle/Low",
"Indoor Air Quality": "High/Middle/Low",
"Lighting/Visual Conditions": "High/Middle/Low",
"Acoustics": "High/Middle/Low"
},
"reasoning": {
"environment_reasoning": {
"Temperature": "<Why this ideal label?>",
"Indoor Air Quality": "<Why this ideal label?>",
"Lighting/Visual Conditions": "<Why this ideal label?>",
"Acoustics": "<Why this ideal label?>"
}
}
}

アクチュエータ選択

ROLE
You are an expert in selecting actuators appropriate for residential indoor environments based on IEQ.

STAGE
Actuator Selection

TASK
Select an actuator only if it logically contributes to achieving ideal IEQ domain

label, based ONLY on actuator metadata/actions.
If no actuator contributes, return empty lists.

INPUT
Ideal Environment:
{ideal_env_info}

Current Environment:
{env_info}

External Information:
{external_info}

Available Actuators (ID → Metadata):
{actuator_info}

DEFINITIONS
IEQ Condition Variables:

1. Thermal Conditions
2. Indoor Air Quality
3. Lighting/Visual Conditions
4. Acoustics

Label meanings (use these rules)

Thermal Conditions:
High: Thermal Conditions deviates to the upper side.
Middle: Thermal Conditions are within a comfortable range.
Low: Thermal Conditions deviates to the lower side.

Indoor Air Quality:
High: high pollution load, insufficient ventilation.
Middle: pollutants exist but within acceptable limits.
Low: low pollution load, air sufficiently renewed.

Here, "High/Low" refers to "high/low pollution load," not "good/poor air quality."

Lighting/Visual Conditions:
High: illuminance is high.
Middle: comfortable brightness for daily life.
Low: low illuminance.

Acoustics:
High: noisy, disrupts concentration.
Middle: normal noise for daily life.
Low: very quiet.

REASONING RULES
You MUST use Chain-of-Thought prompting.
Start your reasoning with the exact phrase: "Let's think step by step."
Produce explicit, short reasoning steps as strings.
Use ONLY the provided inputs above.
Do NOT invent actuators, actuator IDs, actions, or functions not explicitly provided.
selected_actuator_ids MUST be an exact subset of the provided actuator IDs.

OUTPUT RULES
Output ONLY a single valid JSON object.
No markdown.
No extra text.
Output MUST be in English.

OUTPUT FORMAT
{
"cot_steps": ["<step>", "<step>", "<step>"],
"selected_actuators": ["<Actuator title>"],
"selected_actuator_ids": ["<actuator_id>"],

"reasoning": "<Brief explanation>"
}

デバイス間整合性検証

ROLE
You are an expert at identifying and explaining conflicts between selected actuators using ONLY the given actuator metadata.

STAGE
Conflict Verification

TASK
Identify logical or physical conflicts that may arise when multiple actuators are operated simultaneously.
Explain each conflict referencing the environmental context and actuator roles.
Do not invent missing functions/conditions. If not justifiable, conclude no conflict.

INPUT
Selected Actuators (ID → Metadata):
{actuator_info}

DEFINITIONS
Conflicts must be identified using ONLY the given actuator metadata.

REASONING RULES
You MUST use Chain-of-Thought prompting.
Start your reasoning with the exact phrase: "Let's think step by step."
Produce explicit, short reasoning steps as strings grounded ONLY in metadata.
Do not invent missing functions/conditions. If not justifiable, conclude no conflict.

OUTPUT RULES
Output ONLY a single valid JSON object.
No markdown.

No extra text.
Output MUST be in English.

OUTPUT FORMAT
{
"cot_steps": ["<step>", "<step>", "<step>"],
"conflicts": "",
"reasoning": "<Why conflict exists or why none>"
}

デバイス制御の決定

ROLE
You are an expert who, using ONLY the given information, sets variable-level target labels and generates actuator control parameters to achieve ideal indoor environmental quality (IEQ) labels.

STAGE
Control Synthesis

TASK
Compare current vs ideal IEQ labels and infer target variable values (numbers) for each domain variable.
Generate feasible actuator control parameters using ONLY provided actuator actions/params.
If conflicts exist per Consistency Verification Results, adjust parameter choices to avoid conflicts, still using ONLY provided actions/params.

INPUT
Current IEQ Label (labels only):
{current_env_info}

Ideal IEQ Label (labels only):
{ideal_env_info}

External Information:
{external_info}

Selected Actuators (ID → {title, actions/params}):
{actuator_details_json}

Consistency Verification Results:
{consistency_info}

DEFINITIONS
IEQ Condition Variables:

1. Thermal Conditions
2. Indoor Air Quality
3. Lighting/Visual Conditions
4. Acoustics

Label meanings (use these rules)

Thermal Conditions:
High: Thermal Conditions deviates to the upper side.
Middle: Thermal Conditions are within a comfortable range.
Low: Thermal Conditions deviates to the lower side.

Indoor Air Quality:
High: high pollution load, insufficient ventilation.
Middle: pollutants exist but within acceptable limits.
Low: low pollution load, air sufficiently renewed.
Here, "High/Low" refers to "high/low pollution load," not "good/poor air quality."

Lighting/Visual Conditions:
High: illuminance is high.

Middle: comfortable brightness for daily life.
Low: low illuminance.

Acoustics:
High: noisy, disrupts concentration.
Middle: normal noise for daily life.
Low: very quiet.

REASONING RULES
You MUST use Chain-of-Thought prompting.
Start your reasoning with the exact phrase: "Let's think step by step."
Produce explicit, short reasoning steps as strings that map current vs ideal labels → variable targets → actuator params.
Use ONLY the provided inputs above.
Never fabricate sensors, values, actuators, actuator IDs, actions, or parameters not present in actuator metadata.

OUTPUT RULES
Output ONLY a single valid JSON object.
No markdown.
No extra text.
Output MUST be in English.

OUTPUT FORMAT
{
"cot_steps": ["<step>", "<step>", "<step>"],
"target_variable_values": {
"temperature": number,
"humidity": number,
"co2": number,
"co": number,
"illuminance": number,
"noise": number
},

"devices": {
"actuator_id": {
"param": "value"
}
},
"reasoning": "<Explanation>"
}

A.2 センサー覧

- 室温度センサ

```
{
  "@context": "https://www.w3.org/2022/wot/td/v1.1",
  "id": "urn:dev:ops:room-temperature-sensor-001",
  "title": "室温度センサ",
  "description": "室内温度を測定するセンサ",
  "security": ["example_sec"],
  "securityDefinitions": {
    "example_sec": { "scheme": "nosec" }
  },
  "properties": {
    "temperature": {
      "title": "室内温度",
      "description": "室内の温度（°C）",
      "type": "number",
      "unit": "degree celsius",
      "readOnly": true,
      "observable": true,
      "forms": [
        {
          "href":
"https://example.com/temperature/properties/temperature",
          "contentType": "application/json",
```

```
            "op": ["readproperty"]
          }
        ]
      }
    }
}
```

- 室湿度センサ

```
{
  "@context": "https://www.w3.org/2022/wot/td/v1.1",
  "id": "urn:dev:ops:room-humidity-sensor-001",
  "title": "室湿度センサ",
  "description": "室内湿度を測定するセンサ",
  "security": ["example_sec"],
  "securityDefinitions": {
    "example_sec": { "scheme": "nosec" }
  },
  "properties": {
    "humidity": {
      "title": "室内湿度",
      "description": "室内の相対湿度（%）",
      "type": "number",
      "unit": "percent",
      "minimum": 0,
      "maximum": 100,
      "readOnly": true,
      "observable": true,
      "forms": [
        {
          "href": "https://example.com/humidity/properties/humidity",
          "contentType": "application/json",
          "op": ["readproperty"]
        }
```

```
        ]
      }
    }
}
```

- 一酸化炭素センサ

```
{
  "@context": "https://www.w3.org/2022/wot/td/v1.1",
  "id": "urn:dev:ops:co-sensor-001",
  "title": "一酸化炭素センサ",
  "description": "一酸化炭素濃度を測定するセンサ",
  "security": ["example_sec"],
  "securityDefinitions": {
    "example_sec": { "scheme": "nosec" }
  },
  "properties": {
    "co": {
      "title": "CO 濃度",
      "description": "一酸化炭素濃度（ppm）",
      "type": "number",
      "unit": "parts per million",
      "readOnly": true,
      "observable": true,
      "forms": [
        {
          "href": "https://example.com/co/properties/co",
          "contentType": "application/json",
          "op": ["readproperty"]
        }
      ]
    }
  }
}
```

- 二酸化炭素センサ


```
{
  "@context": "https://www.w3.org/2022/wot/td/v1.1",
  "id": "urn:dev:ops:co2-sensor-001",
  "title": "二酸化炭素センサ",
  "description": "二酸化炭素濃度を測定するセンサ",
  "security": ["example_sec"],
  "securityDefinitions": {
    "example_sec": { "scheme": "nosec" }
  },
  "properties": {
    "co2": {
      "title": "CO2 濃度",
      "description": "二酸化炭素濃度（ppm）",
      "type": "number",
      "unit": "ppm",
      "readOnly": true,
      "observable": true,
      "forms": [
        {
          "href": "https://example.com/co2/properties/co2",
          "contentType": "application/json",
          "op": ["readproperty"]
        }
      ]
    }
  }
}
```


- 照度センサ


```
{
  "@context": "https://www.w3.org/2022/wot/td/v1.1",
  "id": "urn:dev:ops:illuminance-sensor-001",
```

```
  "title": "照度センサ",
  "description": "室内照度を測定するセンサ",
  "security": ["example_sec"],
  "securityDefinitions": {
    "example_sec": { "scheme": "nosec" }
  },
  "properties": {
    "illuminance": {
      "title": "照度",
      "description": "室内の照度（lux）",
      "type": "number",
      "unit": "lux",
      "readOnly": true,
      "observable": true,
      "forms": [
        {
          "href": "https://example.com/illuminance/properties/illuminance",
          "contentType": "application/json",
          "op": ["readproperty"]
        }
      ]
    }
  }
}
```

- 騒音センサ

```
{
  "@context": "https://www.w3.org/2022/wot/td/v1.1",
  "id": "urn:dev:ops:noise-sensor-001",
  "title": "騒音センサ",
  "description": "騒音レベルを測定するセンサ",
  "security": ["example_sec"],
  "securityDefinitions": {
```

```
    "example_sec": { "scheme": "nosec" }
  },
  "properties": {
    "soundLevel": {
      "title": "騒音レベル",
      "description": "騒音レベル（dB）",
      "type": "number",
      "unit": "decibel",
      "readOnly": true,
      "observable": true,
      "forms": [
        {
          "href": "https://example.com/noise/properties/soundLevel",
          "contentType": "application/json",
          "op": ["readproperty"]
        }
      ]
    }
  }
}
```

A.3 アクチュエーター一覧

- 空気調和機

```
{
  "@context": "https://www.w3.org/2022/wot/td/v1.1",
  "id": "urn:dev:ops:air-conditioner-001",
  "title": "空気調和機",
  "description": "冷暖房および送風量を制御する装置",
  "security": ["nosec_sc"],
  "securityDefinitions": {
    "nosec_sc": { "scheme": "nosec" }
  },
```

```
"actions": {
  "setMode": {
    "title": "モード設定",
    "description": "冷房または暖房モードを切り替える",
    "input": {
      "type": "string",
      "enum": ["cool", "heat"]
    },
    "forms": [
      {
        "href": "https://example.com/aircon/actions/setMode",
        "contentType": "application/json",
        "op": ["invokeaction"]
      }
    ]
  },
  "setFanLevel": {
    "title": "風量設定",
    "description": "送風の強さを設定する",
    "input": {
      "type": "string",
      "enum": ["low", "medium", "high", "auto"]
    },
    "forms": [
      {
        "href": "https://example.com/aircon/actions/setFanLevel",
        "contentType": "application/json",
        "op": ["invokeaction"]
      }
    ]
  }
}
```

```
}
```

- ヒーター

```
{
  "@context": "https://www.w3.org/2022/wot/td/v1.1",
  "id": "urn:dev:ops:heater-001",
  "title": "ヒーター",
  "description": "室内を加熱する暖房装置",
  "security": ["nosec_sc"],
  "securityDefinitions": {
    "nosec_sc": { "scheme": "nosec" }
  },
  "actions": {
    "setTemperature": {
      "title": "温度設定",
      "description": "目標とする加熱温度を設定する",
      "input": {
        "type": "number",
        "unit": "degree celsius"
      },
      "forms": [
        {
          "href": "https://example.com/heater/actions/setTemperature",
          "contentType": "application/json",
          "op": ["invokeaction"]
        }
      ]
    }
  }
}
```

- 加湿器

```
{
  "@context": "https://www.w3.org/2022/wot/td/v1.1",
```

```
  "id": "urn:dev:ops:humidifier-001",
  "title": "加湿器",
  "description": "室内の湿度を上昇させるために，目標湿度を制御する装置",
  "security": ["nosec_sc"],
  "securityDefinitions": {
    "nosec_sc": { "scheme": "nosec" }
  },
  "actions": {
    "setTargetHumidity": {
      "title": "目標湿度設定",
      "description": "目標とする湿度を設定する",
      "input": {
        "type": "number",
        "unit": "percent"
      },
      "forms": [
        {
          "href":
"https://example.com/humidifier/actions/setTargetHumidity",
          "contentType": "application/json",
          "op": ["invokeaction"]
        }
      ]
    },
    "setMistLevel": {
      "title": "ミストレベル設定",
      "description": "ミストの出力レベルを設定する",
      "input": {
        "type": "string",
        "enum": ["low", "medium", "high"]
      },
      "forms": [
```

```
          {
            "href": "https://example.com/humidifier/actions/setMistLevel",
            "contentType": "application/json",
            "op": ["invokeaction"]
          }
        ]
      }
    }
}
```

- 除湿器

```
{
  "@context": "https://www.w3.org/2022/wot/td/v1.1",
  "id": "urn:dev:ops:dehumidifier-001",
  "title": "除湿機",
  "description": "室内の湿度を低下させる装置",
  "security": ["nosec_sc"],
  "securityDefinitions": {
    "nosec_sc": { "scheme": "nosec" }
  },
  "actions": {
    "setTargetHumidity": {
      "title": "目標湿度設定",
      "description": "目標とする湿度を設定する",
      "input": {
        "type": "number",
        "unit": "percent"
      },
      "forms": [
        {
          "href":
"https://example.com/dehumidifier/actions/setTargetHumidity",
          "contentType": "application/json",
```

```
            "op": ["invokeaction"]
          }
        ]
      },
      "setFanSpeed": {
        "title": "風速設定",
        "description": "送風の強さを設定する",
        "input": {
          "type": "string",
          "enum": ["low", "medium", "high"]
        },
        "forms": [
          {
            "href": "https://example.com/dehumidifier/actions/setFanSpeed",
            "contentType": "application/json",
            "op": ["invokeaction"]
          }
        ]
      }
    }
}
```

- 扇風機

```
{
  "@context": "https://www.w3.org/2022/wot/td/v1.1",
  "id": "urn:dev:ops:fan-001",
  "title": "扇風機",
  "description": "室内の空気を循環させるために，回転速度を制御する送風装置",
  "security": ["nosec_sc"],
  "securityDefinitions": {
    "nosec_sc": { "scheme": "nosec" }
  },
  "actions": {
```

```
    "setSpeed": {
      "title": "速度設定",
      "description": "回転速度（段階）を設定する",
      "input": {
        "type": "integer",
        "minimum": 1,
        "maximum": 5
      },
      "forms": [
        {
          "href": "https://example.com/fan/actions/setSpeed",
          "contentType": "application/json",
          "op": ["invokeaction"]
        }
      ]
    }
  }
}
```

- 空気清浄機

```
{
  "@context": "https://www.w3.org/2022/wot/td/v1.1",
  "id": "urn:dev:ops:air-purifier-001",
  "title": "空気清浄機",
  "description": "空気中の粒子状物質等を低減するために，運転モードを制御する装置",
  "security": ["nosec_sc"],
  "securityDefinitions": {
    "nosec_sc": { "scheme": "nosec" }
  },
  "actions": {
    "setMode": {
      "title": "モード設定",
```

```
        "description": "運転モードを切り替える",
        "input": {
          "type": "string",
          "enum": ["auto", "normal", "turbo"]
        },
        "forms": [
          {
            "href": "https://example.com/air-purifier/actions/setMode",
            "contentType": "application/json",
            "op": ["invokeaction"]
          }
        ]
      }
    }
}
```

- 換気扇

```
{
  "@context": "https://www.w3.org/2022/wot/td/v1.1",
  "id": "urn:dev:ops:ventilation-fan-001",
  "title": "換気扇",
  "description": "室内外の空気を入れ替えるために，風量や運転時間を制御する換気装置",
  "security": ["nosec_sc"],
  "securityDefinitions": {
    "nosec_sc": { "scheme": "nosec" }
  },
  "actions": {
    "setFanSpeed": {
      "title": "換気量設定",
      "description": "換気量を設定する",
      "input": {
        "type": "string",
```

```
          "enum": ["low", "medium", "high"]
        },
        "forms": [
          {
            "href": "https://example.com/ventilation-fan/actions/setFanSpeed",
            "contentType": "application/json",
            "op": ["invokeaction"]
          }
        ]
      },
      "setTimer": {
        "title": "タイマー設定",
        "description": "運転時間を設定する",
        "input": {
          "type": "integer",
          "unit": "minute"
        },
        "forms": [
          {
            "href": "https://example.com/ventilation-fan/actions/setTimer",
            "contentType": "application/json",
            "op": ["invokeaction"]
          }
        ]
      }
    }
}
```

- 自動開閉扉

```
{
  "@context": "https://www.w3.org/2022/wot/td/v1.1",
  "id": "urn:dev:ops:automatic-door-001",
  "title": "自動開閉扉",
```

```
  "description": "出入口の開閉状態を自動制御する装置",
  "security": ["nosec_sc"],
  "securityDefinitions": {
    "nosec_sc": { "scheme": "nosec" }
  },
  "actions": {
    "setPosition": {
      "title": "開閉状態設定",
      "description": "扉を開く／閉じる状態に設定する",
      "input": {
        "type": "string",
        "enum": ["open", "close"]
      },
      "forms": [
        {
          "href": "https://example.com/automatic-door/actions/setPosition",
          "contentType": "application/json",
          "op": ["invokeaction"]
        }
      ]
    }
  }
}
```

- 自動開閉窓

```
{
  "@context": "https://www.w3.org/2022/wot/td/v1.1",
  "id": "urn:dev:ops:automatic-window-001",
  "title": "自動開閉窓",
  "description": "窓の開閉状態を自動制御する装置",
  "security": ["nosec_sc"],
  "securityDefinitions": {
    "nosec_sc": { "scheme": "nosec" }
```

```
  },
  "actions": {
    "setPosition": {
      "title": "開閉状態設定",
      "description": "窓を開く／閉じる状態に設定する",
      "input": {
        "type": "string",
        "enum": ["open", "close"]
      },
      "forms": [
        {
          "href": "https://example.com/automatic-
window/actions/setPosition",
          "contentType": "application/json",
          "op": ["invokeaction"]
        }
      ]
    }
  }
}
```

- 照明

```
{
  "@context": "https://www.w3.org/2022/wot/td/v1.1",
  "id": "urn:dev:ops:main-light-001",
  "title": "照明",
  "description": "室内を照らすために，明るさを制御する照明装置",
  "security": ["nosec_sc"],
  "securityDefinitions": {
    "nosec_sc": { "scheme": "nosec" }
  },
  "actions": {
    "setBrightness": {
```

```
        "title": "明るさ設定",
        "description": "照明の明るさを設定する",
        "input": {
          "type": "integer",
          "minimum": 0,
          "maximum": 100,
          "unit": "percent"
        },
        "forms": [
          {
            "href": "https://example.com/main-light/actions/setBrightness",
            "contentType": "application/json",
            "op": ["invokeaction"]
          }
        ]
      }
    }
}
```

- 遮光カーテン

```
{
  "@context": "https://www.w3.org/2022/wot/td/v1.1",
  "id": "urn:dev:ops:curtains-001",
  "title": "遮光カーテン",
  "description": "外光の遮断・調整のために，開閉位置を制御するカーテン装置",
  "security": ["nosec_sc"],
  "securityDefinitions": {
    "nosec_sc": { "scheme": "nosec" }
  },
  "actions": {
    "setPosition": {
      "title": "位置設定",
      "description": "カーテンの開閉位置を設定する",
```

```
        "input": {
          "type": "integer",
          "minimum": 0,
          "maximum": 100,
          "unit": "percent"
        },
        "forms": [
          {
            "href": "https://example.com/curtains/actions/setPosition",
            "contentType": "application/json",
            "op": ["invokeaction"]
          }
        ]
      }
    }
}
```

- スピーカー

```
{
  "@context": "https://www.w3.org/2022/wot/td/v1.1",
  "id": "urn:dev:ops:smart-speaker-001",
  "title": "スピーカー",
  "description": "音声を出力する装置であり，音量を制御する",
  "security": ["nosec_sc"],
  "securityDefinitions": {
    "nosec_sc": { "scheme": "nosec" }
  },
  "actions": {
    "setVolume": {
      "title": "音量設定",
      "description": "音量(%)を設定する",
      "input": {
        "type": "integer",
```

```
            "minimum": 0,
            "maximum": 100,
            "unit": "percent"
          },
          "forms": [
            {
              "href": "https://example.com/smart-speaker/actions/setVolume",
              "contentType": "application/json",
              "op": ["invokeaction"]
            }
          ]
        }
      }
}
```