

ハイブリッド会議空間自動管理のための遠隔タスク実行サービスの実装

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.eeddept.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 本研究では、ハイブリッド会議空間の自動管理を実現するスマートサービスの設計と実装について述べる。現代の会議室では、オンラインとオフラインの両方の参加者が交流するハイブリッド形式が一般的になりつつある。この新しいニーズに対応するため、本研究では VNC と WebDriver を利用した遠隔タスク実行サービスを開発する。このサービスは、会議の準備の際のオンラインサービスの起動、Web サイトの操作を自動化し、利用者の手間を大幅に削減する。具体的には、Zoom の起動、議事録用 Web サイトの起動など、複数の遠隔操作タスクを自動実行する機能を提供する。本稿では、研究室でのプロトタイプ実装を通じて実際の動作を確かめる。

キーワード IoT, ハイブリッド会議, Society 5.0, Uni-messe, VNC, Selenium , タスク自動化

Implementing Remote Task Execution Service for Hybrid Meeting Spaces

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.eeddept.kobe-u.ac.jp, ^{††}chensinan@gold.kobe-u.ac.jp, ^{†††}masa-n@cmds.kobe-u.ac.jp,
^{††††}saiki.sachio@kochi-tech.ac.jp

Abstract This study describes the design and implementation of a smart service for automated management of hybrid meeting spaces. Hybrid formats in which participants interact both online and offline are becoming common in modern meeting rooms. To meet this new need, we develop a remote task execution service using VNC and WebDriver. This service will automate the launching of online services and website operations when preparing for a meeting, greatly reducing the time and effort required by the user. Specifically, the service provides functions to automatically execute multiple remote control tasks, such as launching Zoom, launching a Web site for meeting minutes, and so on. In this paper, we confirm the actual operation of the system through a prototype implementation in our laboratory.

Key words IoT, Hybrid Meeting, Society 5.0, Uni-messe, VNC, Selenium , Task automation

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と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

実現するためのスマートサービスを実装することである。本稿では先行研究の課題であったオンラインサービスとの連携を解決するために遠隔タスク実行サービスを実装した。本サービスのキーアイデアは **WebDriver** および **VNC** サーバーを遠隔 PC に配備し、コードを用いてタスクを実行することである。本サービスは以下の三つのフェーズで構成される。

A1: リモート環境構築フェーズ

A2: タスク登録フェーズ

A3: タスク実行フェーズ

A1 ではハイブリッド会議で利用する遠隔 PC にサーバーから遠隔で操作を行うために必要なソフトをダウンロードする。A2 では遠隔タスクの種類を定義し、それを実行する条件とルートの紐づけをノーコードで行う。その後 A3 で条件を満たす行動をユーザーが行うことで遠隔タスクが実行される。

本稿では、提案したサービスを我々の研究室内で簡易的に実装し、Zoom の起動、議事録用サイトの起動を行った。

2. 先行研究

2.1 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ことが特徴としてあげられる。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 [3]。

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 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的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 [4]。

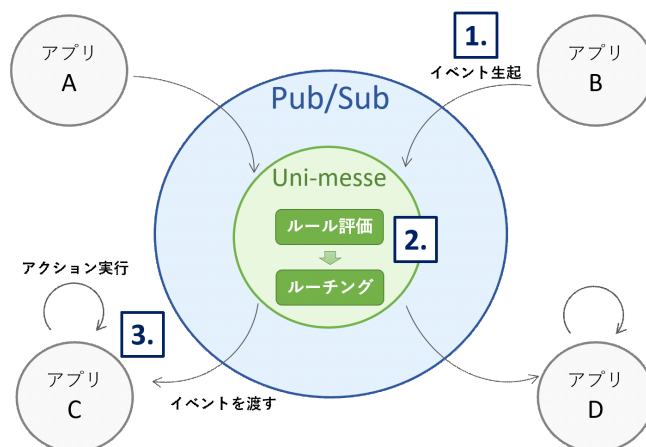


図 1: Uni-messe のアーキテクチャ [4]

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、2.1 で説明した Pub/Sub の登録する。

Step1 イベント発生: 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自動管理サービス

ハイブリッド会議空間自動管理サービスとは、ハイブリッド会議の準備に必要な様々な機器を IoT デバイスと Uni-messe を導入することで、ハイブリッド会議空間の構築を自動化するためのサービスである。全体アーキテクチャを図 2 に示す。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

A1 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, 各種 IoT デバイスの登録, IoT デバイスが指定のルールに沿って動作するよう設定を行う. A2 ではユーザーが Uni-mesese を介して各種 IoT デバイスの ECR ルールをノーコードで設定し会議の登録を行う. A3 の実行フェーズで実際に実行すると A1 で配備したデバイスが A2 の ECR ルールに沿って稼働する.

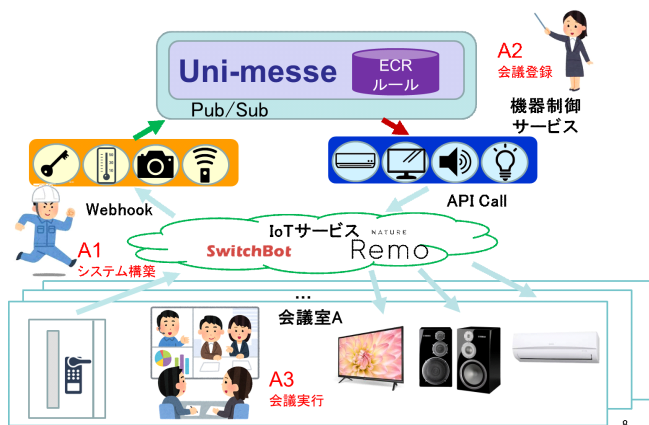


図 2: ハイブリッド会議空間自動管理サービスの全体アーキテクチャ

2.4 先行研究の課題

先行研究の段階では主に二つの課題が残った. 一つ目は, 扉を開閉してから IoT デバイスが起動するまでの時間に遅延が生じていたことだ. しかし, 現在の SwitchBot の API の一種である Webhook [5] を利用してサーバーに接続している仕様上, イベントに対する反応は今の速度が限界だと判明した. 二つ目は, オンラインサービスとの連携基盤が実装に至らなかった点である. 先行研究時点では簡易的な実装のみにとどまり, オンラインサービスとの本格的な連携は行っていなかった. また, 当初の構想では, サーバーから会議室の遠隔 PC にオンラインサービスを起動及び実行する指示を送信するために, 遠隔 PC にコードを配置することを考えていたが, この方法はサーバー間の通信が増えるため実行の遅延につながると考えられた. また, プログラミングを利用しないユーザーにとっては環境構築の負担がかかるという, 新たな問題が発生した. つまり, 本研究で扱う課題としては以下の二点である.

P1: なるべく少ない回数の通信で遠隔タスクを実行する

P2: リモートでタスク実行をするための PC の環境構築に手間がかかる

3. 遠隔タスク実行サービス

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実現するためのスマートサービスを実装することである. 本研究のキーアイデアは WebDriver と VNC クライアントを遠隔 PC に配備し, サーバー側から Web 自動化及び画面操作のコマンドを組み合わせて送ることにより, 様々な遠隔タスクを実行す

ることである.

3.2 機能要件

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: ユーザーが実行したいタスクをリモートサーバーから実行できる

R2: ユーザーは複雑な環境構築をせずに利用できる

R3: ユーザーがタスクを実行するための条件をノーコードで指定できる

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 を満たすことですべての操作がリモートから行われるため解決でき, P2 は R1, R3 を実現することで解決できる.

3.3 全体アーキテクチャ

上記の各要求を満たすための遠隔タスク実行サービスの全体アーキテクチャを図 3 に示した.

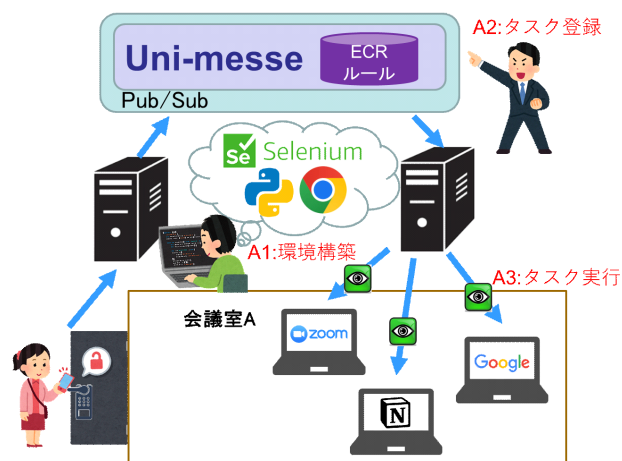


図 3: 遠隔タスク実行サービスのアーキテクチャ

このサービスは以下の 3 要素で構成される.

A1: リモート環境構築フェーズ

A2: タスク登録フェーズ

A3: タスク実行フェーズ

3.4 A1: リモート環境構築フェーズ

3.4.1 IoT デバイスの配備

サービス事業者もしくはユーザーがハイブリッド会議で利用する IoT デバイス, 遠隔 PC を用意する. また, Switchbot ボットのみで操作不可能なデバイスに関しては, Switchbot ハブミニ [6] や Nature Remo [7] などのネット制御可能なデバイスを既存の機器と連携させ赤外線を用いて操作する.

3.4.2 遠隔 PC の整備

遠隔 PC にタスクを実行させるには遠隔 PC をサーバーから制御するための環境を作成する必要がある. そこで遠隔 PC には VNC サーバーと WebDriver を配備する.

a) VNC

Virtual Network Computing(VNC) とは, RFB プロトコルを使用して画面を送信し, かつ操作権限を与えるリモートデスクトップ技術の一種である. VNC はサーバー側とクラ

クライアント（ビューア）側に分かれており、VNC クライアントソフトウェアをクライアント側にインストールし、アクセス権限を持つユーザーがサーバー側から接続を行うことで、遠隔操作を可能にする [8]。比較対象として Windows のリモートデスクトップが挙げられる。リモートデスクトップはマイクロソフト社が Windows OS に直接組み込んでおり、シームレスな統合と高品質なビデオパフォーマンスが特徴として挙げられるが、他の OS との互換性がない。一方で VNC は RFB プロトコルを利用しているため、OS 間での互換性がある。また、リモートデスクトップはクライアント側の PC を操作してしまうとリモート接続ができなくなってしまうが、VNC はクライアントの操作に依存しない。さらに、コードを用いた操作の点でも VNC の方が優れている。

本研究では、異なる OS のクライアント PC を操作対象とする可能性があり、クライアント側の PC で操作するケースも存在することを考慮し VNC を利用した。

b) WebDriver

Selenium WebDriver は、Web ブラウザを自動的に制御するためのフレームワークであり、Java、Python、JavaScript などのプログラミング言語向けに API を提供している。このツールは、Web アプリケーションの自動ナビゲーションやユーザーログインなどの機能を含んでおり、開発者に幅広い操作の柔軟性を提供する。

Selenium WebDriver は、内部的には各ブラウザに固有の自動化サポートを活用するため、プラットフォーム依存のバイナリファイル（ドライバ）を必要としている。これらのドライバは、Selenium WebDriver スクリプトと対象のブラウザの間で通信を可能にし、自動化テストの実行を可能にする [9]。

3.5 A2: タスク登録フェーズ

このフェーズではユーザーが実行したいタスクを定義し、タスクを実行する際の日時や場所などの条件を指定する。それを受けたサービス事業者が実行したいタスクに沿ったコードを書き、ユーザーが Pub/Sub で Topic の管理を行い、Uni-messe に ECR ルールを登録する。

3.5.1 タスク登録

a) Topic の管理

ユーザーは会議室ごと、もしくは用途ごとに複数の Topic を作成し Pub/Sub で管理する。例えば会議室 A についての Topic を作成する場合は、「会議室 A __ OPEN」「会議室 A __ CLOSE」「会議室 A __ MEETING __ START」「会議室 A __ MEETING __ STOP」などその部屋の状態を表現した Topic を作成する。ここで作成した Topic はのちの Route 以降の登録の際に必要なとなる。

b) Route の登録

Uni-messe の Route の登録は以下のように行う。

Route に書く内容

- **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**: a) 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。

c) Condition の登録

Condition に書く内容

- **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。

3.5.2 タスクの実行準備

タスクの実行は実行するコマンドによってコードを書き換える必要がある、ここでは例として Zoom を起動する場合とユーザーログインが必要なサイトを起動する場合に分けて説明する。

a) Zoom の起動

Zoom の起動を Web から行う場合事前にユーザーから URL を聞き取り、コード内に記述する。また、Web ブラウザ上に URL を入力するだけでは Zoom は起動せず、起動する際にアプリに移行するためブラウザ外の操作が必要となる。しかし、本研究で利用する Selenium は単体ではブラウザ外の操作はサポートしておらず今回の Zoom の起動といった OS レベルでの操作には対応できない。そこで、VNC を用いたマウス操作、キーボード操作を利用して、ブラウザ外の参加ボタンやアプリ上のボタンを押している。

b) ログインが必要なサイト

ログインが必要なサイトについてはそのログイン方法によって操作方法が変わるのでコードを変える必要がある。ログイン機能がブラウザ内部に実装されている場合は Selenium の要素を検索する機能から UserID, Password を書き込むと実行できるが、Google のログインやブラウザ外部の操作が必要なログインの場合個別のコードを実装する必要がある。

3.6 A3: タスク実行フェーズ

図4に実際にZoomを起動するタスクを行った際のシーケンス図を示した。

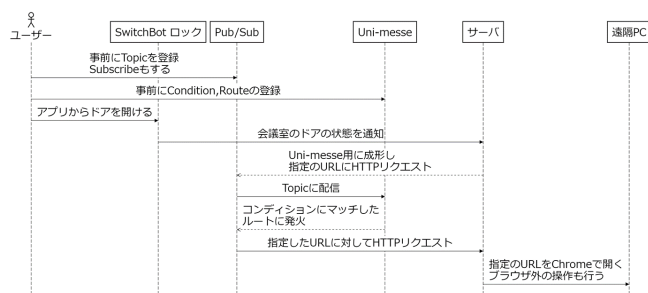


図 4: Zoom を起動する際のシーケンス図

4. 实 装

4.1 利用した技術

開発の際には言語は Python 3 を利用し、フレームワークには FastAPI [10], Pub/Sub, Uni-messe は研究室内で別途実装済みのもので利用した。リモート実行環境作成には UltraVNC [11], Web ブラウザの操作は Selenium-server [12], ChromeDriver, Chrome を利用して行った。

本研究では本サービスの実際の登録の流れや PC の動きを確認するために研究室内で実装を行った。想定したケースとしては、指定された日時に、ある部屋の鍵が解錠された際に Zoom の起動、議事録用サイトの起動、ログインを行った。

4.2 A1：リモート環境構築フェーズの実現

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図 5b のように SwitchBot ロックを設置し、図 5a のように SwitchBot アプリとの連携をおこなった。その後、遠隔用 PC に UltraVNC, Chrome, ChromeDriver, Selenium-server をそれぞれ公式ホームページからダウンロードした。UltraVNC に関してはパスワードの設定と接続 IP の確認を行い、常時バックグラウンドで起動するように設定した。また、Selenium-server に関しては PowerShell 上で起動コマンドを実行した。

4.3 A2:タスク登録フェーズの実現

本研究では多目的部屋の開閉を知らせるために「multipurpose __ room __ door __ open」「multipurpose __ room __ door __ close」の二つの Topic を作成した。

また、今回のケースでは火曜日の朝 8-10 時に入室した際に Zoom 起動、議事録用サイト起動、ログインの三つのタスクが実行されるように、Route は図 6、Condition は図 7 とした。



(a) SwitchBot アプリ

(b) SwitchBot ロック

図 5: SwitchBot デバイス

#12

topicList	multipurpose_room_door_open 
data	data
attributes	attributes { }
owner	owner * yoshitake
application	application * multipurpose_door
createdAt	2024/1/23 22:14:20

更新

図 6: Route の登録

4.4 A3：タスク実行フェーズの動作

実際に起動し、指定の部屋に入室を行うと以下のようなイベントが送られてくる。これを変換しサーバーから Uni-messe に送信し、条件に適した場合にタスクが起動する。

実際に送られてきたイベント

```
{
  "eventType": "changeReport",
  "eventVersion": "1",
  "context": {
    "deviceBattery": "95",
    "deviceType": "WoLock",
    "deviceMac": "DEB096F5176C",
    "lockState": "UNLOCKED",
    "timeOfSample": 1708328204728
  }
}
```

図 8 に本サービスを利用して実行した際の画面を示した。

#7

routeIdList 12

how operator condition

who operator condition

whom operator 完全一致 condition DEB096F5176C

where operator 完全一致 condition DEB096F5176C

when condition **8-10**1

what condition lock_status=="UNLOCKED"

isActive true

owner owner * yoshitake

application application * zoom_rule

createdAt 2024/2/16 16:52:50

更新 再読み込み 削除

図 7: Condition の登録

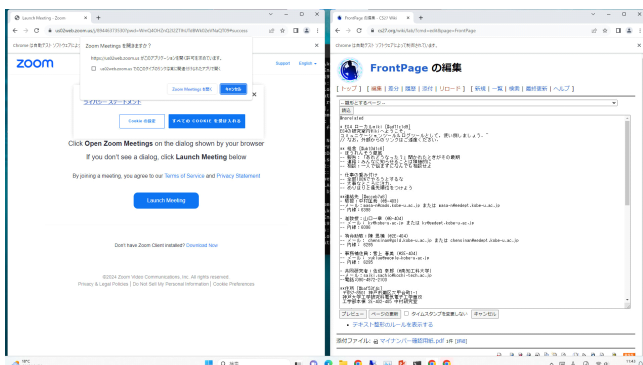


図 8: 左: Zoom の起動 右: 議事録用サイトの起動

4.5 考 察

本研究により、提案されたサービスを用いて遠隔タスクの登録及び実行が効果的に行えることが確認された。問題点 2.4 に記載された P1 および P2 については、遠隔 PC に追加のコードを設置することなく、最小限の環境構築により対応可能であった。特に、本提案手法の顕著な利点は、他の既存タスク自動化サービスと異なり、ブラウザ外の操作も遠隔サーバーから実行可能である点にある。これにより、ブラウザに対応していない Windows アプリケーションの実行も可能となる。

しかし、提案されたサービスにはセキュリティ面での懸念や、ユーザーによる設定変更の制限といった限界も存在する。具体的には、ID やパスワードをコード内に直接記述する設計は、コードが外部に漏れた場合にユーザー情報が露出するリスクを伴う。また、URL、ID、パスワードの登録をコード上で行うため、ユーザーが容易にこれらを変更することができない。これらの問題点は、データベースを活用し、URL、ID、パスワード

を登録できるアプリケーションの開発により、将来的に解決することが期待される。

5. ま と め

本研究はハイブリッド会議空間自動管理のための遠隔タスクサービスの実装を目的とし、遠隔 PC に VNC、WebDriver 等を配備することで環境構築を最小限に抑えた遠隔タスクを実装した。本サービスはリモート環境構築フェーズ、タスク登録フェーズ、タスク実行フェーズの三つで構成されている。リモート環境構築フェーズでは、遠隔 PC に WebDriver、VNC の配備を行い、タスク登録フェーズでは Topic の管理、Route、Condition の登録を行った。今後の課題としてはセキュリティ面の強化やユーザーが単独で設定できるアプリケーションの開発を考えている。

謝 辞 本研究の一部は JSPS 科研費 JP19H01138, JP20H05706, JP20H04014, JP20K11059, JP22H03699, JP19K02973, 若手研究 23K17006 の助成を受けて行われている。

文 献

- [1] 原 辰次, 本多 敏, “超スマート社会におけるシステム科学技術概論,” 計測と制御, vol.55, no.4, pp.284–287, 2016.
- [2] 山田直史, 高島洋典, 木村康則, “超スマート社会 (society5.0) 実現に向けて: cps / iot とその後,” 情報管理, vol.60, no.5, pp.325–334, 2017.
- [3] 坂野遼平, 首藤一幸, “pub/sub メッセージングにおける負荷分散性と低遅延性の適応的制御,” IEICE Conferences Archives The Institute of Electronics, Information and Communication Engineers, pp.●●●●●2019.
- [4] 中田匠哉, 中村匡秀, “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,” 電子情報通信学会技術研究報告, 第 121 巻, pp.32–37, Aug. 2021. オンライン開催.
- [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/>. (Accessed on 19 Feb 2024).
- [6] “Switchbot (スイッチボット),” <https://www.switchbot.jp/>. (Accessed on 19 Feb 2024).
- [7] “Nature remo (ネイチャーリモ),” <https://nature.global/nature-remo/>. (Accessed on 19 Feb 2024).
- [8] 谷成雄, 大城信康, 河野真治他, “Vnc を用いた講義用画面共有システムの設計・開発,” 研究報告システムソフトウェアとオペレーティング・システム (OS), vol.2012, no.19, pp.1–5, 2012.
- [9] 関本大樹, セキモトタイジユ, “Selenium を用いた授業管理業務の簡略化について,” 久留米大学コンピュータジャーナル, vol.36, pp.2–13, 2022.
- [10] “Fastapi - tiangolo,” <https://fastapi.tiangolo.com/ja/>. (Accessed on 19 Feb 2024).
- [11] “Ultravnc vnc official site,” <https://uvnc.com/>. (Accessed on 19 Feb 2024).
- [12] “Selenium ブラウザー自動化プロジェクト,” <https://www.selenium.dev/ja/documentation/>. (Accessed on 19 Feb 2024).