

オントロジを活用した多様な利用者環境に適用可能な IoT サービス構成手法

和室 昂佑¹ 北形 元² 長谷川 剛²

概要: 本稿では, IoT に関する知識が少ない利用者が, IoT デバイスを組み合わせたサービスを簡単に利用できるようにするための IoT サービス構成手法を提案する. 近年, 多様な IoT デバイスを組み合わせてサービスを構築するフレームワークや開発手法が提案されている. しかし, それらを利用する為には, 個々の IoT デバイスやプログラムに関する知識が必要となり, 専門知識の少ない利用者が IoT デバイスを組み合わせて IoT サービスを構成することは難しい. そこで提案手法では, IoT デバイスとその動作及び利用者環境を表現できるオントロジを導入し, それに基づいて IoT サービスのレシピを記述する. また, レシピを基に多様な利用者環境において自動的に IoT サービスを構成するための機構を導入する. 試作システムを用いた実験を通じ, 提案手法の有効性を示す.

An ontology-based IoT service composition method for diversified users' environments

KOUSUKE WAMURO¹ GEN KITAGATA² GO HASEGAWA²

Abstract: In this report, we propose an IoT service composition method that enables users with little knowledge of IoT to use services made with a combination of IoT devices easily. In recent years, frameworks and development methods for building services by combining various IoT devices have been proposed. However, because knowledge of individual IoT devices and programming is required to use them, it is difficult for users with little expertise to compose IoT services by combining IoT devices. In our method, we introduce an ontology that can represent IoT devices, their behaviors, and a user's environment. We describe IoT service recipes based on the ontology. We also introduce a mechanism to automatically compose IoT services in various user environments based on the recipes. We show the effectiveness of our method through experiments using a prototype system.

1. はじめに

近年, あらゆるものをインターネットに接続するという Internet of Things (IoT) の考え方に基づいて, 住宅やオフィス, 学校, 工場などの多様な環境を考慮したシステムやサービスの開発が活発に行われている. IoT デバイスが有するセンサによる環境の観測とアクチュエータによる環境への作用を組み合わせ, デジタルな世界と物理的な環境

を相互に作用させることにより, 多様なタスクの自動化が見込まれる [1]. 以降, IoT デバイスを組み合わせて構成されたサービスを, 単に IoT サービスと呼称する.

IoT デバイス開発者, IoT サービス開発者及び一般の利用者を仲介する Web サービスとして, IFTTT [2] や ConradConnect [3] などの Integration Platform as a Service (IPaaS) と呼ばれるサービスが提供されている [4]. 利用者はサービスに対応した IoT デバイスの動作を組み合わせ, IoT サービスを構成できる. また, 利用者は作成した IoT サービスの設計図をレシピとして公開することができる. これにより, ある利用者が作成したレシピを他の利用者が使用し, IoT サービスを構成することができる. このよう

¹ 東北大学 大学院情報科学研究科
Graduate School of Information Sciences, Tohoku University

² 東北大学 電気通信研究所
Research Institute of Electrical Communication, Tohoku University

なレシピ型の IoT サービス構成手法は、専門知識の少ない利用者が IoT サービスを容易に利用できるようにするために有用である。

既存のレシピ型の IoT サービス構成手法は、特定ベンダの IoT デバイスを対象としてレシピを記述するため、似た効果をもたらす異種の IoT デバイスを同一のレシピで扱うことができない。専門知識を持つ利用者であれば、レシピを書き換えたり新たに記述することで、異種の IoT デバイスを活用した IoT サービスを構成することができる。しかし、このような作業には IoT デバイスに関する知識が求められるため、専門知識の少ない利用者にとってはこの作業は難しい。また、利用者は構成したい IoT サービスに合わせて、自身の環境に存在する多様な IoT デバイスの中から、利用するものを選択する必要がある。しかし、選択できる IoT デバイスが多数存在する場合には、適切な選択が難しい。

そこで我々の研究グループでは、抽象度の高い表現でレシピを記述し、オントロジを参照してレシピに記述された要件を満たす IoT デバイスの組み合わせを自動的に決定することで、専門知識の少ない利用者でも IoT サービスを容易に利用可能とする手法を検討してきた [5]。本稿では [5] で検討した手法を具体化した IoT サービス構成手法を提案する。具体的には、IoT デバイスとその動作及び利用者環境を表現するためのオントロジを導入し、それに基づいて IoT デバイスの動作を基にレシピを記述する。また、多様な IoT デバイスが導入された利用者環境において、レシピを基に IoT デバイスを自動的に選択し、IoT サービスを構成するサービス構成機構を導入する。試作システムを用いた実験を通じ、提案手法の有効性を示す。

本稿の構成は以下の通りである。2章で関連研究を述べ、3章で提案手法を述べる。4章で試作システムの実装について述べ、5章で試作システムを用いた実験と評価について述べ、6章でまとめを述べる。

2. 関連研究

2.1 IoT のシステムの構成要素とそれらの関係性を表現するオントロジ

オントロジとは、ある対象領域を説明するために必要な構成要素を、概念（クラス）と概念間の関係（プロパティ）及び具体物（インスタンス）に分類し、表現したものである [6]。オントロジは、IoT を構成する要素とそれらの関係性を表現するために有用であり、IoT デバイスの動作や生成される情報に概念的な意味を与えることで、多様な IoT デバイスを統一的に扱うことができる。IoT に関連する多様な領域や用途を想定した約 200 種類のオントロジが提案されている [7]。

[8] においては、多様なセンサが取得した情報の不均一性による問題を解決するため、センサ及びセンサが実行する観

測を説明できる Semantic Sensor Network ontology (SSN) 及び Sensor, Observation, Sample and Actuator (SOSA) が提案されている。また、[9] においては、アクチュエータの動作及び環境への影響を表現するためには、SSN では不十分であるとして、アクチュエータの動作と環境への影響を表現できる Semantic Actuator Network ontology (SAN) が提案されている。

一方、センサによる環境の観測やアクチュエータによる環境への作用を表現するためには、環境に関する知識（コンテキスト）が必要となる。例えば、あるセンサが観測する場所や対象物を分類し、関係性を記述することによって、センサと環境との関係を表現できる。そのようなオントロジの 1 つとして、[10] では、ホームオートメーションを実現するため、屋内環境を詳細にモデリングするオントロジとして DogOnt が提案されている。DogOnt では、屋内に存在する多様な物や建造物の構造を表現・分類するために多様なクラスが定義されている。

本稿では、SSN, SOSA, SAN, DogOnt を拡張することで、IoT デバイスとその動作及び利用者の環境を表現するオントロジを導入する。

2.2 オントロジを利用した IoT サービスの構成手法

前節で説明したオントロジを利用し、多様な IoT デバイスを連携させ、IoT サービスを構成するための研究が行われている。Open IoT [11] では、センサが取得した情報を SSN に基づいた注釈をつけて保存する。また、利用者はビジュアルプログラミングツールを用いて、センサが観測する場所や観測情報を選択することで、センサが取得した情報を可視化する IoT サービスを構成できる。

[12] においては、IoT デバイスの動作を抽象的に表現するためにオントロジを導入し、抽象的に IoT サービスを記述・構成するための末端の利用者向けのプログラミング手法が提案されている。この手法では、利用者は IoT デバイスの動作を、オントロジを記述することで抽象化し、記述したオントロジを用いてプログラムを作成することで IoT サービスを構成する。例えば、ライトを点灯する動作とブラインドを上げる動作に対して、“明るくする” という効果を持っていると指定することで、両者を同一の記述で扱い、IoT サービスを構成する。

これらの手法では利用者がオントロジの記述、IoT サービスの設計及びプログラムの作成を行う必要があるが、専門知識の少ない利用者がこれらの作業を行うことは難しい。そこで本稿では、オントロジを利用した抽象的なレシピの記述と、レシピを基にした IoT サービスの自動的な構成を実現する。

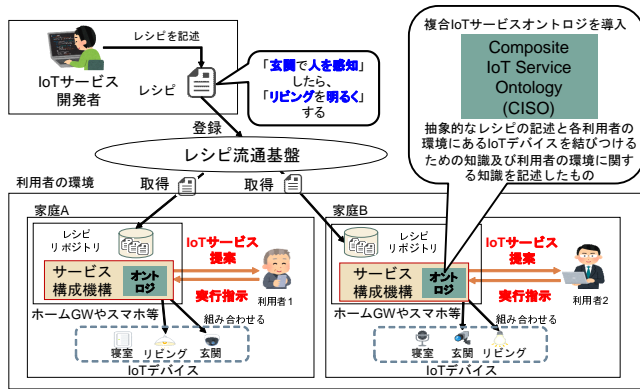


図 1 提案手法の概要

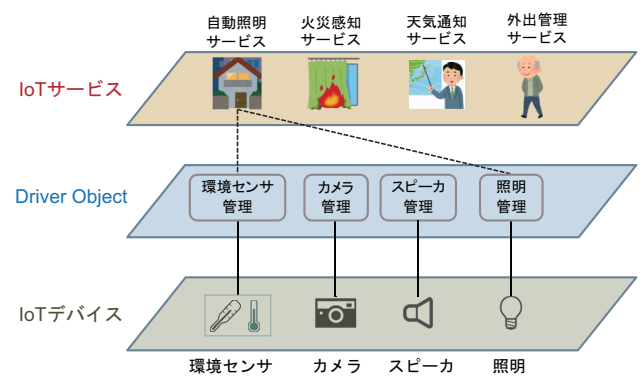


図 2 IoT サービスのアーキテクチャ

3. オントロジを活用した多様な利用者環境に適用可能な IoT サービス構成手法

3.1 概要

図 1 に、本稿で提案する、オントロジを活用した多様な利用者環境に適用可能な IoT サービス構成手法の概要を示す。提案手法は、以下の 3 つの要素から構成される。

- (1) 抽象的な動作表現と具体的な IoT デバイスを結びつけるオントロジ
異なる概念間の関連性を表現可能な情報表現手法であるオントロジを用いて、多様な IoT デバイスを抽象的に表現し、それらを統合的に扱うことを可能とするための Composite IoT Service Ontology (CISO) を導入する。
- (2) IoT デバイスの要件とその連携動作を抽象的に表現可能なレシピ
利用したい IoT デバイスの動作をオントロジを用いることで抽象的に記述し、さらにそれらの IoT デバイスを連携させるためのレシピの記述方法を導入する。
- (3) レシピの適用可否の判定, IoT デバイスの選択, IoT サービスの構成を実現する機構
利用者の環境においてレシピの適用が可能かどうかを判定し、適用可能な場合はレシピに基づいて IoT デバイスを選択・連携させ、IoT サービスを構成する。これを実現する機構として、サービス構成機構を導入する。

また、提案手法のタスクは大別して 2 つある。すなわち、IoT サービス開発者側で行う IoT サービスの開発・登録のタスクと、利用者側で行う IoT サービスの構成のタスクである。以下で各タスクについて述べる。

- (1) IoT サービスの開発・登録のタスク
IoT サービス開発者側で行うタスクであり、IoT サービス開発者は IoT サービスの設計及びレシピの記述を行う。さらに、記述したレシピをレシピ流通基盤に登録する。このとき、IoT サービス開発者は、抽象的な動作表現と具体的な IoT デバイスを結びつけるオント

ロジとして新たに導入する CISO を用いて、IoT デバイスの要件とその連携ルールを抽象的なレシピとして記述する。

- (2) IoT サービスの構成のタスク
利用者側で行うタスクであり、利用者はレシピ流通基盤から利用したいレシピを入手し、利用者が入手したレシピを保管する場所であるレシピリポジトリに登録する。その後、後述するサービス構成機構から提示される利用可能な IoT サービスの中から、利用したい IoT サービスを選択し、サービス構成機構に選択した IoT サービスの実行を指示する。その結果、IoT デバイスが組み合わせられ、その IoT サービスが構成される。

3.2 IoT サービスのアーキテクチャ

図 2 に、提案手法における IoT サービスのアーキテクチャを示す。IoT サービスは 3 つの層で構成される。すなわち、IoT デバイス層、Driver Object 層、IoT サービス層である。IoT デバイスは、物理的な環境の情報取得するセンサーや動作することで環境に作用を与えるアクチュエータ及び、それらを搭載し取得した情報を処理する能力を備えたデバイスである。Driver Object は、IoT デバイスを表すソフトウェアコンポーネントであり、IoT デバイスを管理・操作するためのインターフェイスを提供する。IoT サービスは、レシピに基づいて IoT デバイスを連携させることで構成するサービスであり、Driver Object を介して構成する。例えば、Driver Object は REpresentational State Transfer (REST) API として IoT デバイスを操作するインターフェイスを提供し、それを連携させることで IoT サービスを構成する。

3.3 Composite IoT Service Ontology (CISO)

3.3.1 CISO の概要

図 3 に、多様な IoT デバイスとその動作を抽象的に表現し、それらを統合的に扱うことを可能とする Composite IoT Service Ontology (CISO) の概要を示す。CISO は IoT デバイス及びそれらが保持するセンサーやアクチュエータの

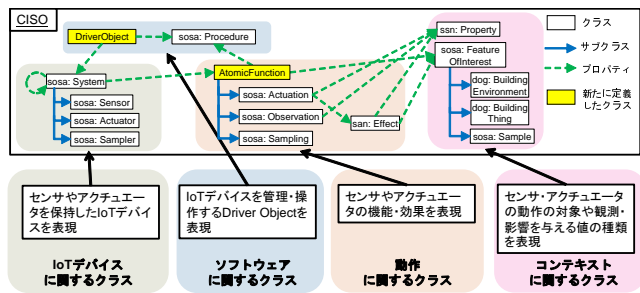


図 3 Composite IoT Service Ontology (CISO)

動作を表現するため、4つのクラスから構成される。すなわち、IoTデバイスに関するクラス、動作に関するクラス、コンテキストに関するクラス、及びソフトウェアに関するクラスである。なお、各クラスの“:”以前の文字列(接頭辞)はそれぞれ以下の外部のオントロジを示しており、“:”以降の文字列がそのオントロジで定義されたクラスを示す。また、接頭辞のないクラスは本提案で新たに追加したクラスである。以下に接頭辞とそれが意味する外部のオントロジを記す。

- ssn: Semantic Sensor Network Ontology [8]
- sosa: Sensor, Observation, Sample, and Actuator [8]
- dog: DogOnt [10]
- san: Semantic Actuator Network Ontology [9]

以下に大別した4つのクラスについて述べる。

- IoTデバイスに関するクラス
物理的もしくは仮想的なIoTデバイスを表現するためのクラスであり、SOSAで定義されたクラス及びプロパティを利用する。IoTデバイスを表現するクラスとして“sosa: System”クラス及びそのサブクラスを利用する。また、IoTデバイスがこれらのセンサやアクチュエータなどの機器を保持している関係をプロパティを用いて表す。
- 動作に関するクラス
IoTデバイスの動作を抽象的に表現するためのクラスであり、SOSA, SANで定義されたクラス及びプロパティを利用する。具体的には、IoTデバイスが実現する動作を、本提案で新たに導入する“Atomic Function”クラスを用いて表現する。
- コンテキストに関するクラス
IoTデバイスの動作の対象を示す場所や物及び観測・作用する値を表現するためのクラスであり、SSN, SOSA, DogOntで定義されたクラス及びプロパティを利用する。“sosa: FeatureOfInterest”クラスは観測や作用の対象となる物や場所などを表す。また、“ssn: Property”クラスは観測される値や作用によって影響を受ける値を表す。
- ソフトウェアに関するクラス
IoTデバイスを管理・操作するソフトウェアコンポーネントを表現するためのクラスであり、SOSAで定義さ

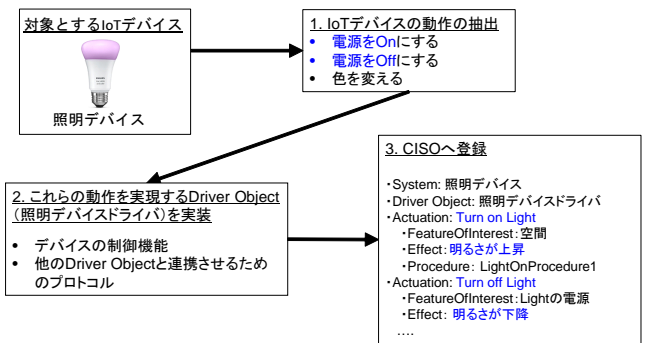


図 4 Driver Object の設計・開発及び CISO への登録手順

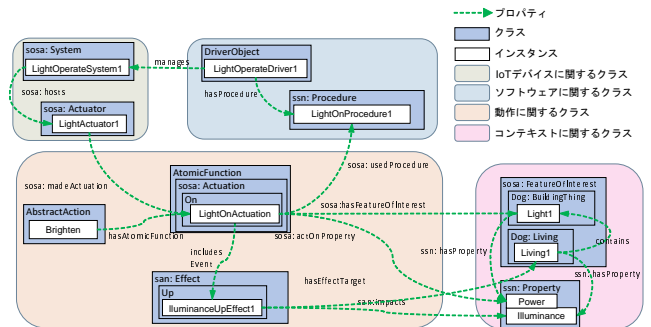


図 5 CISO を用いたアクチュエータを保有する IoT デバイスの表現

れたクラス及びプロパティを利用する。本提案で新たに定める“Driver Object”クラスはIoTデバイスを管理・操作するソフトウェアコンポーネントを表す。また、“Atomic Function”クラスが表現する機能を実装したソフトウェアを、“sosa: Procedure”クラスとして表す。また、“Driver Object”が“sosa: Procedure”を持つという関係をプロパティを用いて表す。

3.3.2 Driver Object の設計・開発及び CISO への登録

本項では、Driver Object の設計・開発及び CISO への登録の手順について述べる。図 4 に、ある照明デバイスにおける Driver Object の設計・開発及び CISO への登録を行う例を示す。以下にその手順を記述する。

- (1) IoT デバイスの動作の抽出
IoT デバイスの動作を抽出する。この例では、照明デバイスの On/Off 及び色を変える動作の3つを抽出している。
- (2) Driver Object の構築
後述するサービス構成モジュールからの指示に従い、対象の IoT デバイスを制御するためのソフトウェアとして Procedure を実装し、その IoT デバイスが持つすべての Procedure を持つ Driver Object を構築する。この例では、照明デバイスの On/Off や色を変える制御を行う動作をそれぞれ3つの Procedure として実装し、さらにそれらを保有する Driver Object を構築する。また、他の Driver Object と連携するためのプロトコルを導入する。具体的には、Message Queuing

Telemetry Transport (MQTT) や REST API などの既存の通信プロトコルの導入や、対象の IoT デバイス専用の独自のプロトコルを用いることができる。

(3) CISO のインスタンスとして登録

IoT デバイスの情報及び IoT デバイスを制御する Driver Object の情報を、CISO のインスタンスとして登録する。具体的には、図 5 に示すようなオントロジを構成する。なお、図 5 は照明を On にするアクチュエータのインスタンスとそれらの関係を示している。IoT デバイスに関するクラスでは、この IoT デバイスがアクチュエータを持つことを示している。動作に関するクラスでは、アクチュエータが何かを On にする動作を持ち、またその動作は何かを上昇させる効果を持つことを示している。コンテキストに関するクラスでは、動作の対象となる物がリビングにある照明の電源であり、その動作による上昇効果はリビングの明るさに影響を与えることを示している。ソフトウェアに関するクラスでは、この IoT デバイスを管理・操作する Driver Object 及び照明を On にする動作を実装した Procedure を示している。

3.4 レシピの記述方法

多様な利用者環境に適用可能なレシピを記述するため、CISO における動作に関するクラスとコンテキストに関するクラスを利用する。例えば、レシピを適用するために人がいることを検知する動作が必要であり、異なる利用者の環境に、人を認識する機能を備えたカメラ、Passive Infrared Ray (PIR) センサ及び赤外線温度センサがそれぞれ配置されている場面を考える。この時、それぞれの IoT デバイスは人がいることを検知できると CISO のインスタンスとして登録しておくことで、抽象的な動作記述と具体的な IoT デバイスを結びつけることができる。これにより、同一のレシピを基に利用者環境に導入された多様な IoT デバイスを活用し、IoT サービスを構成できる。なお、レシピは 2 つの要素で構成する。すなわち、必要とする IoT デバイスの動作、連携ルールである。

図 6 に、エントランスに人が来た場合にリビングを明るくするレシピを作成する場合に、必要とする IoT デバイスの動作を記述した例を示す。必要とする IoT デバイスの動作は、“sosa: Observation” や “sosa: Actuation” などのクラスを用いて示す IoT デバイスの動作の種類，“sosa: featureOfInterest” クラスを用いて示す観測や作動の対象となる物や箇所、及び “ssn: Property” クラスを用いて示す観測される値・作動の影響を受ける値を利用して記述する。図 6 における例では、エントランスで人の存在を観察する動作とリビングの明るさを上昇させる効果を持つ動作の 2 つを要求している。

```

"functions": [
  {
    "name": "detectHumanInEntrance",
    "required": "true",
    "description": "玄関に人がいるかどうかを検知する機能",
    "type": "observation",
    "featureOfInterest": "entrance",
    "observes": "humanExistence"
  },
  {
    "name": "brightenLiving",
    "required": "true",
    "description": "リビングを明るくする効果を持つアクチュエーション",
    "type": "actuation",
    "effect": {
      "type": "up",
      "featureOfInterest": "living",
      "property": "illuminance"
    }
  }
]
    
```

図 6 IoT デバイスの動作の記述例

```

"rules": {
  "script": "detectHumanInEntrance -a | humanDetected | brightenLiving",
  "ruleMethods": [
    {
      "name": "humanDetected",
      "language": "python",
      "script": "humanDetected.py"
    }
  ]
}
    
```

図 7 連携ルールの記述例

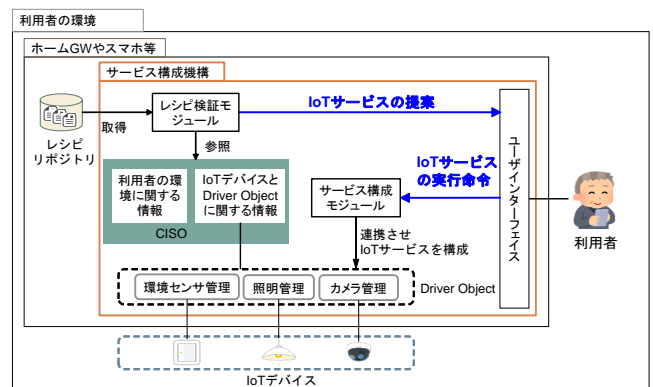


図 8 サービス構成機構

図 7 に、エントランスに人が来た場合にリビングを明るくするレシピを作成する場合に、連携ルールを記述した例を示す。連携ルールは、IoT デバイスの動作をどのように組み合わせるかを記述したものである。図における例では、“rules” として記述した部分に相当する。この例では、観測や作動の結果を受け取って計算処理や判定を行うプログラムを “ruleMethods” として定めており、“script” として “functions” 及び “ruleMethods” をどのように組み合わせるかを示している。また、IoT サービスの動作順序は “|” を用いて記述し、前の動作による出力を受け取り次の動作の入力へと渡す仕組みとした。

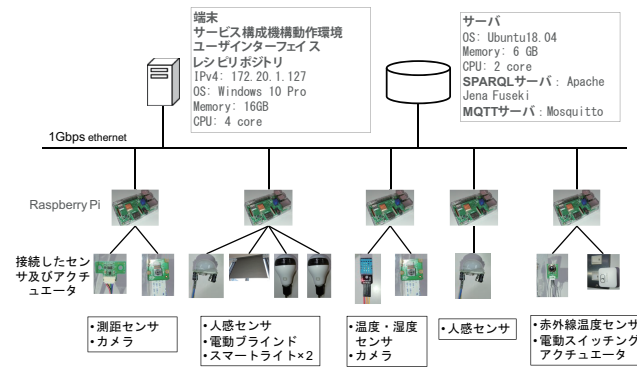


図9 試作システムの構成

3.5 サービス構成機構

異種のIoTデバイスが導入された多様な利用者の環境において、レシピの適用可否を判断し、さらに適用可能なIoTデバイスを選択・連携させ、IoTサービスを構成することを実現するため、これらの機能を備えたサービス構成機構を導入する。図8に、サービス構成機構の概要を示す。サービス構成機構は、収集されたレシピを保管するレシピリポジトリと同様に、利用者の持つホームゲートウェイやスマートフォンなどのデバイス上に配置する。

以下にサービス構成機構の動作を示す。

- (1) レシピの適用可否の判定及び利用可能なIoTデバイスの選択

レシピ検証モジュールは、利用者の環境及び保有するIoTデバイスとDriver Objectに関する情報がインスタンスとして登録されたCISOを参照し、レシピが必要とする動作を実現できるIoTデバイスが、利用者の環境に存在するかどうかを確認する。もし存在する場合はレシピを適用できると判定し、利用者に対して発見されたIoTデバイスを連携させるIoTサービスの提案を行う。

- (2) IoTサービスの構成

IoTサービスの提案を受けた利用者からIoTサービスの実行命令が発せられた場合、サービス構成モジュールに実行命令が通知される。サービス構成モジュールは、Driver Objectを介してIoTデバイスを管理・操作し、またレシピにおける連携ルールに従ってそのIoTデバイスを連携させることで、IoTサービスを自動的に構成する。

4. 試作システムの設計と実装

図9に、試作システムの構成を示す。試作システムで使用する各デバイスは、1 Gbpsの有線LANで接続されたローカルネットワーク上に配置した。また、サービス構成機構の実装を行い、レシピリポジトリと共にPC上に導入した。CISOはResource Description Framework (RDF)の

表1 実験1: センサの動作と観測する箇所

デバイスのインスタンス名	pir sensor 201	pir sensor 203	distance sensor 402	camera 202	camera 402	temperature sensor 202	humidity sensor 202	infrared temperature sensor 204
実現する動作	人を検知	人を検知	人を検知 距離を測定	人を検知	人を検知	温度を測定	湿度を測定	表面温度を測定
配置1-1	entrance	living	entrance	entrance	kitchen	entrance	entrance	entrance
配置1-2	living		living		kitchen	bathroom	bathroom	entrance
配置1-3	bathroom	living			entrance			entrance

語彙を拡張したWeb Ontology Language (OWL)を用いて記述した。記述したオントロジは、SPARQLサーバであるApatch Jena Fuseki [13]上に配置した。SPARQLサーバは、RDFクエリ言語であるSPARQLによる問い合わせを受け付け、レスポンスを返す。また、IoTデバイスが持つセンサの値を送受信するためのMQTTサーバであるMosquitto [14]を稼働させた。さらに、本試作システムではRaspberry Pi3B+及びRaspberry Pi4にセンサやアクチュエータを接続したものをIoTデバイスとした。また、サービス構成機構の実装及びIoTデバイスの制御プログラムの記述にはJava, Python, Perlを使用した。なお、接続した具体的なセンサやアクチュエータは、人感センサ (PIR Motion Sensor Module DYP-ME003) [15], 温度・湿度センサ (DHT11) [16], 赤外線温度センサ (オムロン D6T-44L-06) [17], レーザー測距センサ (AE-VL53L0X) [18], カメラモジュール (Raspberry PiCamera Module V2.1) [19], 電動ブラインド (IKEA FYRTUR フィルトゥール) 及び電動ブラインドリモコン [20], スマートライト (AZUCK LED スピーカーライト) [21], スイッチボット (SwitchBot S1) [22]である。

5. 実験と評価

5.1 実験1: レシピの適用可否の判定の検証

本実験では、レシピで必要とされた動作より具体的なIoTデバイスを選択し、さらにレシピが適用可能か判定できることを検証する。実験手順は以下のとおりである。

- (1) レシピを4つ用意し、それぞれをレシピリポジトリに登録する
- (2) IoTデバイスの配置を3種類用意する
- (3) 各IoTデバイスの配置毎にサービス構成機構にレシピの適用可否を判定させる

また、判定が正しくできていること及びレシピが必要とするIoTデバイスが選択できていることを、配置場所やIoTデバイスの動作、レシピを比較して確認する。登録した4つのレシピにおいて、それぞれ必要とするIoTデバイスの動作を2つ記述している。なお、本実験結果では、これらのレシピで必要とされるIoTデバイスの動作を「バスルームの気温を観測する」「バスルームを温かくする」のように口語化して示す。表1及び表2に、それぞれ実験で用いるセンサ及びアクチュエータの動作と観測及び影響を与

表 2 実験 1: アクチュエータの動作と影響を与える箇所

デバイスのインスタンス名	light actuator 201	light actuator 202	blinds actuator 201	switchBot actuator 204
実現する動作	明るくする 暗くする	明るくする 暗くする	明るくする 暗くする	明るくする 暗くする
配置 1-1	entrance	living	entrance	entrance
配置 1-2		living		
配置 1-3			living	entrance

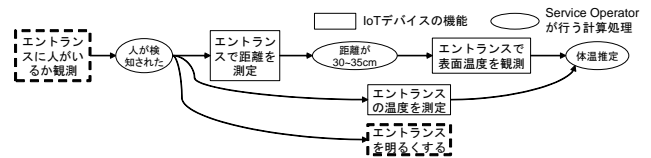


図 11 実験 2: レシピの概要

Available recipes	Unavailable recipes	Working IoT Services
バスルームの気温と湿度を測定する		
キッチンに人がいた場合にキッチンを明るくする		
バスルームの温度を測定し条件によってバスルームを温める		

Description
IoTサービスの説明: 玄関に人が来た場合にリビングを明るくするサービス
組み合わされる機能:
1. 玄関に人がいるかどうかを検知する機能
→ 利用する機器: distanceSensor402
2. リビングを明るくするアクチュエーション
→ 利用する機器: lightActuator201

(a) 構成可能なレシピのリスト (b) 構成不可能なレシピのリスト

図 10 実験 1: レシピの適用可否の判断結果

表 3 実験 1: サービス構成機構による IoT デバイスの選択結果

レシピの名	必要とする IoT デバイスの動作	必要とする IoT デバイスの動作	必要とする IoT デバイスの動作	必要とする IoT デバイスの動作	必要とする IoT デバイスの動作	必要とする IoT デバイスの動作
配置 1-1	distance sensor 402	light actuator 201				
配置 1-2			temperature sensor 202	humidity sensor 202	camera 402	temperature sensor 202
配置 1-3	camera 402	blinds actuator 201				

える箇所を示す。なお、斜線はそのデバイスを配置していないことを示す。

図 10 に、配置 1-1 の場合にサービス構成機構にレシピの適用可否の判断をさせた場合のインターフェイスの様子を示す。図 10(a) は、適用可能なレシピのリストを示している。これにより、エントランスに人がいた場合にリビングを明るくするレシピに基づき、IoT サービスを構成できることを確認した。また、Description の項に選択された IoT デバイスのインスタンス名が出力されているが分かる。図 10(b) は、適用不可能なレシピのリストを示している。3つのレシピがこちらのリストに示されていることが分かる。また、Description の項に必要とする IoT デバイスの動作を実現できる IoT デバイスが不足していることが出力されていることが分かる。

表 3 に、サービス構成機構が、それぞれのレシピで必要とされる動作を実現できる IoT デバイスを選択した結果を示す。表 1 と表 2 の実験条件、及び表 3 の実験結果から、選択された IoT デバイスの動作と観測箇所及び影響を与える箇所がレシピに記述された必要とする IoT デバイスの動作と合致していることを確認した。以上の実験結果より、レシピで必要とされた動作を基に具体的な IoT デバイスを選択でき、レシピが適用可能か判定できることを確認した。

表 4 実験 2: センサの動作及び観測する箇所

デバイスのインスタンス名	pir sensor 201	pir sensor 203	distance sensor 402	camera 202	camera 402	temperature sensor 202	humidity sensor 202	infrared temperature sensor 204
実現する動作	人を検知	人を検知	人を検知 距離を測定	人を検知	人を検知	温度を測定	湿度を測定	表面温度を測定
配置 2-1	entrance	living	entrance			entrance	entrance	entrance
配置 2-2			entrance	entrance	kitchen	entrance	entrance	entrance
配置 2-3			entrance			entrance	entrance	entrance

表 5 実験 2: アクチュエータの動作及び影響を与える箇所

デバイスのインスタンス名	light actuator 201	light actuator 202	blinds actuator 201	switchBot actuator 204
実現する動作	明るくする 暗くする	明るくする 暗くする	明るくする 暗くする	明るくする 暗くする
配置 2-1	entrance	living		
配置 2-2			entrance	
配置 2-3				entrance

(a) 配置 2-1 (b) 配置 2-2 (c) 配置 2-3

組み合わせられる機能:	組み合わせられる機能:	組み合わせられる機能:
1. エントランスに人がいるかどうかを検知 → 利用する機器: pirSensor201 2. 対象物の表面温度を測定 → 利用する機器: infraredTemperatureSensor204 3. エントランスの温度を計測する → 利用する機器: temperatureSensor202 4. エントランスを明るくする → 利用する機器: lightActuator202 5. センサから対象物までの距離を測定 → 利用する機器: distanceSensor402	1. エントランスに人がいるかどうかを検知 → 利用する機器: camera202 2. 対象物の表面温度を測定 → 利用する機器: infraredTemperatureSensor204 3. エントランスの温度を計測する → 利用する機器: temperatureSensor202 4. エントランスを明るくする → 利用する機器: blindsActuator201 5. センサから対象物までの距離を測定 → 利用する機器: distanceSensor402	1. エントランスに人がいるかどうかを検知 → 利用する機器: distanceSensor402 2. 対象物の表面温度を測定 → 利用する機器: infraredTemperatureSensor204 3. エントランスの温度を計測する → 利用する機器: temperatureSensor202 4. エントランスを明るくする → 利用する機器: switchBotActuator402 5. センサから対象物までの距離を測定 → 利用する機器: distanceSensor402

(a) 配置 2-1 (b) 配置 2-2 (c) 配置 2-3

図 12 実験 2: IoT デバイスの選択結果

したがって、既存のレシピ型の IoT サービス構成手法の課題として挙げた、専門知識の少ない利用者がレシピの適用可否を判断し、IoT デバイスを選択することが難しい点を解決できた。

5.2 実験 2: 異種の IoT デバイスを同一のレシピで利用できることの検証

本実験では、異種の IoT デバイスを同一のレシピで活用でき、提案された IoT サービスが利用者の環境で構成できることを検証する。実験手順は以下の通りである。

- (1) レシピを 1 つ用意し、レシピリポジトリに登録する
- (2) IoT デバイスの配置を 3 種類用意する
- (3) レシピに基づき IoT サービスを構成させる

図 11 に本実験で用いるレシピの概要を示す。このレシピの目標は人の表面温度と気温より人の体温を推定し、その結果を表示すること及び人が検知された場合にエントランスを明るくすることである。表 4 及び表 5 に、それぞれ実験で用いるセンサ及びアクチュエータの動作と観測及び影響を与える箇所を示す。なお、斜線はそのデバイスを配置していないことを示す。

図 12 に、IoT デバイスの配置を変化させた場合の、サービス構成機構による IoT デバイスの選択結果を示す。図

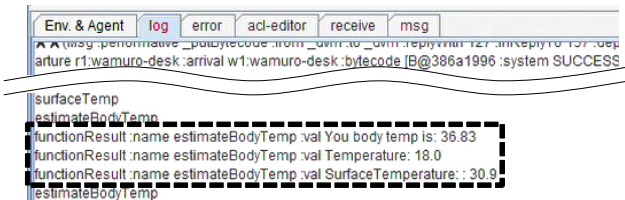


図 13 実験 2: 人の体温推定の出力例

12(a) (b) (c) を確認すると, IoT デバイスの配置を変更した時に「エントランスに人がいるか観測する」ために人感センサ, カメラ, 測距センサがそれぞれ選択されている。また, 「エントランスを明るくする」ためにスマートライト, 電動ブラインド及び壁面の照明のスイッチを On にするスイッチポットが選択されている。また, それぞれの配置時にインターフェイスよりレシピの実行指示を出した際, 選択された IoT デバイスを連携させ, 図 11 のレシピに記述された動作が実行された。

図 13 に, エントランスの気温と人の表面温度を用いて人の体温を推定した結果の例を示す。点線枠で囲んだ部分に, 推定された人の体温が出力されていることを確認した。また, IoT デバイスの配置 2-1, 2-2, 2-3 の全てにおいて, 同様の結果を得られることを確認した。以上の実験結果より, 異種の IoT デバイスを同一のレシピで活用でき, 提案された IoT サービスが利用者の環境で構成できることを確認した。したがって, 既存のレシピ型の IoT サービス構成手法の課題として挙げた, 専門知識の少ない利用者が似た振る舞いの異種の IoT デバイスを持っていても活用できない点を解決できた。

6. おわりに

本稿では, IoT に関する知識が少ない利用者が, IoT サービスを簡単に利用できるようにするため, オントロジを活用した多様な利用者環境に適用可能な IoT サービス構成手法を提案した。また, 試作システムを用いた実験を行い, その有用性を示した。

今後の課題として, より柔軟なレシピの記述を可能とするため, 複数のレシピを組み合わせたレシピの記述を可能とすることが考えられる。また, 利用者にとってより有用な IoT サービスを構成するため, IoT サービスの構成時に利用者の特性を考慮した, 適切な IoT デバイスの選択を実現することが重要であると考えられる。

参考文献

[1] Guth, J., Breitenbücher, U., Falkenthal, M., Leymann, F. and Reinfurt, L.: Comparison of IoT platform architectures: A field study based on a reference architecture, *2016 Cloudification of the Internet of Things (CIoT)*, IEEE, pp. 1–6 (2016).

[2] IFTTT: online: <https://ifttt.com/>.

[3] ConradConnect: online: <https://conradconnect.com/>.

[4] Kim, J. and Lee, J.-W.: OpenIoT: An open service framework for the Internet of Things, *2014 IEEE world forum on internet of things (WF-IoT)*, Ieee, pp. 89–93 (2014).

[5] 和室昂佑, 北形元, 長谷川剛: 抽象度の高いサービス記述に基づく IoT サービスの構成手法の検討, 技術報告 163, 信学技報 (2020).

[6] 溝口理一郎, 池田満, 來村徳信: オントロジー工学基礎論: 意味リンク, クラス, 関係, ロールのオントロジー的意味論 (「オントロジーの基礎と応用」), 人工知能, Vol. 14, No. 6, pp. 1019–1032 (1999).

[7] Bajaj, G., Agarwal, R., Singh, P., Georgantas, N. and Issarny, V.: 4W1H in IoT semantics, *IEEE Access*, Vol. 6, pp. 65488–65506 (2018).

[8] Haller, A., Janowicz, K., Cox, S. J., Lefrançois, M., Taylor, K., Le Phuoc, D., Lieberman, J., García-Castro, R., Atkinson, R. and Stadler, C.: The modular SSN ontology: A joint W3C and OGC standard specifying the semantics of sensors, observations, sampling, and actuation, *Semantic Web*, Vol. 10, No. 1, pp. 9–32 (2019).

[9] SAN: online: <https://www.irit.fr/recherches/MELODI/ontologies/SAN>.

[10] Bonino, D. and Corno, F.: Dogont-ontology modeling for intelligent domotic environments, *International Semantic Web Conference*, Springer, pp. 790–803 (2008).

[11] Soldatos, J., Kefalakis, N., Hauswirth, M., Serrano, M., Calbimonte, J.-P., Riahi, M., Aberer, K., Jayaraman, P. P., Zaslavsky, A., Žarko, I. P. et al.: Openiot: Open source internet-of-things in the cloud, *Interoperability and open-source solutions for the internet of things*, Springer, pp. 13–25 (2015).

[12] Corno, F., De Russis, L. and Roffarello, A. M.: A semantic web approach to simplifying trigger-action programming in the IoT, *Computer*, Vol. 50, No. 11, pp. 18–24 (2017).

[13] ApacheJena: Apache Jena Fuseki. online: <https://jena.apache.org/documentation/fuseki2/>.

[14] mosquito: Eclipse Mosquitto. online: <https://mosquitto.org/>.

[15] LEMONA: PIR Motion Sensor Module:DYP-ME003 SEN005. online: <https://www.lemona.ee/LIUSE/TE/En/Pdf/EF10030.pdf>.

[16] 秋月電子通商: DHT11. online: http://akizukidenshi.com/download/ds/aosong/DHT11_20180119.pdf.

[17] OMRON: 形 D6T MEMS 非接触温度センサ. online: <https://www.omron.co.jp/ecb/product-detail?partNumber=D6T>.

[18] 秋月電子通商: VL53L0X 使用 レーザー測距センサモジュール (ToF). online: <https://akizukidenshi.com/catalog/g/gM-12590/>.

[19] RaspberryPiFoundation: Raspberry Pi Camera Module v2. online: <https://www.raspberrypi.org/products/camera-module-v2/>.

[20] IKEA: FYRTUR フィルトウール. online: <https://www.ikea.com/jp/ja/p/fyrtur-block-out-roller-blind-wireless-battery-operated-grey-90408212/>.

[21] AZUCK: LED 電球スピーカー. online: <https://wowma.jp/item/383425975>.

[22] SwitchBot: SwitchBot bot. online: <https://www.switchbot.jp/bot>.