

metaBox: 使い方を定義可能な exUI 設計の IoT Box の試作

中里健也^{†1} 金子翔麻^{†1} 渡邊恵太^{†2}

概要: シェアリングエコノミーの拡大により個人間でのモノのやり取りが増えている。また、宅配ボックスや宅配ロッカー、無人コンビニなどモノのやり取りを省力化するサービスが登場したが、サービスによって専用のハードウェアが必要であり、汎用性がない。本研究では、exUI 設計に基づいた、スマートフォンでユーザインタフェースを提供する IoT Box, metaBox を提案する。metaBox は箱のユーザインタフェースをソフトウェアで提供することで箱の用途をその都度定義でき、様々なモノのやり取りのシチュエーションやサービスに使える。

キーワード: IoT, exUI, シェアリングエコノミー

1. はじめに

シェアリングエコノミーとは、モノを個人で単独所有せずに必要なときに所有している人から借りる、もしくは売買するという経済体系である。必要なときは借り、所有物は共有すれば良いという考えを持つ人が増えており、インターネットやスマートフォンの普及によって、両者を引き合わせるプラットフォーム・サービスが登場し注目を集めている。特にメルカリ[1]やラクマ[2]などを始めとする個人間でモノの売買ができるサービスは大きな市場を形成している。

また、それ以前から普及し始めていた Amazon[3]や楽天市場[4]に代表される EC サイト上での購買は当たり前に行われるようになってきた。こうしたオンライン上での購買活動は、個人または EC サイト事業者から個人宅への小口で多頻度な宅配を急増させ、再配達問題とともに社会問題化している。いくつかの EC サイト事業者や配送業者は宅配ロッカーを設置し、対策を講じているが、特定の事業者・サービスでしか利用できないものもあり、ロッカーという共通の形状であるにもかかわらず互換性や汎用性がない。

ところでロッカーは昔から自動（無人）販売に使われている。とりわけ、現在では電子決済の普及や省力化のために店頭販売においても無人化を進める動きもある。入店後、商品をカバンにしまい店を出ると、カメラによるセンシングで持ち出した商品が自動的に決済される Amazon Go[5]や RFID タグのついた商品がショーケースに並べられていて、自由に取り出したものをクレジットカードで決済する無人コンビニ「600」[6]などがある。これらは、大きさは異なるが形状は共通して箱状である。

ロッカーも箱の集合だと考えることができる。つまり、箱のインタフェースさえ変更できれば、1 つの箱で宅配ロッカーや無人販売のサービスが可能である。また、個人間取引も宅配せずに箱を使って受け渡すなど、これまでになかった新たなサービスも可能になる。後述する exUI[7]の設計を取り入れることにより、箱のユーザインタフェースを自由に変更することができ、モノのやり取りをする汎用性

のある箱が作れる。exUI とはプロダクトのユーザインタフェースをスマートフォンに外在化し、機能を Web アプリケーションとして提供することで、汎用性のあるハードウェアを専用機のように使う、プロダクトの設計方法である。

本研究では、モノのやり取りを行うプラットフォームとなる箱を exUI の設計に基づいて製作し、アプリケーション次第で多様な用途に使える metaBox を提案する。

2. metaBox

metaBox とは、exUI 設計に基づいた物理的なユーザインタフェースのない、使い方を都度できる IoT Box である。本章では、metaBox に使っている exUI と metaBox の特徴を記す。

2.1 exUI

exUI とは、プロダクトについているユーザインタフェースをハードウェア本体から原則無くし、スマートフォンから提供する手法である（図 1）。ハードウェア自体の機能を操作するユーザインタフェースがハードウェア側に固定されていると特定の用途にしか使えないが、ユーザインタフェースを変更可能なスマートフォンで提供することで、用途次第で最適のユーザインタフェースを提供できる。また、プロダクトのユーザインタフェースがなくなることで外形の設計が比較的自由になることや製造コストの削減ができる。そして、多言語対応やレコメンドなどのパーソナライズやターゲティング、A/B テストやアジャイル開発、従量課金やサブスクリプションモデルなど、これまでの Web サービスの柔軟な開発や運用の構造を取り入れることができるメリットがある。



図 1 exUI に基づいたプロダクト

2.2 metaBox とは

metaBox とはインターネットから電子錠の制御が可能な exUI 設計に基づいた箱で、様々なモノのやり取りに使うことができる汎用的な箱システムである (図 2)。用途ごとに専用のユーザインタフェースを用意することで、その用途専用のハードウェアのように使うことができる。将来的にはデベロッパがアプリを開発し、ユーザが複数のアプリから目的のアプリを選び、metaBox に適用して使うことを想定している。また、箱の内部にセンサやアクチュエータを搭載することでより多用途に使えるが、今回は最小構成の電子錠のみで試作した。



図 2 metaBox と metaBox Definer を開いたスマートフォン

■ metaBox Definer

今回は metaBox の利用用途を探索するために metaBox の使い方をユーザ自身が定義できるユーザインタフェース (以下、metaBox Definer) を開発した。

metaBox Definer は metaBox を様々な用途に使うために用意した API をラップしたアプリケーションである。metaBox のタイトル、説明、どのユーザがどういう条件で解錠できるかを定める解錠条件、一連の利用の終了のタイミングを決める終了条件を定義する。ユーザは利用開始時に metaBox Definer を使って metaBox の使い方を定義して metaBox を利用する。

今回想定した使い方は、metaBox を使ったモノの受け渡しや数人のユーザ間のみで開けられるように設定し、そのユーザ間でのモノの共有、1 日に開ける回数を制限することによるお土産などのディストリビューションなどである。また、研究室メンバーは日常的に送金アプリを利用しているため、metaBox を使ったモノの販売も想定した。

2.3 利用の流れ

metaBox の利用場面は次の 2 つである。1 つは metaBox の使い方を定義して利用を開始することで、1 つは使い方を定義された metaBox を利用することである。また、ユーザが metaBox の使い方を定義して利用を開始し、定義され

た終了条件に基づいて利用が終了するまでの一連の進行をプロセスと呼ぶ。

■ metaBox Definer による metaBox の定義

ユーザがプロセスの動いていない、使い方を定義されていない metaBox にスマートフォンをかざして、NFC タグか QR コードを読むと、metaBox Definer が開く。ユーザは metaBox Definer で metaBox のタイトル、説明、解錠条件、終了条件を定義し、必要であれば metaBox を開きモノを設置して利用を開始する (図 3 左)。

■ metaBox Definer で定義後の利用

ユーザがプロセスの動いている metaBox にスマートフォンをかざして NFC タグか QR コードを読むと、同様にユーザインタフェースが開く。このユーザインタフェースは metaBox Definer で定義したタイトルと説明を表示し、そのユーザに解錠権限がある場合のみユーザは metaBox を開けられる。なお、定義された終了条件が満たすとプロセスが終了し、再び使い方を定義し利用可能な状態に移る (図 3 右)。

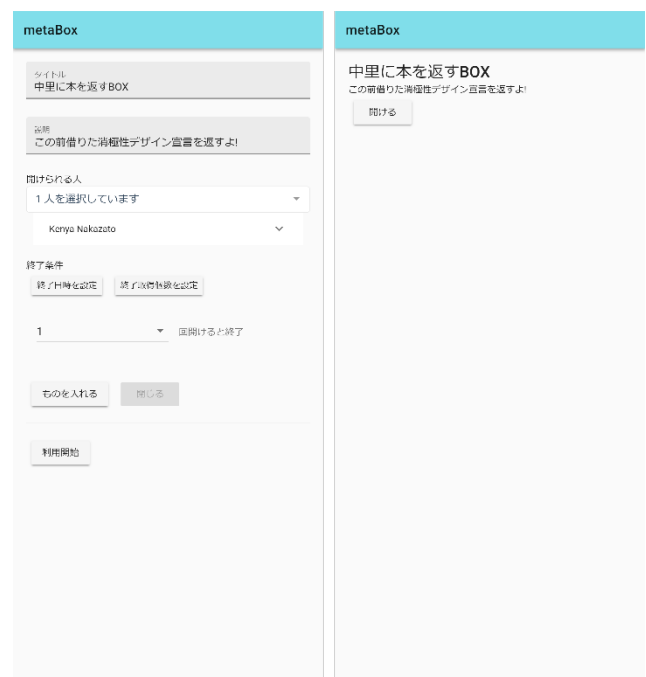


図 3 metaBox のユーザインタフェース (左: metaBox Definer. 右: プロセス中のインタフェース)

3. 実装

本章では metaBox とサーバ、metaBox Definer の実装について解説する。なお、システムアーキテクチャを図 4 に示す。

3.1 metaBox

本節では、metaBox と metaBox 内のマイコンで動くプログラムについて解説する。

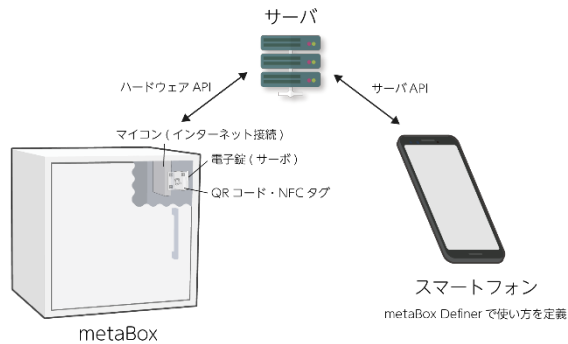


図 4 metaBox のシステムアーキテクチャ

■ ハードウェア

metaBox はインターネット経由で操作可能な電子錠を備えた箱である。metaBox は様々な用途に使えるという汎用的でメタな点の特徴のため、何色にも染まっていない白を筐体の色として選んだ。内部には扉の施錠解錠を制御するためのサーボモータとそれ制御するマイコンボードの Raspberry Pi を組み込んだ。Raspberry Pi 上では Node.js を動作させ、WebSocket でサーバに接続する。サーバからの命令を受け付けると動作するようにプログラムした。筐体前面には、サーバの URL が埋め込まれた QR コードと NFC タグを貼り付けた。

■ ハードウェア API

metaBox はサーバに接続すると複数の metaBox から一意に識別するための metaBox ID を送信する。また、metaBox はサーバに対して次の API を提供する。

- lock — 鍵の施錠
- unlock — 鍵の解錠
- keyState — 鍵の状態の取得

サーバはユーザのアクションを発端として、指定した metaBox ID の metaBox を操作できる。

3.2 サーバ

本節では、metaBox の状態を管理し、ユーザインタフェースをホスティングするサーバサイドの構造や、API 設計などについて解説する。

■ システム

サーバサイドは Node.js で実装した。Web アプリケーションフレームワークに Express を使い、クライアント向けにユーザインタフェースのホスティング、metaBox を利用するための API の公開をしている。起動中は常時 metaBox からの WebSocket での接続を待ち受け、metaBox の増減にも動的に対応する。これらをインターネットからアクセス可能にするために、Amazon Web Services の Amazon Elastic Compute Cloud 上で実行している。また、簡単のためデータベースに Firebase Realtime Database、ユーザ認証に Firebase Authentication を用いた。

■ サーバ API

サーバサイドでは、次の API を提供している。各 API は

RESTful な設計になっており、フロントエンドからの POST リクエストを受け付ける。また、この API を使用すれば metaBox を利用するアプリケーションを作成することも可能である。その場合、セキュリティを考えるとユーザ認証サーバやデベロッパアカウントの発行などが必要になる。

- api/<metaBox ID>/open — 指定した ID の metaBox の鍵を解錠する
- api/<metaBox ID>/close — 指定した ID の metaBox の鍵を施錠する
- api/<metaBox ID>/get — 指定した ID の metaBox からモノを取り出すために鍵を解錠する
- api/<metaBox ID>/set — 指定した ID の metaBox にモノを設置するために鍵を解錠する
- api/<metaBox ID>/keyState — 指定した ID の metaBox の鍵の状態を得る
- api/<metaBox ID>/info — 指定した ID の metaBox の情報を得る
- api/<metaBox ID>/meta — 指定した ID の metaBox の使い方を定義し利用開始する
- api/list — 現在稼働している metaBox のリストを得る

list 以外の各 API はそれぞれ、呼び出したユーザが認証され、実行する権利を有しているか、metaBox の操作を実行する条件を満たしているかを確認した後実行する。

metaBox の定義には meta API を利用する。リクエスト時にリクエストボディに含める解錠条件と終了条件を説明する。解錠条件は以下の通りでユーザごとに設定でき、条件属性を指定していないユーザは無条件で解錠できる。

- timeout — 指定した時間まで解錠できる
 - day-count — 一日に指定した回数まで解錠できる
- 終了条件は以下のタイプの条件を複数指定できる。なお、複数の条件を設定した場合、いずれかの条件を満たした場合に終了する。
- timeout — 指定した時間が経過すると終了する
 - count — 指定した回数解錠すると終了する

3.3 フロントエンド

metaBox Definer およびプロセス中のインタフェースは Web ページとして実装した。

4. 運用

本章では、metaBox の運用について解説し、その結果について考察する。

4.1 metaBox の運用

ユーザが metaBox を利用してどのような使い方をするのか観察するため、1 台の metaBox を 2 週間研究室で運用する実験を行った。利用できるユーザは研究室に配属された学部 2 年～修士 2 年と教員の計 32 名である。metaBox は

アクセスしやすい出入り口付近の棚の上に設置した。運用実験の結果、以下の利用があった。なお、終了条件が1週間経過に設定されたプロセスは実験期間の都合上、途中で強制終了した。

忘れ物の保管

用途: 忘れ物の保管

解錠条件: 持ち主 (無条件)

終了条件: 1 回解錠する (count)

玩具の隠蔽

用途: 研究室にあるけん玉の隠蔽

解錠条件: けん玉の利用頻度が低いユーザ (無条件)

終了条件: 1 週間経過 (timeout)

レシートの受け渡し

用途: 立て替え払いで物品を購入した際のレシートの受け渡し

解錠条件: 指導教員 (無条件)

終了条件: 1 回解錠する (count)

個人 BOX

用途: 個人の荷物入れ

解錠条件: 荷物を入れた本人 (無条件)

終了条件: 3 日経過 (timeout)

利用は4件と少なく、大きく分けるとモノの受け渡しとモノの保管という使われ方の2パターンのみで、多様な使い方はされなかった。ただし、モノの隠蔽に使うという使い方は想定外だった。そして、解錠条件に複雑な設定をする利用はなかった。

4.2 考察

metaBox の利用が少なかった原因についていくつかの視点から考察する。

■ 抽象度の高さによる使いにくさ

metaBox を利用したユーザからは、ユーザインタフェースの分かりづらさと、具体的な使用用途が示されていないと使用用途が想像できない点を指摘された。本システムは、個別のアプリケーションを作るのではなく、metaBox Definer によって metaBox の使い方を自ら定義する設計にした。結果として、複雑なユーザインタフェースになってしまい、分かりづらくなってしまったと考える。

また、多様な使い方をするためには、どのように設定すればどのように使えるかを提示すればよかったと考える。身近にある汎用的なメディアとしてスマートフォンがある。アプリとしてあるものに特化したユーザインタフェースを提供することで、専用機のような体験とそれができるとユーザに認知させている。metaBox のユーザインタフェースも解錠条件や終了条件のプリセットを適用できるようにしておき、想定していた使い方のプリセットを用意して

おけば、プリセットに手を加えた派生の使い方などが生まれていた可能性がある。

■ 価値の移動

運用の結果、利用用途としてモノの受け渡しとモノの保管という2パターンの使われ方があった。モノの受け渡しの注目すると、モノという価値が人から人へ移動していることがわかる。今回の運用では、忘れ物や領収書などのモノが元々所有しているべき人に届ける場面に用いられているが、譲渡や売買などといった使われ方は実験期間が短かったこともあり、なかった。そもそもモノには価値があるため、モノが人から人へ移動することは、価値が一方向的に流れることになる。送金 API などを用意することで、モノに対する対価を支払うことができるようになり、双方向的な価値の流れができる。その結果、利用シーンやアプリも増えると考えられる。

■ 状態の不明瞭さ

運用時に出た意見に metaBox がどのような状態なのかわからないという意見もあった。また、今回は1台で運用していたが、同じ場所に複数の metaBox がある場合も同様に自分の使う metaBox はどれか、metaBox がプロセス中か否か、スマートフォンをかざしてユーザインタフェースを表示するまでわからない。exUI ではプロダクトの操作中に AR を用いてプロダクト自体にプロダクトの機能を重量表示して解決している。将来的に AR グラスなどが普及すれば、常に metaBox の状態を認識できそうである。そして、exUI の設計思想から少々反れてしまうが、現状ではディスプレイを設置して状態を表示することで解決できる。他に筐体を一部透明にして内部状態を提示する解決方法もある。この場合、metaBox を置く場所によってはプライベートなモノには使いづらいという欠点もあるが、低コストで実現できる。

5. 議論

現在、運用の結果を踏まえた、新たな metaBox を開発中である。筐体は前面を透明にし、内部の状態を見えるようにした。また、セキュリティ強化のため、小型のディスプレイを設置し一定期間で動的に変化する QR コードを表示するようにした。スマートフォンのユーザインタフェースはアプリごとに個別のものを用意する設計にし、metaBox でできることを明確にした。筐体の QR コードを読みこむと metaBox のポータルページが開くようになっており、そこで使いたいアプリを選択すると、そのアプリのユーザインタフェースを表示する仕組みである。将来的には、送金機能も組み込む予定である。

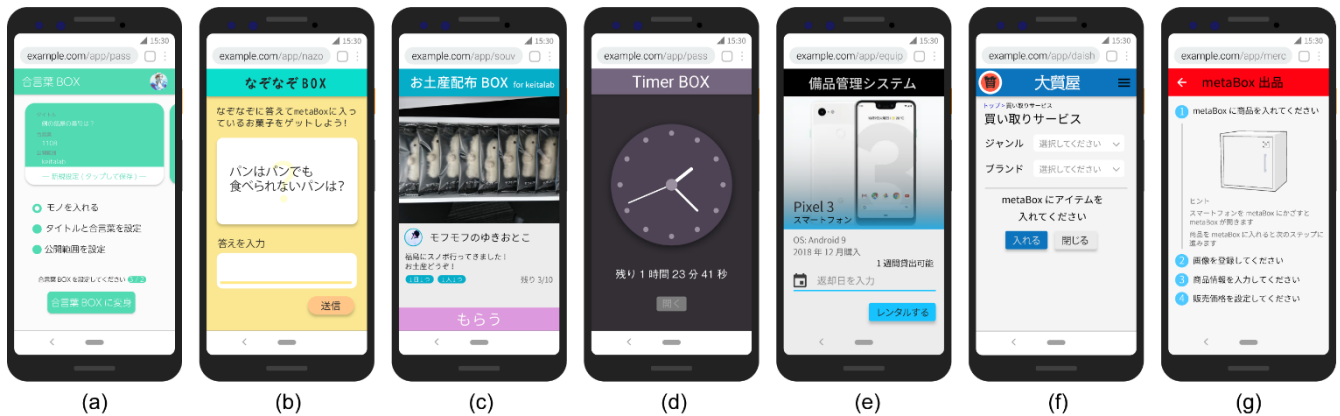


図 5 想定する metaBox のアプリケーション例

5.1 我々が考えたアプリケーション例

今回試作した metaBox は運用実験において多様な使われ方をしなかったが、現在開発中の新しいシステムではより多様な用途に使えると考える。図 5 は想定するアプリケーション例である。(a)～(e) は特定のコミュニティ内で利用するアプリの例で、(f)、(g) はサービスのインタフェースとして metaBox を利用するアプリの例である。なお、ここで紹介しているアプリケーションの一部は試作した metaBox でも解錠条件や終了条件の組み合わせにより同様の使い方が可能である。

(a) **合言葉 BOX**: 合言葉 BOX は合言葉を知っている人しか開けることができない箱で、プロジェクト内で合言葉を共有することで簡単にモノの共有ができる。

(b) **なぜなぜ BOX**: なぜなぜに正解すると metaBox を開けることができ、オフィスやラボで使える遊び心のあるアプリである。

(c) **お土産配布 BOX**: 一人 1 度のみ開けられるように設定でき、一人が複数のお土産を食べてしまうことなく、広く行き渡らせることができる。

(d) **Timer BOX**: 指定した時刻まで metaBox を開けることができなくなるため、禁欲のためにモノをしまっておける。

(e) **備品管理システム**: 誰が何を借りたかを自動でログを残せるため、備品管理のコスト削減を望める。

(f) **買い取りサービス**: metaBox に不要物を入れ、買い取りを依頼する。

(g) **フリマサービス**: フリマサービスの出品に metaBox を利用できるサービスである。購入者が直接 metaBox から商品を取り出したり、宅配業者が取り出して宅配したりできる。

5.2 ハードウェア構成

アプリケーションを作る上で制約となるのが、metaBox のハードウェア構成である。例えば、内部を冷やす機能を持たせれば飲食物関係のアプリケーションができ、ガスセンサを持たせれば内容物の鮮度を測り、適切なタイミングで破棄できるアプリケーションも可能である。基本的にア

プリケーションの多様性とハードウェアの複雑性はトレードオフの関係にある。ただし、カメラによる画像の取得は活用の幅が広く、人が見るために使うだけでなく機械学習などと組み合わせることでモノの状態を推定することもできるため、様々なアプリケーションに活用できることが期待できる。このような多用途に活用できるハードウェアコンポーネントを組み入れることで、最小コストで多彩なアプリケーションを展開できる。

5.3 metaBox の評価

metaBox はアプリケーションの多様さや汎用性が評価の一つの軸になると考える。図 5 にていくつかのアプリケーションを提示したが他にも Web サービスと連携したアプリや IoT 機器と連携したアプリ、スマートコントラクトを利用したアプリなどが考えられ、metaBox を使ったアプリケーションの幅は広いと考える。これまで、箱は人がモノを入れて主に保管するために使われていた。metaBox は箱をインターネットに繋げ、exUI によってユーザインタフェースを提供することで、Web 上や身の回りのフィジカルな情報などを含めた様々なインタラクションが可能になった。これはメタメディアであるスマートフォンがアプリによって多様なインタラクションを可能にしていることに似ている。さらに、metaBox はスマートフォンのモノに対するインタフェースであるとも考えると、ハードウェア的な制約を除き、metaBox の汎用性はおよそスマートフォンで可能なインタラクションの多様さであると考えられる。

6. おわりに

本研究では、モノのやり取りに箱が使われていることに着目して、exUI 設計に基づいた IoT Box, metaBox を試作した。そして、実運用の結果からは多様な使い方は見られず、その原因について考察した。最後に、考察で得られた問題点を解決することで多様なアプリケーションの可能性を議論した。

参考文献

- [1] “メルカリ スマホでかんたん フリマアプリ”.
<https://www.mercari.com/jp/>, (参照 2019-06-25)
- [2] “ラクマ (旧フリル) | 楽天のフリマアプリ - 中古/未使用品
がお得!”. <https://fril.jp/>, (参照 2019-06-25)
- [3] “Amazon | 本, ファッション, 家電から食品まで | アマゾン”.
<https://www.amazon.co.jp/>, (参照 2019-06-25)
- [4] “【楽天市場】 Shopping is Entertainment! : インターネット最
大級の通信販売、通販オンラインショッピングコミュニテ
ィ”. <https://www.rakuten.co.jp/>, (参照 2019-06-25)
- [5] “Amazon.com: : Amazon Go”.
<https://www.amazon.com/b?ie=UTF8&node=16008589011>, (参照
2019-06-25)
- [6] “オフィス向け無人コンビニ 600 | キャッシュレスで決済”.
<https://www.600.jp/>, (参照 2019-06-25)
- [7] 金子翔麻, 中里健也, 高田一真, 渡邊恵太. exUI: ユーザイン
タフェースを外在化する手法とプロダクトデザイン. 第 26
回インタラクティブシステムとソフトウェアに関するワーク
ショップ(WISS2018), 2018.