

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

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^{†††}

[†] 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

^{†††} おゆみの中央病院 〒266-0033 千葉市緑区おゆみ野南 6 丁目 49 番地 9

E-mail: †rmurate@es4.eedept.kobe-u.ac.jp, ††tomorrow@ws.cs.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 近年、超スマート社会を未来社会の姿として現実に向けた動きが Society5.0 として推進されている。この社会の動きの中で、サービスは実に多様化しており、サービスの利用にあたってユーザの負担が増加している。そこで、本稿では、ボタン駆動スマートサービスに着目し、ボタン駆動スマートサービスの実装を容易化するプラットフォーム「TapTrack」の提案を行う。TapTrack では、ボタンを押したときに実行されるサービスをノーコードで作成可能であり、ユーザは自分に必要なサービスをボタンを選んで割り当てることができる。ボタンが押された際に実行可能な処理は、メディアファイルの再生、外部サービスに対するコマンド送信ボタンである。ボタンの使用履歴は、ライフログとして後から振り返ることができる。ケーススタディでは、TapTrack を用いて天気予報確認サービスと生活記録サービスを作成し、実際に動作を確認した。

キーワード ボタン駆動スマートサービス, ライフログ, ノーコード開発

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

Ryouta MURATE[†], Tomoro NAKAHASHI[†], Sinan CHEN[†], Sachio SAIKI^{††}, Masahide
NAKAMURA[†], Kiyoshi YASUDA[†], and Yuta TSUYUZAKI^{†††}

[†] 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

^{†††} Oyumino Central Hospital, 6-49-9 Oyumino Minami, Midori-ku, Chiba-shi, Chiba 266-0033, JAPAN

E-mail: †rmurate@es4.eedept.kobe-u.ac.jp, ††tomorrow@ws.cs.kobe-u.ac.jp, †††chensinan@gold.kobe-u.ac.jp,
††††saiki.sachio@kochi-tech.ac.jp, †††††masa-n@cmds.kobe-u.ac.jp

Abstract In recent years, in the super smart society that has progressed as Society 5.0, the use of a wide variety of services has increased, and the burden on users has also increased. Therefore, in this paper, we propose “TapTrack”, a platform that easily realizes button-driven smart services. TapTrack allows users to create a service that runs when a button is pressed with no code, and allows users to assign their desired services to the button. In the case study, we created two services using TapTrack and verified its operation.

Key words Button-driven smart service, Life log, No-code development

1. はじめに

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くに対応でき、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」とされ、その実現に向けた一連の取組が「Society5.0」として推進されている。[2]

このような現代社会では多様なサービスが生まれ、その利用においてスマートフォンやPCのアプリ上でサービスを使用することが一般的である。しかしこれにはインターフェイスの多様化という問題が生じ、ユーザの負担が増加している。

ユーザビリティの向上のために「ボタン駆動スマートサービス」に着目する。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。関連技術として、Amazon Dash Button [3] や AWS IoT Button [4] があるが、その実装には制約があり難易度が高いことが問題点として浮かび上がる。

そこで本研究では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」の提案を行う。よって満たされるべき要件として以下を定義する。

- R1: 日常繰り返し行う行動を「ボタン1つ」で実行したい
- R2: ユーザが手軽・簡単に自分好みのサービスを作りたい

要件を満たすため、TapTrackの全体アーキテクチャは以下の4要素により構成される。

- A1: ボタン機能生成
- A2: ボタン機能割り当て
- A3: ボタンによる駆動
- A4: 振り返り

ユーザは、ボタン機能を作成し、ボタンに割り当て、ボタンを押すことでサービスを実行され、また実行履歴を振り返ることができる。

ケーススタディでは、TapTrackを用いたボタン駆動スマートサービスの実装および実行の具体例として「天気予報確認サービス」、「生活記録サービス」を作成する。これにより、提案手法が様々なボタン駆動スマートサービスの実装のサポートに効果的であることが期待できる。

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くに対応でき、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・A

Iとの共生、ユーザの多様なニーズにきめ細かく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 [5] 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2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。

2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 [6] 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 関連技術

2.4.1 Amazon Dash Button

Amazon Dash Button [3] は Amazon が提供するボタン駆動スマートサービスである。ボタンを押すと商品を発注するこ

とができる。消耗品の発注に特化して利用されている。好きなサービスを好きな時に実行するものではない。

2.4.2 AWS IoT Button

AWS IoT Button [4] は Amazon が提供するボタン駆動サービスである。ボタンを押すと Lambda 関数を実行することができる。多くのことができるが、玄人向けである。

2.5 着目する問題

本研究ではボタン駆動スマートサービスの実装が困難であるという問題に注目する。ボタン駆動スマートサービスは、ボタンを押すとサービスが実行されるという単純な仕組みであるが、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がかかる。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、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。

3. 提案手法

3.1 目的

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する。

3.2 要件

満たされるべき要件を以下に示す。

- R1: 日常繰り返し行う行動を「ボタン1つ」で実行したい
- R2: ユーザが手軽・簡単に自分好みのサービスを作りたい

3.3 アプローチ

この要求を満たすため、ボタン駆動スマートサービス実装基盤「TapTrack」の提案をする。TapTrack はノーコードで自分好みのボタン駆動サービスを作成するであり、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をキーマインドとする。

3.4 全体アーキテクチャ

図1に TapTrack を導入した全体アーキテクチャを示す。TapTrack は、以下の要素により構成される。

- A1: ボタン機能生成
- A2: ボタン機能割り当て
- A3: ボタンによる駆動
- A4: 振り返り

サービス開発者は、TapTrack 上にサービスを実装する人を指す。A1: ボタン機能生成により、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、A2: ボタン機能割り当てにより、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 A3: ボタンによる駆動により、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、A4: 振り返りにより、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。

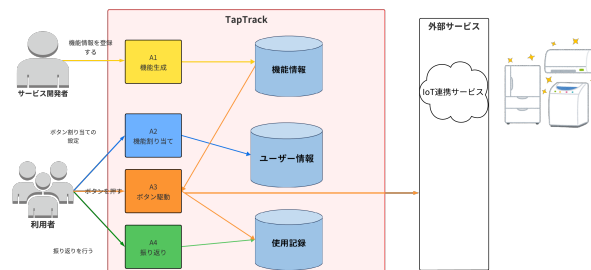


図1 全体アーキテクチャ

3.4.1 A1: ボタン機能生成

ここでは、ユーザがボタンに割り当てたい日常生活を支援するボタン機能を作成する。以下に示すような設定をノーコードで行うことができる。

- S1: ボタン機能のふるまいの定義
- S2: ボタンを押す動作に対するリアクション
- S3: HTTP リクエストの送信
- S4: ボタン機能の公開範囲

S1: ボタン機能のふるまいの定義に関して、単にボタン1つで、操作可能と言っても、求められるふるまいは1つではない。具体的には、ボタンを押すと天気予報を表示するようなボタン機能の場合、「天気の詳細を取得する」というボタンを押すごとに同じふるまいをすればよいが、電気をつけるボタン機能の場合は、「電源を入れる」と「電源を切る」という2つのふるまいが必要になる。このようなふるまいを以下のような3タイプに分けて設定する。

- トリガー: 押すと1つのサービスを実行する
- スイッチ: 1度押すとサービス0を実行する、2度押すとサービス1を実行する

ループ n : i 回目に押すとサービス i を実行し、 n 回目に押すとサービス0を実行する

S2: ボタンを押す動作に対するリアクションに関して、ボタンを押したとしてもユーザには何もフィードバックがないと、押されたのかどうかかわらず、複数回押してしまう可能性がある。このような設計は避けるべきである。そこで、リアクションとして再生するメディアファイルを設定できるようにする。提案手法 TapTrack では、以下の4つから設定方法を選択できる。

- S2-1: メディアファイルのアップロード
- S2-2: URL の参照

S2-3: テキストの読み上げ

S2-4: なし

S3: HTTP リクエストの送信に関して、この設定は外部サービスとの連携を行うための設定である。ボタンが押されたときに、外部サービスに対して HTTP リクエストを送信することができる。

S4: ボタン機能の公開範囲に関して、この設定はボタン機能を公開する範囲を設定する。ユーザが作ったボタン機能は共有可能にすることができる。この設定があることで誰でも開発者となることができる。設定は以下の3つから選択できる。

S4-1: 公開

S4-2: 非公開

S4-3: 限定公開

ボタン機能にはそれぞれ ID が割り当てられており限定公開の場合は、その ID を知っている人のみが利用できる。

3.4.2 A2: ボタン機能割り当て

ボタン駆動サービスの各ボタンに A1 で作成したボタン機能を割り当てる。ユーザが好みのサービスをボタンから呼び出し可能になる。自分が作成したボタン機能と他人が公開範囲を「公開」にしているボタン機能をボタンに割り当てることができる。限定公開のボタン機能は、ID を使って検索することでボタンに割り当てることができる。

3.4.3 A3: ボタンによる駆動

ボタンを押すと A2 によって割り当てられたボタン機能が実行される。ボタンには「状態」があり、A1 で設定したふるまいが「ボタン」のとき状態はボタンを押しても変化しない。(例: 1→1→1→...)

- ユーザがボタンを押すとサービスが割り当てられたボタン機能を実行する
- 画面はユーザのパーソナルなボタンになる

3.4.4 A4: 振り返り

ユーザがどのボタンをいつ、何回押したかを確認できる。次の3種類のグラフ形式で表示する。棒グラフはボタンを何時に何回押したかを確認できる円グラフは指定した期間内において、ボタンの状態が続いた時間とその割合を確認できる。タイムラインはいつ、どの状態のボタンを押したのかがタイムライン上に表示される。

4. 実装

4.1 利用した技術

本稿では、提案するボタン駆動スマートサービス「TapTrack」を実装した。バックエンドは Java 言語と Spring Boot を用いて実装した。データベースサーバには MySQL を使用し、Web サーバには Apache Tomcat を利用した。フロントエンドは HTML, CSS, JavaScript, google charts を用いて実装した。

4.2 画面

実装した TapTrack は、ボタン駆動型スマートサービスの実行および実装の基盤として GUI を提供する。

図2、図3、図4はボタン機能作成画面である。図2では、ボタン機能のふるまいの定義およびリアクションとして再生するメディアファイルの設定方法の選択を行う。図3では、ボタン機能名、リアクションとして再生するメディアファイル、ボタン機能の説明を入力する。図4では、ボタン機能が送信する HTTP リクエストの設定を行う。

どのような機能を作りたいですか？

作りたい機能の種類を選択してください

--選択--

ボタンを押したときの反応の種類を選択してください

--選択--

作成画面へ

図2 ボタン機能のふるまいの定義

機能追加

機能名

機能名	ファイル1	ファイル2
サンプル1	ファイル1個めのファイル [ファイルを選択] 140111 (720p).mp4	ファイル2個めのファイル [ファイルを選択] 180386 (720p).mp4

作成

図3 ボタン機能名とリアクションとして使うファイル

名前

URL

メソッド

選択してください

ヘッダー	値	操作
キー		

ボディ

実行する順番

登録

図4 HTTP リクエストの送信設定

図5はボタン機能選択画面である。この画面には自分が作成したボタン機能と他人が公開範囲を「公開」にしているボタン機能が表示されている。「ボタン機能 ID から探す」欄にボタン機能ごとに割り振られている ID を入力することで、他のユーザが「限定公開」として追加したボタン機能もボタンに割り当てることができるようになる。

機能管理



図 5 ボタン機能選択

図 6 は実際にサービスを駆動させるボタン画面である。図 6 のように 0 ~ 10 のボタンが表示されており、ボタンを押すとサービスを駆動させることができる。



図 6 ボタン

5. ケーススタディ

5.1 アプリケーション概要

ケーススタディでは、ボタン駆動スマートサービスの作成を目的として、4. で実装した TapTrack を用いて、ボタン駆動スマートサービスを作成する。

5.2 天気予報確認サービス

A1: ボタン機能作成

まず、ボタン機能作成画面に遷移する。このボタン機能は「天気の情報取得する」という同じ動作を毎回行えばよいので、ふるまいは「トリガー」とする。リアクションに設定するメディアファイルを天気予報の情報としたときに、リアクションとして再生するメディアファイルの設定方法は外部サービスを利用するため、「URL の参照」を選択する。次に、ボタン機能名、リアクションとして再生するメディアファイル、ボタン機能の説明を入力する。ボタン機能名は「兵庫県の天気予報」とした。今回、天気予報の情報は tenki.jp [7] が提供する 3 時間後の雨雲レーダーの画像を利用する。

A2: ボタン機能割り当て

次に、ボタンにボタン機能を割り当てる。今回はボタン 2 にボタン機能を割り当てることにする。ボタンボタン機能設定画面からボタン 2 を選択し、表示されるボタン機能のリストの中から先ほど作成した「兵庫県の天気予報」を選ぶ。

A3: ボタンによる駆動

ボタン 2 を押すと、図 7 のように天気予報の画像が表示さ

れる。



図 7 表示される画像

5.3 生活記録サービス

ボタンを押すことで生活記録をつけることができるボタン機能を作成する。勉強を終了するとき、電気スタンドの電源を切るようにする。

A1: ボタン機能作成

まず、ボタン機能作成画面に遷移する。記録の項目を「睡眠」、「勉強」、「家事」、「遊び」、「その他」の 6 つに設定するため、ふるまいは「ループ 6」とする。リアクションとして再生するメディアファイルの設定方法は「アップロード」とする。

次に、ボタン機能名、リアクションとして再生するメディアファイル、ボタン機能の説明を入力する。ボタン機能名は「生活記録」とした。メディアファイルは簡単な効果音の mp3 ファイルをアップロードする。説明欄にはどの状態をどのように記録するかを分かるように記述する。(1. 睡眠, 2. 勉強, 3. 家事, 4. 移動, 5. 遊び, 6. その他)

最後に、Switch Bot [8] を用いて、勉強を終えるときに電気スタンドの電源を切るようにする。図 4 の画面で、「HTTP リクエストの送信」の設定を行う。Switch Bot アプリからデバイス ID とトークンを取得し、以下のように設定する。

URL: <https://api.switch-bot.com/v1.0/devices/デバイスID/commands>

ボディ: { "commandType": "command", "command": "turnOff", "parameter": "default" }

ヘッダー:

Authorization: トークン

Content-Type: application/json; charset=utf8

A2: ボタン機能割り当て

次に、ボタンにボタン機能を割り当てる。今回はボタン 3 にボタン機能を割り当てることにする。

A3: ボタンによる駆動

ボタン 3 を押すと、状態の数字が変化する。ボタンを押した

びに状態が変化し、6 のとき次に押すと 1 に戻る。状態の数字が記録したい行動に合うようにボタンを押すことで、生活記録をつけることができる。例えば、睡眠を記録したいときは、ボタン 3 の状態が 1 になるまでボタンを押すようにする。

A4: 振り返り

つけた生活記録は後から図 8 のように確認することができる。なお、状態とは以下のような対応を持つ。(1. 睡眠, 2. 勉強, 3. 家事, 4. 移動, 5. 遊び, 6. その他)



図 8 生活記録

5.4 考察

以上により、提案手法 TapTrack を用いたボタン駆動スマートサービスの作成を行い、天気予報確認サービス、生活記録サービスの動作を確認した。2.5 節で述べた P1 について、TapTrack を用いることでボタン機能をノーコードで作成できるため、ボタン駆動スマートサービスの実装の難しさを解決できることが期待できる。2.5 節で述べた P2 についても、TapTrack を用いることで、様々なサービスを 1 つのプラットフォーム上で実装、利用することが可能となっており、インターフェイスの多様化を解決できることが期待できる。

しかしながら、TapTrack ではボタン入力が web ブラウザ上の操作に限定されているという限界があることが分かった。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、十分であるとは言えない。例えば、スマートフォンからのリモート操作、人感センサによる操作などというような様々な方法ができるようになると、より様々なスマートサービスに対応できるようになると考えられるがこれについては今後の課題とする。

また、この提案手法の有益性を明確に評価するためには実際の利用に基づく評価が必要であるが、これについても今後の課題とする。

6. まとめ

本稿では、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することを目的として、TapTrack を提案した。

TapTrack は、ボタン駆動スマートサービスの実行基盤とし

て GUI を提供し、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である。ユーザは、このプラットフォーム上にサービスをノーコードで実装することができる。また、ユーザは、自身のニーズに合わせて必要なサービスを自身で選ぶボタンに割り当て、ボタンを押すことでサービスが実行されること、そして、サービスの実行履歴を確認することができる。

本研究ではケーススタディとして、天気予報確認サービス、生活記録サービスの作成を行い、その動作を確認することで、提案手法 TapTrack の有用性を示した。今後の課題としては、サービス駆動時における入力の多様化および実際の利用に基づく評価が挙げられる。

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] 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] 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,” 計造生, vol.55, no.4, pp.284–287, 2016.
- [3] Amazon, “Amazon dash 利用規約,” <https://www.amazon.co.jp/gp/help/customer/display.html?nodeId=201730770>, 2023. (Accessed on 02/06/2024).
- [4] AWS, “Aws iot 1-click (aws lambda トリガーのワンクリック作成) | aws,” <https://aws.amazon.com/jp/iot-1-click/>, 2023. (Accessed on 02/06/2024).
- [5] 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
- [6] 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
- [7] “tenki.jp,” <https://tenki.jp/>, 2024. (Accessed on 02/06/2024).
- [8] SwitchBot, “Switchbot (スイッチボット公式サイト),” <https://www.switchbot.jp/>, 2024. (Accessed on 02/06/2024).