

機能の番号割り当てによる遠隔操作ボタンシステムの集約と効率化

村手 亮太[†] 陳 思楠[†] 佐伯 幸郎^{††} 中村 匡秀^{†††}

[†] 神戸大学 〒657-8501 神戸市灘区六甲台町 1-1

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

あらまし 日本社会は内閣府により情報社会 Society 4.0 と定義され、インターネットやスマートフォンの普及によりネットワークで世界が繋がる社会とされる。近年、これを発展させた超スマート社会 Society 5.0 が提唱され、「必要なものを必要な人に必要な時に提供し、誰もが快適に暮らせる社会」と定義されている。このような社会の中、様々な生活支援サービスが誕生しており、その課題は多様なサービスへの容易なアクセスである。本研究では、ボタンを押すだけで利用可能な「ボタン駆動スマートサービス」に着目する。先行研究 TapTrack は、ボタン駆動スマートサービスの実装を容易にし、開発基盤を提供することを目的として提案された。TapTrack は「ボタン機能生成」「機能割り当て」「ボタン駆動」「振り返り」の四つの機能を持ち、他サービスとの連携や機能集約を実現している。しかし、機能集約により画面の情報量が増え、目的のボタンを探すのが困難になる問題が生じる。そこで、本稿ではスマートフォン操作でサービス実行を可能にする「DigiTrack」を開発し、この課題を解決する。DigiTrack は「ユーザー情報取得」「UI 提供」「コマンド操作」「サービス提供」の四つのアーキテクチャを持つ。

キーワード ボタン駆動スマートサービス、番号割り当て、スマートホーム、ユーザーインターフェース

A Study on Platforms Facilitating the Implementation of Button-Driven Smart Services

Ryouta MURATE[†], Sinan CHEN[†], Sachio SAIKI^{††}, and Masahide NAKAMURA[†]

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

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

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

Abstract Japanese society is defined as an “Information Society” by the Cabinet Office, characterized by a globally connected network enabled by the proliferation of the internet and smartphones. Recently, the concept of a “Super Smart Society (Society 5.0)” has been proposed, defined as “a society where necessary goods and services are provided to the right people at the right time, enabling everyone to live comfortably.” In this context, various life-support services have emerged, with the key challenge being how to ensure easy access to diverse services. This study focuses on “button-driven smart services,” which are accessible with a single button press. The prior study, TapTrack, was proposed to simplify the implementation of such services and provide a development platform. TapTrack incorporates four functionalities: “button feature generation,” “feature assignment,” “button-driven operation,” and “review,” facilitating service integration and collaboration. However, the consolidation of features often increases on-screen information, making it difficult for users to locate their desired buttons. To address this issue, this paper presents the development of “DigiTrack,” which enables service execution through smartphone operations. DigiTrack is built on four key architectural components: “user information acquisition,” “UI provision,” “command execution,” and “service delivery.”

Key words Button-Driven Smart Service, Number Assignment, Smart Home, User Interface

1. はじめに

現在の日本社会は、内閣府により情報社会 (Society 4.0) [1] と定義づけられている。情報社会とは、インターネットやスマートフォンなどの通信技術の普及により、世界がネットワークで繋がる社会である。そして近年より、情報社会を発展させた超スマート社会 (Society 5.0) [2] が提唱されている。超スマート社会は、「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、言語といった様々な違いを乗り越え、生き活きと快適に暮らすことのできる社会」であると定義されている。このような第4次産業革命といえる社会情勢の中、人々の生活を支援するサービスが数多く生まれている。その中での課題は、多様なサービスにいかにか容易にアクセスできるかである。

本研究では解決方法として、ボタンの押下により利用可能なスマートサービス「ボタン駆動スマートサービス」に着目する。先行研究 TapTrack ではボタン駆動スマートサービスの実装困難性に対してボタン駆動スマートサービスの実装を容易化し実行基盤を提供すること目的として、ボタン駆動スマートサービス開発実行基盤「TapTrack」の提案というアプローチを行った。TapTrack はこの要件を満たすため、「TapTrack-A1: ボタン機能生成」「TapTrack-A2: ボタン機能割り当て」「TapTrack-A3: ボタンによる駆動」「TapTrack-A4: 振り返り」の四つの機能を持つ。この実装により TapTrack には非デジタル化された要素をソフトウェア化・仮想化すること、他のサービスとの連携を容易にすること、多様な機能を一つのサービスや UI に集約することの三つの役割があることが分かった。

本稿では多様な機能を一つのサービスや UI に集約する役割について、一つの画面に表示される情報量が膨大になり、ユーザーが目的のボタンや機能を探し出すことが難しくなる問題の解決を目的とする。アプローチとして、スマートフォン操作によりサービスの実行を可能にする「DigiTrack」の開発を行う。DigiTrack は A1: ユーザー情報取得、A2: UI の提供、A3: コマンド操作、A4: サービスの提供のアーキテクチャを持つ。

2. 準備

2.1 「超スマート社会」の実現と「Society5.0」

内閣府の第5期科学技術基本計画 [1] によれば、超スマート社会は、「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、言語といった様々な違いを乗り越え、生き活きと快適に暮らすことのできる社会」であると定義されている。

この社会では、生活の質の向上をもたらす人とロボット・AI との共生、ユーザーの多様なニーズにきめ細かに応えるカスタマイズされたサービスの提供、潜在的ニーズを先取りして人の活動を支援するサービスの提供、地域や年齢等によるサービス格差の解消、誰もがサービス提供者となれる環境の整備等が期待されている。

サイバー空間とフィジカル空間を融合させた取り組みにより、「超スマート社会」を未来社会の姿として共有し、その実現に向けた一連の取り組みを更に進化させつつ、「Society 5.0」として推進されている。

現在の社会変化、いわゆる第4次産業革命 [2] ともいえる時代の中で、人々の生活を支援するサービスが数多く生まれている。その中での課題は、多様なサービスにいかにか容易にアクセスできるかである。

2.2 サービス利用の現状

現在のサービスはスマートフォンや PC 上のアプリ上でサービスを利用することが一般的である。ここには解決すべき課題が二つある。

1 つめは、インターフェイスが多様化していることである。例えば、家電を操作するサービスと、音楽を再生するサービスを利用する場合、それぞれのアプリのインストール、起動、操作が必要になり、2 つのアプリを行き来する必要がある。これによってサービスのアクセスに対してユーザの負担が増加している。

二つめは、サービスの利用に対話操作が必要でない場合が存在するところである。家電を操作する場合を考えると、電源を入れる、消す、起動する、停止する、といった操作はユーザにデータの入力を要求するような対話的操作が必要がない。

2.3 ボタン駆動サービスへの期待

ここで「ボタン駆動スマートサービス」とはボタンを押すという単純な操作で呼び出し可能なサービスのことを指す。ボタン駆動スマートサービスはシンプルなアクションによって素早くサービスを利用でき、複雑な動作を簡略化する上で便利である。

ボタン駆動スマートサービスはユーザにとって非常に直感的で手軽であり、ユーザは特定のサービスを利用するために、複雑な手続きや入力を行う必要がなく、単にボタンを押すだけで目的を達することができる。これにより、新規ユーザにとっても利用の敷居が低くなり、手間をかけずにサービスを利用できる。

さらに、ボタン駆動スマートサービスは、特に IoT デバイス [3] との組み合わせにおいて効果的な制御手段となる。例えば、スマートホームの照明や温度制御、セキュリティシステムなど、様々なデバイスに対してワンクリックでアクセスすることで、ユーザは手軽に環境をカスタマイズできる。

以上の特徴からボタン駆動スマートサービスの需要は高まっているといえる。

2.4 先行研究: TapTrack

2.4.1 概要

TapTrack はボタン駆動スマートサービスの実装困難性に着目し、ボタン駆動スマートサービスの実装を容易化する開発実行基盤として提案された。開発者が新たにサービスを作成する場合、計算資源の確保、計算資源の確保、サーバーの構築、UI のフロントエンド開発、サーバーサイドの開発、データベースの設計、サービスのデプロイ、といった一連の作業により、ボタンが押されたことの検知機能やボタンが押された際に実行さ

れる処理といった機能を実装する必要がある。さらにこれにはコーディング知識が必須となる。ボタン駆動スマートサービスは、ボタンを動作させる方法として画面のクリックをする必要があり押すとサービスが実行されるという一見単純なサービス提供体制であるが、実際には開発者にとっては通常のサービスと同じ程度の実装コストを要する。

2.4.2 アプローチ

ボタン駆動スマートサービスの実装を容易化するプラットフォームの研究として、専門知識がなくてもボタン押下の検知機能とボタン押下時に実行される処理をノーコードで作成可能な GUI をサービスとして提供することでボタン駆動サービスを登録、実装、使用できるプラットフォームを提案したのが TapTrack である [4]。

2.4.3 アーキテクチャ

TapTrack のアーキテクチャは「TapTrack-A1: ボタン機能生成」「TapTrack-A2: ボタン機能割り当て」「TapTrack-A3: ボタンによる駆動」「TapTrack-A4: 振り返り」の四つの機能から構成されている。「TapTrack-A1」により、サービス開発者はボタンに割り当てたいボタン機能を作成しボタン機能情報データベース上に登録する。ユーザとは、TapTrack 上でサービスを利用する人を指す。ユーザは、「TapTrack-A2」により、必要とするボタン機能をボタン機能情報データベースから選択し、ユーザ情報データベース上に保存する。そして「TapTrack-A3」により、ボタンを押すとユーザ情報データベース上に保存されたボタン機能がボタン機能情報データベースから呼び出され実行される。ボタン機能が実行されたとき、「誰が」「いつ」「どのボタン機能を実行したのか」が保存され、「TapTrack-A4」により、ユーザおよび開発者はサービスの実行履歴を確認することができる。

2.4.4 TapTrack の特徴

ボタン駆動型スマートサービスである TapTrack の実装を通じて、ボタン駆動型スマートサービス基盤には大きく三つの役割があることが明らかになった。以下にその特徴を詳述する。

一つ目の役割は、非デジタル化された要素をソフトウェア化・仮想化することである。TapTrack を使用することで、専門的な開発知識がなくてもボタン駆動型スマートサービスを簡単に構築することができる。この仕組みにより、誰でも手軽に独自のサービスを作成し、利用することが可能となる。これまでデジタル化が進んでいなかった分野でも、新たな利便性を提供できる点が特筆される。例えば、操作が複雑な家電の手順を忘れた場合、事前に各手順を写真や動画で記録し、それを TapTrack を用いてボタン駆動型スマートサービスとして登録しておくことで、ボタンを押すだけで操作方法を簡単に思い出せる仕組みを提供する。これにより、従来の記憶やマニュアルに頼る必要が減少し、よりスムーズな利用体験が可能になる。

二つ目の役割は、他のサービスとの連携を容易にする点である。TapTrack は、HTTP を利用した API を通じてデータの送受信機能を提供しており、他のサービスやシステムとの連携が容易である。この特性により、従来は異なるプロトコルやシステム間の統合に課題があった場面でも、迅速かつ柔軟に接続

できる。たとえば、TapTrack を介してスマート家電やクラウドサービスとの連携を実現することで、ユーザーは複数のデバイスやプラットフォームをまたがる統合的な操作環境を構築できる。

三つ目の役割は、多様な機能を一つのサービスや UI に集約する点である。TapTrack は、多様な機能を一つのサービスや UI に統合する役割を果たす。従来は別々のツールやアプリケーションで提供されていた機能を一元管理することで、操作性を大幅に向上させている。たとえば、スマート家電の操作、情報の収集、音楽の再生といった異なる用途のサービスを利用する際、従来はそれぞれ個別のアプリケーションを起動し、個別の操作方法を習得する必要があった。TapTrack を利用することで、これらの機能が一つのプラットフォーム上に統合され、画面移動や操作の手間を省くことが可能となる。このような集約化により、ユーザー体験の効率性と直感性が向上し、特に多機能なデバイスの利用時に顕著な効果を発揮する。

2.5 着目する問題

ここで、TapTrack の三つめの役割である「多様な機能を一つのサービスや UI に集約する点」に着目する。サービスが集約されることで利便性が向上する一方、一つの画面に表示される情報量が膨大になり、ユーザーが目的のボタンや機能を探し出すことが難しくなる問題が生じる。この問題は視認性や操作性が低下させ、結果としてユーザーエクスペリエンスを損なう可能性が大いにある。この問題に対処するためには、情報の適切な分類、優先順位付け、および直感的なナビゲーション設計が求められる。

2.6 仮説

本研究では、ボタン駆動スマートサービスの集約と効率化を目指し、以下の二つの仮説を立てた。それぞれの仮説について詳細を述べる。

2.6.1 H1: 番号との紐づけによるボタンの押下の有効性

直接のボタン押下ではなく、番号との紐づけを活用することで、UI 上の混雑を緩和し、効率的な操作を可能にする。具体的には、番号でボタンを識別することで、視覚的負荷を軽減し、情報量の多い画面でも迅速に目的の操作が可能になる。さらに、この方法は音声入力やリモート操作との相性が良く、操作の柔軟性が高まると考えられる。

2.6.2 H2: サービス実行端末とトリガ端末の分離の有効性

サービスを実行する端末と、その実行をトリガする端末を分離することで、複雑な操作や多機能統合の効率化が期待される。例えば、スマートフォンをトリガ端末として使用し、実際の処理をクラウドや専用デバイスで実行する仕組みを採用することで、端末ごとの役割分担が明確化し、操作が直感的になる。この分離により、サービス全体のスケーラビリティやユーザー体験の向上も見込まれる。

これらの仮説に基づき、ユーザーインターフェースやサービスの設計を行う

3. 提案手法

3.1 目的

本研究の目的は、ボタン駆動スマートサービスの実装を容易化するためのプラットフォームを研究・開発することである。特に、ボタンに番号を割り当てる操作方式と、手元デバイスを用いたリモート動作を組み合わせることで、以下を実現することを目指す。

- ボタン駆動スマートサービスの効率的な集約
- ユーザーインターフェースの簡素化と直感的な操作性の向上
- 多機能サービスの一元管理による利便性の向上

これにより、従来のボタン駆動型サービスが抱える操作性の課題を解決し、幅広いユーザーにとって利便性の高いプラットフォームを提供することを目指す。

3.2 キーアイデア

番号により操作するスマホ用 web アプリ [5] を作成し、TapTrack と連携を行う。

3.3 アプローチ

スマートフォン操作によりサービスの実行を可能にする「DigiTrack」の開発を行う。DigiTrack は、スマートフォン上で番号を押すことで、TapTrack 上で登録されたボタン機能呼び出し、実行することができる基盤であり、TapTrack から登録済みのボタンの情報を取得し、ユーザーはボタンに割り当てられている番号を入力することで、そのボタン機能を TapTrack が動作している端末上で実行することができる。

3.4 全体アーキテクチャ

図 1 に DigiTrack の全体アーキテクチャを示す。DigiTrack は、以下の四つの要素により構成されている。

- A1: ユーザー情報取得
- A2: UI の提供
- A3: コマンド送信
- A4: サービスの提供

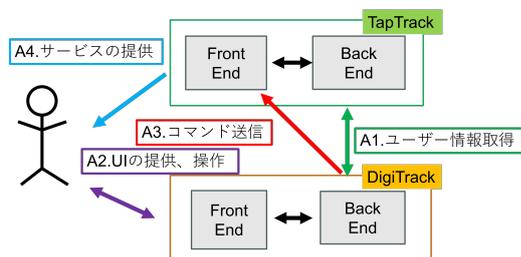


図 1 DigiTrack の全体アーキテクチャ

ユーザーはまず、DigiTrack にユーザー ID を登録する。A1 では、DigiTrack がその ID を用いて TapTrack にアクセスし、ユーザー名と、ボタンに割り当てられた機能の対応表を受け取る。A2 では、受け取った対応表をもとに操作用の UI を画面に提供する。この UI により、ユーザーは簡単に目的の操作を選択できる。A3 では、ユーザーの操作に応じて DigiTrack が TapTrack に対してコマンドを送信し、対応する動作を実行さ

せる。最後に、A4 で TapTrack が指定されたサービスを実行し、ユーザーにその結果を提供する。

3.4.1 A1: ユーザー情報の取得

ボタン駆動スマートサービス実装基盤 TapTrack には、開発者が開発した機能の情報、ユーザーが登録しているボタンとそこに割り当たっている機能に関する情報を持つデータベースがある。DigiTrack ではこれを使用する。

3.4.2 A2: UI の提供

DigiTrack は A1 において取得したボタン番号と機能の割り当てをユーザーが確認できるように表として表示する画面と実際に番号を入力する画面の二つを提供する。誤入力による予期せぬ実行を防ぐため、番号を入力する画面で入力を行うと実行される機能名が表示されるようにしておく。

3.4.3 A3: コマンド送信

番号を入力し実行ボタンを押すと対応するボタンを押すように TapTrack のフロントエンドにコマンドを送信する。TapTrack のフロントエンドはユーザーのブラウザ上で動作されるため、双方向かつリアルタイムな通信を可能にする必要がある。これには WebSocket [6] を使用する。DigiTrack のバックエンドに WebSocket 用のエンドポイントを立て、TapTrack のフロントエンドに WebSocket で接続するスクリプトを加えることで実現する。

3.4.4 A4: ボタン駆動サービスの提供

実際のサービス提供は TapTrack において実行される。

4. 実装

4.1 利用した技術

本稿では、前述のアーキテクチャを基に DigiTrack を実装した。実装は以下の技術スタックを用いて構築されており、主にフロントエンドとバックエンドの二つの Docker コンテナ [7] で構成される。

4.1.1 フロントエンド

フロントエンドは、静的リソースの配信を担当する。以下の技術を用いて実現した。

- **使用技術:** httpd (Apache HTTP Server) [8] の Docker [9] イメージを使用
- **構成内容:** HTML, CSS, JavaScript を静的リソースとして配置
- **役割:** ユーザーインターフェースを提供し、ユーザーが DigiTrack を操作するためのエントリーポイントを形成

4.1.2 バックエンド

バックエンドは、ビジネスロジックやデータ管理を担当する。以下の技術を用いて実現した。

- **使用技術:** java 環境の [10]OpenJDK-17 の Docker イメージを使用
- **開発フレームワーク:** Spring Boot [11]
- **役割:** ユーザーリクエストを受け付け、データベースや外部サービス (TapTrack) と連携して必要な処理を実行

4.1.3 デプロイ

上記のフロントエンドとバックエンドを Docker コンテナ

として構築し、公開用サーバーにデプロイした。これにより、DigiTrack のサービスを外部から利用可能な状態にした。

4.2 ケーススタディ

本章ではケーススタディとして、実際の画面を用いて DigiTrack システムを使用する様子を説明する。まずは TapTrack の機能「TapTrack-A1」および、「TapTrack-A2」を用いて、ボタン機能の作成と割り当てを完了させる。これにより、図 2 に示すように、ボタンが TapTrack 上に表示される。

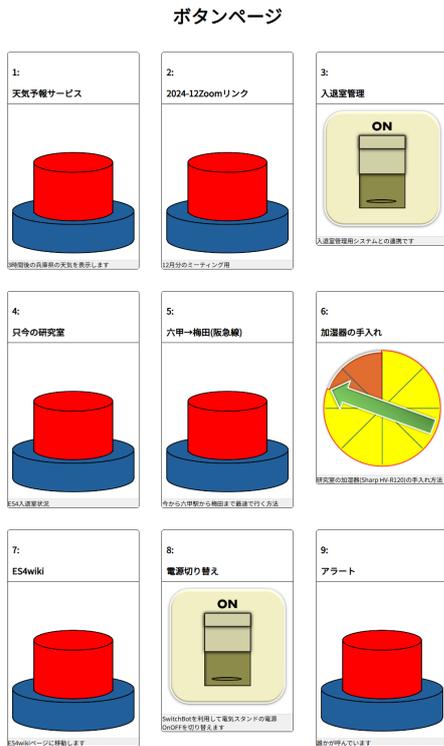


図 2 TapTrack のボタン表示画面

この画面を PC 等固定されたデバイスで表示したままの状態にしておく。

次にスマートフォンのような他のデバイス DigiTrack の画面を開くと、図 3 に示すように、ボタン番号入力画面が表示される。



図 3 DigiTrack の操作画面

この画面ではボタン番号を入力し、実行ボタン「Go」を押すことで、TapTrack 上で実行するようにコマンドが送信される。

「対応表」ボタンをクリックすると、図 4 に示すような対応表が表示される。この表は TapTrack から情報を取得し表示される。

番号	機能名	説明
1	天気予報サービス	3時間後の兵庫県の天気を表示します
2	2024-12Zoomリンク	12月分のミーティング用
5	六甲→梅田(阪急線)	今から六甲駅から梅田まで最速で行く方法
6	加湿器の手入れ	研究室の加湿器(Sharp HV-R120)の手入れ方法
4	只今の研究室	ES4入室状況
3	入退室管理	入退室管理用システムとの連携です
8	電源切り替え	SwitchBotを利用して電気スタンドの電源OnOFFを切り替えます
7	ES4wiki	ES4wikiページに移動します
9	アラート	誰かが呼んでいます

図 4 ボタン番号対応表

この対応表により、ユーザーはどの番号を入力するべきかを確認できる。今回は例として、登録されている天気予報サービスを利用することとする。図 4 の対応表によるとそれは「1」である事が分かる。

次に、図 3 の画面に戻り、「1」と入力すると図 5 のように実行されるボタン機能名がディスプレイに表示される。

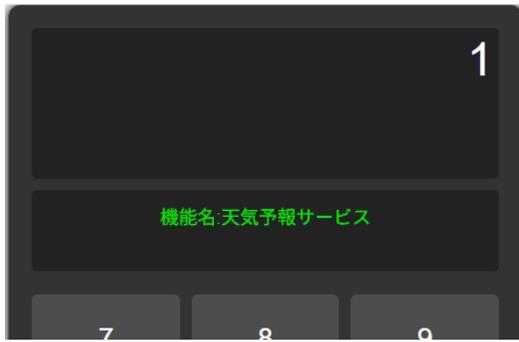


図5 ボタン番号入力状態

表示されている機能名と目的の機能が一致している事が確認出来たら、「Go」ボタンを押すことでその機能が TapTrak 上で実行される。図6はこの時の様子を表す画面である。表示されている画像は tenki.jp [12] の画像を使用している。

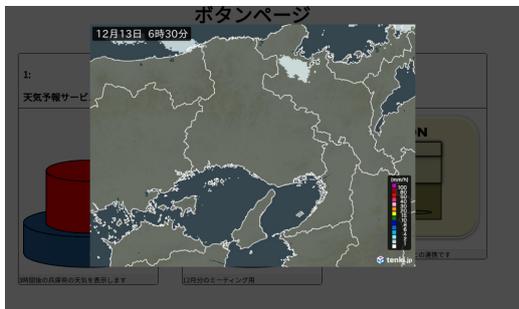


図6 TapTrack 上での実行

4.3 考察

以上により、DigiTrack を用いたボタン駆動スマートサービスの番号実行及びリモート実行の動作を確認した。H1 への回答として、DigiTrack の導入により、登録したボタン駆動スマートサービスが増えたとしても、画面表示を最小限とすることで情報が削減され迅速な操作が可能となった。H2 の回答として、表示デバイス (例: パソコンや固定されたタブレット端末) から離れていても、手元のスマートフォンを用いてリモートで操作を実行できるようになった。これにより、物理的な制約を受けずにサービスを利用でき、ユーザーの利便性が向上した。

5. 研究の限界

本稿は予備的なものであり、ボタン駆動スマートサービスの実行における操作性を課題を解決するためのアプローチとして番号操作を行うスマートフォン用アプリケーションの提案および開発に焦点を当てています。

制約として今回の仮説について十分議論するための定量的な評価を行うことができなかった。今後データ収集と詳細な評価が必要である。特に SQuaRE の利用時の品質特性として上げられる「有用性」「満足性」について評価する必要がある。これらについては今後の課題とする。

6. まとめ

本研究ではボタン駆動スマートサービスの集約と効率化を目

的とし、番号との紐づけによるボタンの押下の有効性とサービス実行端末とトリガ端末の分離の有効性という二つの仮説のもと、ボタン駆動スマートサービスの実行開発基盤との連携を行い番号によるリモート実行を可能にするサービス DigiTrack の提案を行った。

DigiTrack は TapTrack からの情報の取得、UI の提供、コマンドの送信、サービスの提供の機能を持っている。

本研究ではケーススタディとして実際に登録されているボタン駆動スマートサービスを DigiTrack で実行する様子を説明した。今後の課題としてはこのサービスの有用性について評価を行うこととする。

謝辞 本研究の一部は JSPS 科研費 JP20H05706, JP22H03699, JP22K19653, JP23H03401, JP23H03694, JP23K17006, 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。

This research was partially supported by JSPS KAKENHI Grant Numbers JP20H05706, JP22H03699, JP22K19653, JP23H03401, JP23H03694, JP23K17006.

文献

- [1] 内閣府, “第5期科学技術基本計画,” <https://www8.cao.go.jp/cstp/kihonkeikaku/5honbun.pdf>, 2016.
- [2] 山本昭二, “第4次産業革命とサービス科学の拡張,” 横幹, vol.11, no.1, pp.6–10, 2017.
- [3] 岩野和生, 高島洋典, “サイバーフィジカルシステムと IoT (モノのインターネット) 実世界と情報を結びつける,” 情報管理, vol.57, no.11, pp.826–834, 2015.
- [4] 村手亮太, 中橋友郎, 陳思楠, 佐伯幸郎, 中村匡秀, 安田清, 露崎雄太, “ボタン駆動スマートサービスの実装を容易化するプラットフォームの研究,” 電子情報通信学会技術研究報告, vol.123, no.429, LOIS2023-61, pp.078–083–078–083, March 2024.
- [5] 命 栢山, 実 寺田, “構造や動作のつながりに着目した web アプリケーションの可視化,” 第65回プログラミング・シンポジウム予稿集, 第2024巻, pp.25–32, jan 2024.
- [6] S. Loreto and S.P. Romano, Real-time communication with WebRTC: peer-to-peer in the browser, " O'Reilly Media, Inc.", 2014.
- [7] 廣木智栄, 小澤伸也, 道幸雄真, 清水尚希, “コンテナ仮想化を用いたサーバ環境構築技術の修得,” 技術部活動報告集, vol.29, pp.8–11, 2024.
- [8] B. Laurie and P. Laurie, Apache: The definitive guide, " O'Reilly Media, Inc.", 2003.
- [9] I. Miell and A. Sayers, Docker in practice, Simon and Schuster, 2019.
- [10] J. Bloch, Effective java, Addison-Wesley Professional, 2017.
- [11] C. Walls, Spring in action, Simon and Schuster, 2022.
- [12] “tenki.jp,” <https://tenki.jp/>, 2024. (Accessed on 02/06/2024).