

Uni-messe と IoT を活用した ハイブリッド会議空間の自動管理 サービスの提案

吉田 健[†] 陳 思楠[†] 中村 匡秀^{†,††} 佐伯 幸郎^{†††}

[†] 神戸大学 〒 657-8501 神戸市灘区六甲台町 1-1

^{††} 理化学研究所・革新知能統合研究センター 〒 103-0027 東京都中央区日本橋 1-4-1

^{†††} 高知工科大学 〒 782-8502 高知県香美市土佐山田町宮ノ口 185

E-mail: [†]yoshitake@es4.edept.kobe-u.ac.jp, ^{††}chensinan@gold.kobe-u.ac.jp, ^{†††}masa-n@cmds.kobe-u.ac.jp,
^{†††}saiki.sachio@kochi-tech.ac.jp

あらまし 近年、我が国では「超スマート社会」が提唱されている。超スマート社会では環境・エネルギー・医療・食糧・災害といった社会的課題を解決するべく、様々な科学技術を用いようとしている。その中で IoT デバイスの活躍が期待されている。2019 年以降、新型コロナウイルス感染症の影響により、ミーティングや会議がオンラインで行われることが多くなった。2023 年 2 月現在では規制が緩和され、オンラインと現地両方で行うハイブリッド会議というものが流行し始めている。しかしながら様々な設備が必要なハイブリッド会議を行うには、準備・撤収に手間がかかる。そこで本研究ではハイブリッド会議の空間づくりに着目し、それらの準備・撤収を IoT デバイスを用いて自動化することができないかと考えた。本研究の目的はハイブリッド会議空間の自動管理を容易に実現するサービスを提案することである。キーアイデアはサービス連携プラットフォーム Uni-messe を利用して配備された IoT デバイスとオンラインサービスを連携することである。Uni-messe を用いることで複数の IoT デバイスとサービスをローコードで自動管理することができる。また、準備段階で実際の会議に必要な設備を想定し、ケーススタディにおいて IoT デバイスを自動管理し正常に動作するかを確かめた。

キーワード IoT, ハイブリッド会議, 超スマート社会, Uni-messe

Automated Management of Hybrid Meeting Space Using Uni-messe

Takeshi YOSHIDA[†], Sinan CHEN[†], Masahide NAKAMURA^{†,††}, and Sachio SAIKI^{†††}

[†] Kobe University Rokkodai-cho 1-1, Nada-ku, Kobe, Hyogo 657-8501 Japan

^{††} Riken AIP 1-4-1 Nihon-bashi, Chuo-ku, Tokyo 103-0027 Japan

^{†††} Kochi University of Technology, 185 Miyanokuchi, Tosayamada, Kami City, Kochi 782-8502, JAPAN

E-mail: [†]yoshitake@es4.edept.kobe-u.ac.jp, ^{††}chensinan@gold.kobe-u.ac.jp, ^{†††}masa-n@cmds.kobe-u.ac.jp,
^{†††}saiki.sachio@kochi-tech.ac.jp

Abstract In recent years, a "super-smart society" has been proposed in Japan. In the super-smart society, various scientific technologies are being used to solve social issues such as the environment, energy, medicine, food, and disasters, in which IoT devices are expected to play an active role. IoT devices are expected to play an active role in this context. As of February 2023, regulations have been relaxed and hybrid meetings, both online and on-site, are becoming popular. However, hybrid meetings require a variety of equipment, which makes the preparation and dismantling time-consuming. In this study, we focused on the creation of a space for hybrid conferences and considered the possibility of automating the preparation and dismantling of such a space using IoT devices. The purpose of this study is to propose a service that facilitates automated management of hybrid meeting spaces. The key idea is to link deployed IoT devices and online services using the service integration platform Uni-messe, which enables low-code, automated management of multiple IoT devices and services. In the preparatory stage, the equipment required for an actual conference was assumed, and the automatic management of IoT devices was verified in a case study to ensure that it worked properly.

Key words IoT, Hybrid Meeting, a smart tech society, Uni-messe

1. はじめに

近年、内閣府により決定された「第五期科学技術基本計画」の中で**超スマート社会**が提唱され、情報通信技術の発達により様々な分野が影響を受けている。また第四次産業革命を超える概念として、**Society5.0**をうたっている [1]。また、超スマート社会の実現に向けて、近年では **IoT 技術**の発展も著しい。IoT とは電子機器や家電などモノの状況をセンサーなどで感知し、インターネットにつなげてそれらのデータを利用してシステムの最適化を図る概念のことを言う。IoT 技術の発展により様々なものがインターネットに接続され、サイバー空間と物理空間との融合が進展している [2]。

そこで本研究では新型コロナウイルス感染拡大により増加した**ハイブリッド会議**に着目した。ハイブリッド会議を行うハイブリッド会議空間を作成しようとする様々な準備が必要となる。そこで、IoT デバイスを導入することでハイブリッド会議空間の準備・撤収を自動管理することを考えた。

本研究の目的は、ハイブリッド会議空間の自動管理を容易に実現するスマートサービスを提案することである。そのキープアイデアとしてサービス連携プラットフォームである Uni-messe を利用してハイブリッド会議空間に配備された複数の IoT デバイスとオンラインサービスを連携することがあげられる。本サービスは以下の三つの要素で構成される。

A1: システム構築フェーズ

A2: 会議登録フェーズ

A3: 会議実行フェーズ

本稿では、提案したサービスの一部を我々の研究室内で簡易的に実装し、準備段階で想定した会議空間にあわせたケーススタディを行った。ケーススタディを通して実際に実装した際には各デバイス間の通信による遅延が発生することや、赤外線非対応の機器が操作できないことからさらなるサービスの改良が必要ことがわかった。

2. 準備

2.1 IoT 技術の発展と超スマート社会

近年、内閣府により決定された、「第五期科学技術基本計画」では「必要なもの・サービスを、必要な人に、必要な時に、必要なだけ提供し、社会の様々なニーズにきめ細かに対応でき、あらゆる人が質の高いサービスを受けられ、年齢、性別、地域、言語といったさまざまな違いを乗り越え、生き活きと快適に暮らすことのできる社会」を「**超スマート社会**」と呼び、その実現を目標に掲げるとともに、第四次産業革命を超える概念として、「**Society5.0**」をうたっている。これからの科学技術は、環境・エネルギー・医療・食糧・災害といった社会的課題解決に大きく貢献すべし、という強いメッセージであり、まさに新しい価値の創出が科学技術に求められている [1]。

また最近では、パソコンや携帯端末だけでなく、電化製品、自動車産業用機械、医療機器など物理的なものが、インターネットに接続され、サイバー空間と物理世界との融合が進展している。その中で、**CPS (Cyber-Physical System)** や

IoT (Internet of Things) といった言葉が生まれ、社会への普及拡大が進んでいる。CPS とは物理世界から得られるデータをサイバー空間に反映させて、コンピュータによる処理と物理的な制御を統合し、制御最適化を行う技術である。IoT とは電子機器や家電などモノの状況をセンサーにより感知し、インターネットにつなげてそれらのデータを集約、分析し、その結果をフィードバックさせることでシステムの最適化を図るという概念である。IoT と CPS の言葉の違いとしては IoT はインターネットを中心としたオープンな環境を想定しているのに対して、CPS は外部ネットワークに接続していない閉じたシステムも連想させるところである。この技術が発展してからその対象領域は交通・製造業・農業・医療・介護といった様々な分野に広がりを見せている [2]。また近年の IoT 化の流れを受け、多様なデバイスを統合管理し、連携を効率化するための土台である **IoT プラットフォーム**が提供されている。IoT プラットフォームと IoT デバイスが連携することにより、各デバイスの仕様や通信方式の違いが取り払われ、共通規格での通信が可能となる。これによりスマートサービス・アプリケーションの生産性が向上し、また分野を超えたデータやサービスの連携が可能になる [3]。

2.2 利用した技術の説明

2.2.1 API と Webhook

API (Application Programming Interface) とはソフトウェアやアプリケーションの一部を公開することにより第三者が開発したソフトウェアと機能を共有できるようにするものである。API を利用するメリットとしてはソフトウェア開発の効率化やセキュリティの向上などがあげられる。その中でも特に **WebAPI** は Web 上に公開されており、外部から利用可能な API のことで、だれでも使用できるかつ、無料であることが多い [4]。

Webhook とはある特定のイベントが発生した際に、そのイベントが発生したことを指定のアドレスなどに通知することで、**イベントトリガー**とも呼ばれる。のちに説明する Pub/Sub もこの Webhook を使っている。API と Webhook の違いは、API ではクライアントが情報をリクエストしなければサーバーからデータの送信が行なわれないのに対して、Webhook では状況の更新などのあらかじめ設定した条件が満たされた場合に自動的にサーバーからクライアントにデータの送信が行なわれる点である。[5]。

2.2.2 Pub/Sub

Publish/Subscribe メッセージ基盤 (Pub/Sub) とはメッセージ、データ、イベントを Publish/Subscribe パターンでやり取りする通信方式のことで、メッセージ送信側と受信側は非同期である。またメリットとしては Publish 側が通信相手を直接認識する必要がないため、データの生成・消費に専念できるという高い疎結合性を有するほか、システム全体の**柔軟性**と**堅牢性**が実現できることである [6]。

Pub/Sub には **Publisher** (メッセージ送信オブジェクト)、**Subscriber** (メッセージ受信オブジェクト)、**Broker** (Publisher と Subscriber を仲介するオブジェクト) の三つのオブジェクト

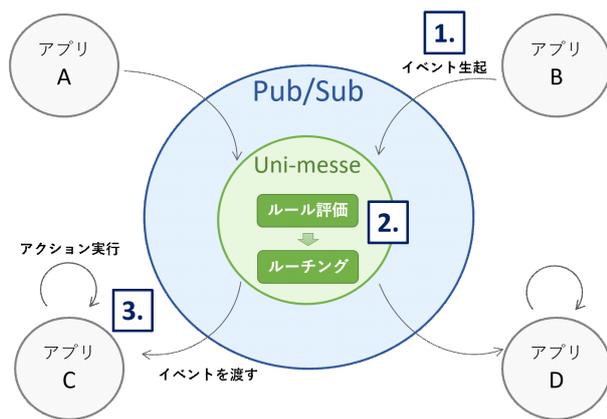


図 1: Uni-messe のアーキテクチャ

が存在する。利用の流れとしては以下のとおりである。

1. ユーザーがあらかじめ Broker に **Topic** を登録する。Topic とはメールの私書箱のようなものである。
2. Subscriber が、Broker に購読する Topic 名や通知先の URL などをリクエストする。これを **Subscribe** と呼ぶ。
3. Publisher が、データやイベントが発生したタイミングで Topic にメッセージを伝える。これを **Publish** と呼ぶ。
4. Topic にメッセージが Publish されると、Broker がその Topic を登録した Subscriber に同メッセージを配信する。

2.2.3 Uni-messe

Uni-messe (Unified Rule-based Message Delivery Service) は、スマートサービスを効率的に実装するためのアプリケーション連携基盤である。スマートサービスに共通する特徴として、複数のサービスを連携することで提供できるサービスの質や量が改善することである。スマートサービス開発では、サービスの連携を効率的に行うための技術が豊富に存在し、それらの技術をサービス体系に組み込むことで、様々なサービスが連携した新たなスマートサービスを提供している。しかしながら、従来の枠組みでは連携可能なサービスが決まっていたり汎用性に欠けるなどの問題からルールベースシステムを中立に扱う基盤は存在しなかった。そこで Uni-messe ではアプリケーションに中立な形式で ECR (Event/Condition/Route) ルールによる管理・評価を行うことで、コストの削減とアプリ開発の効率化を図っている [7]。

図 1 に Uni-messe のアーキテクチャを示した。Uni-messe は、PubSub メッセージ基盤の上に、ルール評価及びルーティングを行うサービスとして構築される。もともとルールベースシステムとは ECA (Event/Condition/Action) ルールに戻って処理を行うが、Uni-messe では Pub/Sub を利用することで Action (サービスを処理すること) ではなく Route (処理を実行するサービスを指示すること) という概念を導入し、サービス同士の疎結合性を高めている。

Step0. アプリの準備 : イベントを発生させるアプリ, 受け取るアプリの作成, 2.2.1 で説明した Pub/Sub の登録する。

Step1. イベント生起 : イベント発生アプリが部屋や人の状況

(コンテキスト) を認識し, Pub/Sub を介して Uni-messe に **6W1H** 形式でイベントを渡す。

Step2. イベントを渡す : Uni-messe が ECR ルールで評価。評価の結果として Pub/Sub 上のトピックを選び選んだすべてのトピックにイベントを渡す。

Step3. アクション実行 : アプリは, Pub/Sub を介してイベントを受け取る。イベントを自由に解釈し, 任意の処理を実行する。

2.3 ハイブリッド会議空間

2019 年以降新型コロナウイルス感染症の影響により世の中では外出を控え, 在宅で勤務する機会が増えた。それに伴い, それ以前は対面により行われていたミーティングや会議もオンラインで行われることが多くなった。現在ではその状況も少し緩和され, オンラインと対面を融合したハイブリッド会議が流行し始めた。

2.4 着目される課題

ハイブリッド会議空間で用いられる機器は曜日, 時刻や気温によっても使用するかどうかが変わってくることもある。具体的な例でいうと「気温が 10 °C 以下ならエアコンをつける」や「夕方は西日がまぶしいのでカーテンを閉める」などがあげられる。このような変化する状況に応じて準備・撤収を一人で行うには負担がおおきい。これらを解決するために IoT デバイスを用いて自動化を図ろうとすると, 各デバイスごとに別々の指令をださなければならない。しかし, そうなると手作業でやるのと大して変わらず完全に自動化できたとは言えない。したがって完全に自動化をおこなうには複数のデバイスに同時に別々の指令をだすプログラミングを行う必要がある。つまりここで着目される課題は以下の二点である。

- P1: ハイブリッド会議空間の準備・撤収に手間がかかる
- P2: ハイブリッド会議空間を自動管理するには複数のデバイスを自動で動かすプログラミング技術が必要である

3. 提案手法

3.1 研究の目的とキーアイデア

本研究の目的は, ハイブリッド会議空間の自動管理を容易に実現するスマートサービスを提案することである。本研究のキーアイデアはサービス連携プラットフォーム Uni-messe を利用してハイブリッド会議空間に配備された複数の IoT デバイスとオンラインサービスを連携することである。

そこで 2.4 の課題を受けて, この目的を達成するために必要とされる要求を以下に示す。

- R1 会議の自動準備: ユーザーが会議室を開けるとその時刻その会議室に適切なハイブリッド会議の設定が行われること
- R2 会議の自動撤収: ユーザーが会議室を退出するとハイブリッド会議が自動的に終了すること
- R3 ノーコードによる会議設定: ユーザーが複雑なプログラミングを必要とせずハイブリッド会議を設定できること

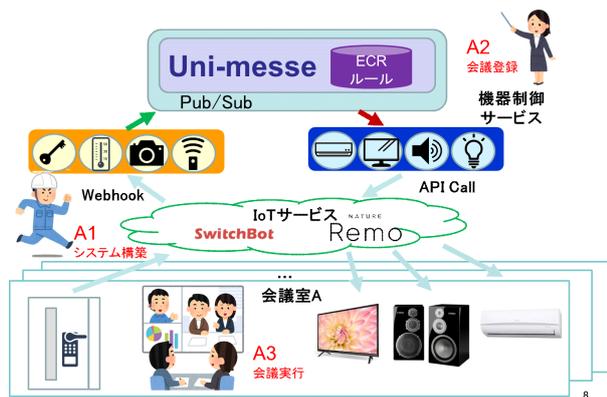


図 2: 全体アーキテクチャ

これらの要求を満たすことで 2.4 の課題を次のように解決する。P1 は R1, R2 によってユーザーは部屋に入退室するだけで準備・撤収が行われるので、解決でき、P2 は R3 を実現することで解決できる。

3.2 全体アーキテクチャ

上記の各要求を満たすためのハイブリッド会議空間自動管理サービスの全体アーキテクチャを図 2 に示した。

このサービスは以下の 3 要素で構成される。

A1: システム構築フェーズ

A2: 会議登録フェーズ

A3: 会議実行フェーズ

3.3 A1: システム構築フェーズ

このシステム構築フェーズではサービス事業者がハイブリッド会議空間に IoT デバイスを配備し Uni-messe と連携可能にする。

3.3.1 IoT デバイスの配備

サービス事業者もしくはサービス利用者がハイブリッド会議で利用する IoT デバイスを用意する。もし必要な機器の IoT デバイスが用意できない場合は代わりに SwitchBot ハブミニ [8] や Nature Remo 3 [9] などのネット制御が可能なデバイスを既存の機器に設置する。

3.3.2 中継サービスの構築

2.2.3 の Step1 で説明した通り Uni-messe では 6W1H 形式でないイベントを評価できない。しかしながら IoT デバイスから送られてくるイベントはそれぞれのデバイスによって異なり、そのほとんどが 6W1H 形式ではない。そこで 2.2.3 の Step0 でも説明したように、まず初めに Webhook により IoT デバイスから送られてくるイベントを Uni-messe に適した 6W1H 形式に変換して Pub/Sub に送信するサービスを構築しなければならない。このサービスをイベントハンドラサービスという。

また、その後 Uni-messe がルーチングした後のメッセージを受け取り、各種 IoT デバイスに適したコマンドに変換するサービスも必要である。このサービスを機器制御サービスという。

3.4 A2: 会議登録フェーズ

このフェーズではユーザーが配備された IoT デバイスに対して、Topic の管理を Pub/Sub で行い、Condition と Route の登録を Uni-messe で行う。

3.4.1 Topic の管理

ユーザーは部屋ごともしくは用途ごとに複数の Topic を Pub/Sub で管理する。例えば会議室、多目的室、休憩室などの部屋が複数ある場合には「会議室 OPEN」「会議室 CLOSE」などそれぞれの部屋ごとに鍵が開いたときと閉まったときの Topic を分けて作る。また時間によって使用の有無が変わる家電などがある場合も、それ専用「会議室 17 時以降 OPEN」など Topic を作る。ここで作った Topic が後の Route の登録の際に必要なになってくる。

3.4.2 Route の登録

Uni-messe の Route 登録は以下のように行う。

RouteID: この Route の ID。Uni-messe が自動で振り分ける。Condition とひもづけられる。

topicList: 3.4.1 で PubSub に登録した Topic を入力する。

data, attributes: 送られてきたデータの中身の書き換えができる。

owner: 管理者の名前を登録する。

application: 利用するアプリの名前を登録することで検索に利用できる。

3.4.3 Condition の登録

Uni-messe の Condition 登録は以下のように行う。

routeidList: この Condition に紐づける Route の ID を入力する。

how, who, whom, where: 演算子を「常に真, 完全一致, 不一致, 部分一致, 包含」の中から一つ選択して登録, condition には deviceId など各項目に適したものを登録する。

when: cron 式で記述する。

what: Java 論理式を記述, JSON キーバリューを条件式に当てはめて評価する。

why: Uni-messe が自動で記述する (スタックトレース)。

owner: 管理者の名前を登録する。

application: 利用するアプリの名前を登録することで検索に利用できる。

3.5 A3: 会議実行フェーズ

図 2 のアーキテクチャの流れは以下のとおりである。

1. ユーザーが起こしたアクションに反応して IoT デバイスから Event がイベントハンドラサービスに送られる。
2. 送られてきた Event は、イベントハンドラサービスによって Uni-messe に適した 6W1H 形式に変換され、Pub/Sub に送られる。
3. Pub/Sub は受け取った Event をすべて Uni-messe に送る。
4. Uni-messe が設定した Condition に基づいて評価、のち

Route を行い指定の Topic に送る。

- Pub/Sub が Topic に連携している機器制御サービスに送り機器制御サービスが IoT デバイスを操作する。

つまり A2 までの作業が完了していればユーザーが会議室などの扉を開けた際に指定の IoT デバイスが作動するようになっている。

4. ケーススタディ

4.1 利用した技術

開発の際には言語は Python 3 を利用し、またイベントハンドラサービスや機器制御サービスには FastAPI [10], Pub/Sub は研究室で別途実装済みのものを利用した。

ケーススタディでは本サービスの実際の登録の流れや IoT デバイスの連携について知るために研究室で一部実装を行った。今回想定したケースとしては、ある部屋の鍵が開いた際に自動で電気、換気扇、TV がつくように登録を行った。

4.2 A1: システム構築フェーズの実現

4.2.1 IoT デバイスの配備

今回は研究室内の部屋の扉に SwitchBot アプリから解錠・施錠することのできる SwitchBot ロックを設置し、電気二つと換気扇を動かす用に SwitchBot アプリから操作が可能な SwitchBot ボット三つを用意した、またテレビを操作するために SwitchBot ハブミニを用意した。SwitchBot ハブミニは赤外線を利用して赤外線対応の家電を操作することができる他に、SwitchBot デバイスをインターネットに接続することができる。その後、SwitchBot アプリのインストール、アカウントを作成し、SwitchBot アプリから図 4 のように SwitchBot ハブミニをアカウントに追加後、各デバイスを接続した。そしてそれらのデバイスを図 3 のように設置した。



(a) SwitchBot ボット



(b) SwitchBot ロック

図 3: SwitchBot デバイス

4.2.2 中継サービスの構築

a) イベントハンドラサービス

まずは SwitchBot 社の提供する API の中から Jupyter で Webhook を構築した。その際に通知先として URL が必要だったため、FastAPI を利用して作成した。次に Webhook で飛んできた情報を処理するイベントハンドラサービスを立ち上げた。イベントハンドラサービスは送られてきた情報を以下のように変換する。

data: イベント内のデータから必要な形に成形、評価には影響しない。



図 4: SwitchBot アプリでの登録

publisher: イベントがどのサービスから飛んできたのかを示す。評価には影響しない。

topic: イベントを Pub/Sub のどの Topic に渡すか、ここでは全て unimesse.event に渡す。

how: イベントを検知したサービス・センサの名前・ID。イベント内のデバイスの種類を抽出する。

who/whom: イベントを起こした/起こされた、人や機器の名前。イベント内のデバイス ID を抽出する。

where: イベントが発生した場所。今回は場所を示すイベントがないためデバイス ID で代用する。

when: イベントが発生した時刻 (ISO8061 形式)。イベント内のタイムスタンプを変換する。

what: 詳細なイベントの説明。イベント内のデバイスの状態を JSON 形式に変換する。

why: イベントが発生してきた経緯をたどるスタックトレース (リスト形式)。Uni-messe が自動で付与、評価には影響しない。

b) 機器制御サービス

Uni-messe から受けとった際に対象の IoT デバイスに適切な指示を送るための API を呼び出す。API を呼び出す際には動かすデバイスの ID、紐づけられているアカウントの ID、指示の内容 (各デバイスによって形式が異なる) が必要。

4.3 A2: 会議登録フェーズの実現

今回のケーススタディでは「smartspace_door_open」「smartspace_door_close」というふたつの Topic を新たに作成した。また Route は図 5、Condition は図 6 のように登録を行った。



図 5: Route の登録例

4.4 A3: 会議実行フェーズの動作

実際にスマートスペースのドアを開けると以下のようなデータが実際に飛んてくる。

routeIdList	operator	condition
how	完全一致	WoLock
who	operator	condition
whom	完全一致	D8A429142B0C
where	完全一致	D8A429142B0C
when	condition	*****
what	condition	lock_status=="UNLOCKED"
isActive	true	
owner	yoshihake	
application	application	est_door

図 6: Condition の登録例

実際に送られてきたイベント

```
{
  "eventType": "changeReport",
  "eventVersion": "1",
  "context": {
    "deviceType": "WoLock",
    "deviceMac": "D8A429142B0C",
    "lockState": "LOCKED",
    "timeOfSample": 1672991842220
  }
}
```

イベントハンドラサービスで変換した後のイベント

```
{
  'data': 'WoLock D8A429142B0C is LOCKED',
  'publisher': 'sbot2unimesse',
  'topic': 'unimesse.event',
  'attributes': {
    'how': 'WoLock',
    'whom': 'D8A429142B0C',
    'where': 'D8A429142B0C',
    'when': '2023-01-06T16:57:22',
    'what': '{"lock_status": "LOCKED"}'
  }
}
```

上記の変換後のデータが Uni-messe におくられたのち、図 6 で評価され、実際の IoT デバイスが動くことが確認された。

4.5 考 察

以上により、本サービスを用いたハイブリッド会議空間の自動管理の流れ・動作を確認した。2.4 の P1 に関しては自動化をおこなうことによって準備・撤収にかかる時間は確実に短縮したといえる。また P2 に関してもユーザーとサービス事業者という二つに分けることでユーザー側はノーコードで実装することに成功した。また、この提案手法のほかの利点としては、Uni-messe を用いることで IoT デバイス同士の連携だけでなく、Web サービスの中で API を提供しているものがあればそれとも連携がおこなえるという点にある。一方で、今回提案したサービスの限界として、鍵を開けてから家電が操作されるまでに数秒かかることがあげられる。今回のケーススタディでは各スイッチに SwitchBot ボットをとりつけて操作させたので

SwitchBot ハブミニとの通信が複数回おこなわれることから数秒の遅延が発生したと思われる。イベントが起きてからアクションを行うまでの時間短縮は今後の研究の課題としたい。また、ユーザーが新しく家電を追加する際には、そのデバイスに合わせたセットアップを行わなければならない。この問題に対してはサービス事業者側が事前に機器毎に登録マニュアルを作成するか、その登録の段階も含めたアプリを開発することによりユーザー単体で登録が行えるようにするという解決策が考えられたが、これについても今後の課題としたい。

5. ま と め

本研究ではハイブリッド会議空間の自動管理を容易に実現するスマートサービスの提案を目的とし、Uni-messe と複数の IoT デバイス自動管理サービスを提案した。システム構築フェーズでは、イベントハンドラサービス、機器制御サービスを作成することで複数の IoT デバイスを同時に管理することを可能とし、会議登録フェーズでは Uni-messe を用いることでユーザーがノーコードで管理できるようになっている。本研究ではケーススタディとして実際の会議を想定して入室をトリガーとしていくつかの家電を操作した。今後の課題としては入退出後の家電操作の遅延を減らすことと、ユーザーがノーコードで実装できるようにサービスを改善することを考えている。

謝 辞 本研究の一部は JSPS 科研費 JP19H01138, JP20H05706, JP20H04014, JP20K11059, JP22H03699, JP19K02973, 特別研究員奨励費 22J13217, および、立石科学技術振興財団の研究助成を受けて行われている。

文 献

- [1] 原 辰次 C 本多 敏 C “超スマート社会におけるシステム科学技術概論 C” 計測と制御 C vol.55Cno.4Cp.284-287C2016D
- [2] 山田直史 C 高島洋典 C 木村康則 C “超スマート社会 (society5.0) 実現に向けて: cps / iot とその後 C” 情報管理 C vol.60Cno.5Cp.325-334C2017D
- [3] 中橋友郎 C 陳 思楠 C 中村匡秀 C “異種 iot とプラットフォームの連携を容易化するサービスの研究開発 C” 電子情報通信学会技術研究報告 C æ121³C pp.45-52CMarch 2022D オンライン D
- [4] “今さら聞けない it 用語 やたらと耳にするけど「api」って何?,” <https://data.wingarc.com/what-is-api-16084CDec.2022D> (Accessed on 01/31/2023)D
- [5] “Webhook とは? api との違いや対応サービスについて,” <https://it-kyujin.jp/article/detail/402/D> (Accessed on 01/31/2023)D
- [6] 坂野遼平 C 首藤一幸 C “pub/sub メッセージングにおける負分散性と低遅延性の適応的制御 C” IEICE Conferences ArchivesThe Institute of Electronics, Information and Communication EngineersC pp.●●●●2019D
- [7] 中田匠哉 C 中村匡秀 C “コンテキストに基づくサービス連携を支援するイベントルーチングサービスの検討 C” 電子情報通信学会技術研究報告 C æ121³C pp.32-37CAug. 2021D オンライン開催 D
- [8] “Switchbot (スイッチボット),” <https://www.switchbot.jp/D> (Accessed on 01/31/2023)D
- [9] “Nature remo (ネイチャーリモ),” <https://nature.globa1/nature-remo/D> (Accessed on 01/31/2023)D
- [10] “Fastapi - tiangolo,” <https://fastapi.tiangolo.com/ja/> (Accessed on 02/08/2023).