

卒業研究報告

題目

サウンドコード技術を利用した IOT セキュリティシステムのシステム開発

指導教官

高橋寛教授

王森レイ

報告者

青野智己

平成 30 年 2 月 9 日提出

愛媛大学工学部情報工学科計算機システム工学講座

目次

第 1 章 まえがき	1
第 2 章 諸定義	3
第 3 章 UML を用いたシステム設計	7
3.1 要求定義	7
3.2 設計	9
第 4 章 実装および検証	14
4.1 実装	14
4.2 検証結果	16
第 5 章 評価と考察	21
第 6 章 あとがき	23
謝辞	24
参考文献	25

第 1 章

まえがき

サウンドコードとは文字情報を信号処理技術によって音声に変換して受送信する新しい音声通信技術である。サウンドコードは普段人間の耳には聞こえない高周波音であり、通常の身近なデバイスのスピーカーとマイクで受送信可能であるという特徴を持つ。サウンドコード技術を利用した IOT セキュリティシステムは、従来の IOT セキュリティシステムに比べ、聞こえない音のため暗証番号錠のような暗証番号漏れがなく、また、ID カード錠のようなカード紛失のリスクを持たないという利点がある。さらにスマートフォン、タブレットで利用可能であり専用のデバイスが不要であるため導入コストを低くできる。そのため、サウンドコード技術を利用することにより革新的な IOT セキュリティシステムの開発を行うことができると考える。

本研究の目的は、サウンドコード技術を利用した IOT セキュリティシステムの応用開発および評価を行うことで、IOT セキュリティシステムにサウンドコード技術を利用する利点と問題点を考察することである。

サウンドコード技術を利用した IOT セキュリティシステムの応用として宅配ボックスシステムを提案する。その理由を以下に述べる。現在、宅配業者による配達の 23.5% が不在配達であり、配達回数の増加から年間 418,271t の CO₂ が排出されている [1]。このように、再配達により宅配業者の労働生産性と環境面で大きな損失が発生している。サウンドコード技術を用いて安価で機密性の高い宅配ボックスシステムを開発することにより、宅配業者の再配達を大幅に減らすことができると考える。

本研究ではグループ（青野智己、井早千裕、山田琢也）で、V 字モデルに従って宅

配ボックスシステムの開発を行う。要求定義や設計ではシステム化対象を厳密に定義するために UML 図を利用する。次に設計の中から実装対象を決定し、3 人で分担してボードコンピュータの Raspberry Pi に実装を行う。そして、UML 図を基に実装したシステムの評価をすることで今後の課題を考察する。グループのスケジュール管理は brabio というガントチャートツールを用いる。

本論文は以下のような構成をとる。第 2 章では用語、V 字モデルおよび宅配ボックスシステム概要の説明を述べる。第 3 章では UML 図を用いて宅配ボックスシステムの要求定義および設計を示す。第 4 章では本研究で実装する対象と検証結果を示す。第 5 章では実装したシステムの評価および考察を示す。第 6 章では本研究のまとめを行う。

第 2 章

諸定義

本章では、本論文で用いる用語、V 字モデルの開発フローおよび宅配ボックスシステムの概要について述べる。

サウンドコード

文字情報を信号処理技術によって音声に変換して受送信する音声通信技術である。

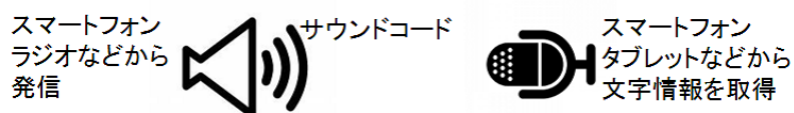


図 2.1. サウンドコード通信技術

図 2.1 のように、スピーカから発信したサウンドコードをマイクを持つ端末で解析することにより、端末から端末に音で文字情報を送信することができる。サウンドコードは 16 進数で 32bit の文字情報で構成されるライセンスキーと 16 進数で 4bit の文字情報で構成される UID を持つ。

UML (Unified Modeling Language)[2]

オブジェクト指向のソフトウェア開発において、データ構造や処理の流れなどソフトウェアに関連する様々な設計や仕様を図示するための記法を定めたものである。UML

で定義される図にはユースケース図、シーケンス図、クラス図などがある。

ユースケース図 [2]

システムが外部に提供する機能を表現する図である。システム開発において、利用者の要求を分析してシステムが果たすべき役割を明確化するために作成されることが多い。

シーケンス図 [2]

要素間の相互作用を時系列で表したものである。相互作用図の一種で、図の上から下に向かって時間の流れが表される。

クラス図 [2]

システムを構成するクラスと、クラス間の相互の関係を表現する図である。システム全体の静的な構造を明らかにするために作成される。

V 字モデル [3]

以下の図 2.2 に V 字モデルの開発フローを示す。

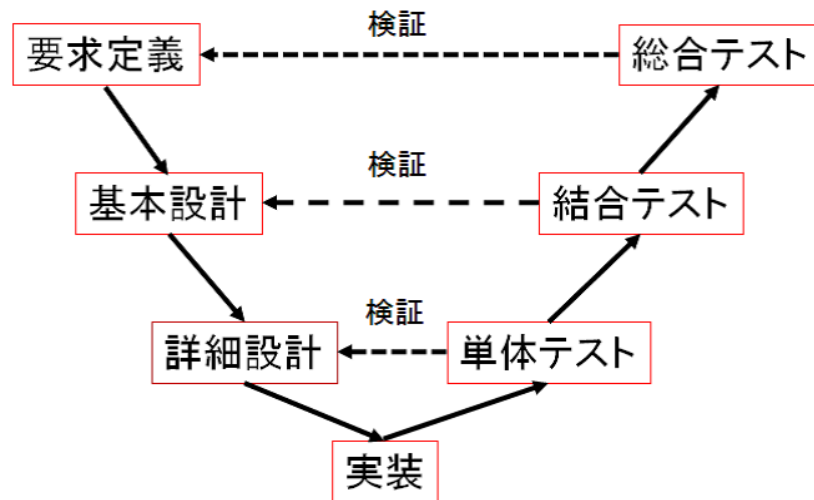


図 2.2. V 字モデル

図 2.2 はシステムの開発と確認の流れを模式的に示した様子である。横軸は開発の時間軸であり、縦軸は詳細化の程度を表している。開発の要求定義、基本設計、詳細設計の過程を実装後に、それぞれ単体テスト、結合テスト、総合テストによって検証することが V 字モデルの特徴である。

宅配ボックスシステム

宅配ボックスシステムの概要を以下の図 2.3 に示す。

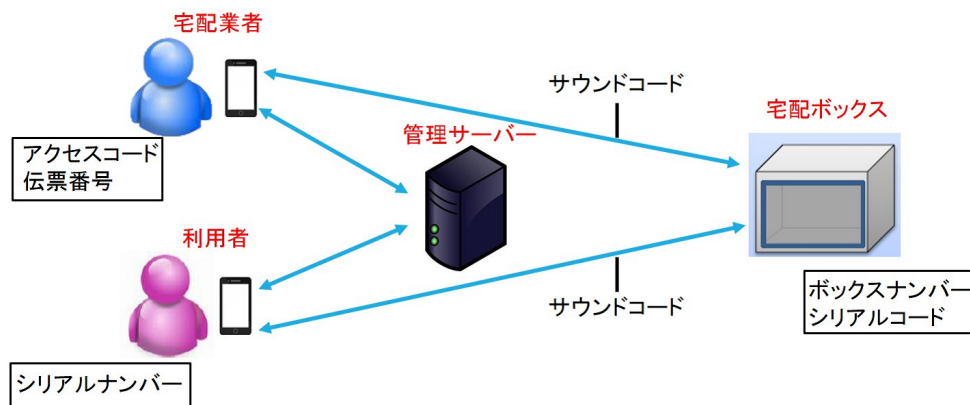


図 2.3. 宅配ボックスシステム概要図

宅配ボックスシステムは宅配業者が配達のため、利用者を訪問した際に不在の場合でも、宅配ボックスに荷物を預け入れることにより配達を可能にする。宅配業者と利用者はスマートフォンアプリを通じて宅配ボックスの扉の開け閉めを行える。また、管理サーバは、利用者および宅配業者の個人登録番号とボックスの識別番号や商品の受領書およびボックス内の荷物有無情報を管理する。通信手段として、スマートフォンアプリと宅配ボックス間はサウンドコード、スマートフォンアプリと管理サーバ間はインターネットを用いるものとする。

利用者、宅配業者は個人登録用のデータとしてそれぞれシリアルナンバー (7bit)、アクセスコード (8bit) を持っている。宅配ボックスは箱の識別番号としてシリアルコード (8bit) およびボックスナンバー (3bit) が当てられている。

宅配ボックス、スマートフォンアプリ、管理サーバを開発対象とし、実装をする際にはそれぞれ、山田琢也、井早千裕、青野智己が担当をする。

第 3 章

UML を用いたシステム設計

本章では、V 字モデルにおける要求定義および設計について述べる。本章の構成は以下の通りである。3.1 節ではユースケース図を用いて、宅配ボックスシステムの要求定義を説明する。3.2 節では、シーケンス図およびクラス図を用いて、宅配ボックスシステムの基本設計および詳細設計を述べる。

3.1 要求定義

ユーザが宅配ボックスシステムに求める機能や動作を決定する。以下に要求定義をまとめた図 3.1 を載せる。

図 3.1 では、ユーザ（利用者、オーナー、宅配業者）がシステム（宅配ボックス、管理サーバ）に対してどのような動作を行えるのかをユーザ視点で記載してある。利用者はシステムに対して荷物の取り出しと利用者登録が可能であり、宅配業者はシステムに対して荷物の預け入れと利用者登録が可能である。取り出し、登録、預け入れパッケージにはより具体的な動作が記載されており、どのユーザがどのシステムに関連しているかが線分で示されている。オーナーは利用者の汎化関係であり、オーナーのみが利用者登録の変更を行うことができる。

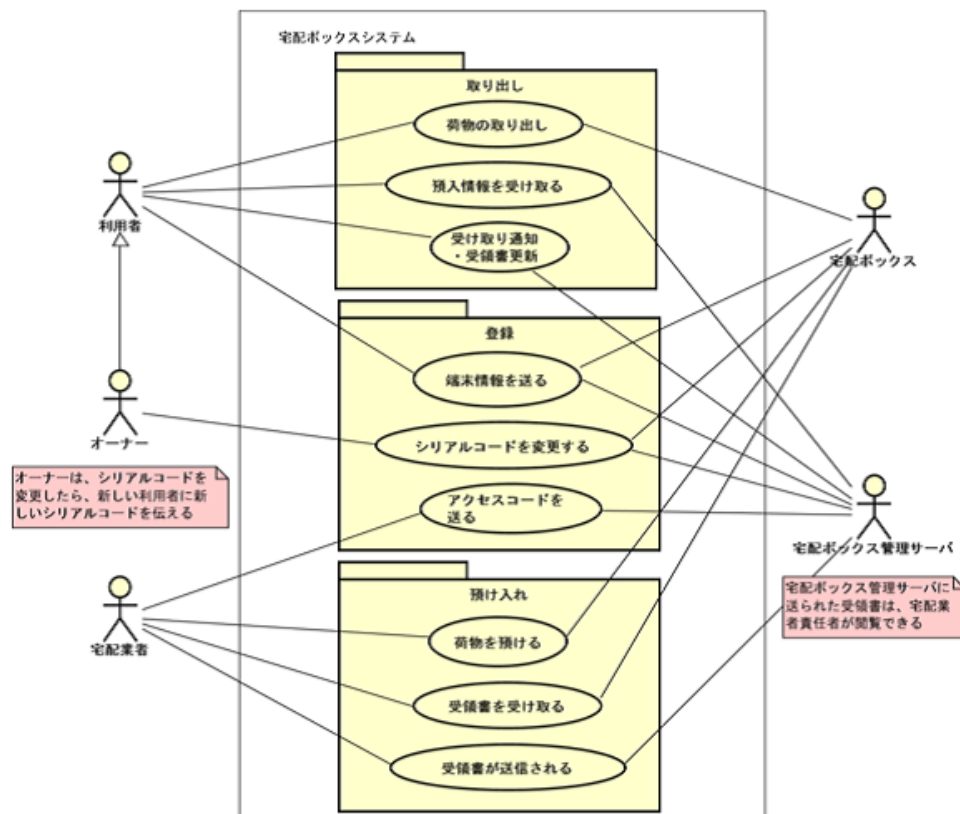


図 3.1. システムのユースケース図

3.2 設計

以下に基本設計および詳細設計を示した宅配ボックスシステムのクラス図を図 3.2 に載せる。

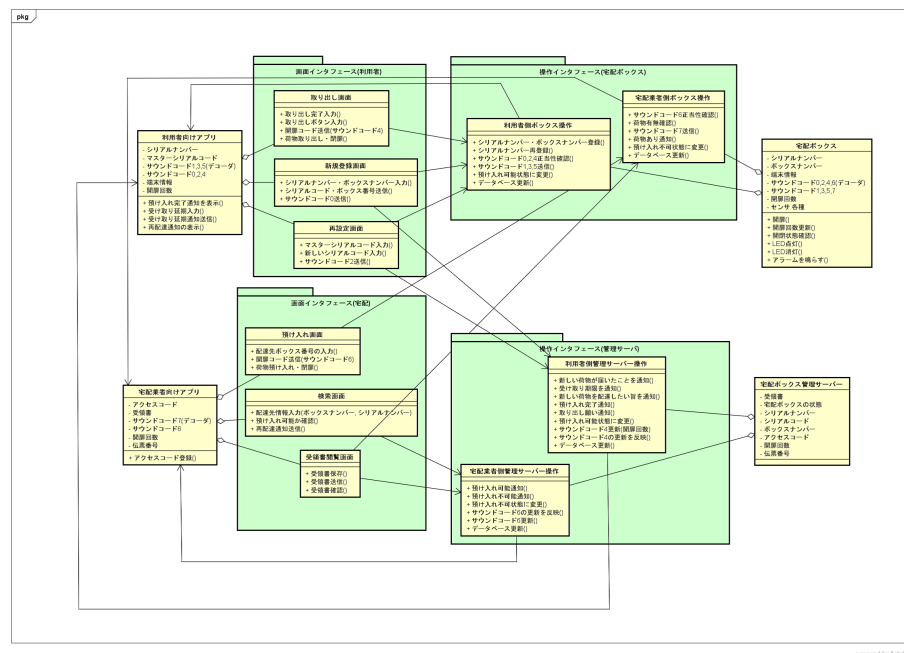


図 3.2. システムのクラス図

図 3.2 は利用者向けアプリ、宅配業者向けアプリ、宅配ボックス、管理サーバをクラスとしたときにそれぞれが持つ属性と操作を静的に表した様子を示している。画面インタフェース(利用者)パッケージは利用者向けアプリが持つ画面クラスをまとめたものである。同様に画面インタフェース(宅配)パッケージは宅配業者向けアプリが持つ画面クラスをまとめている。操作インタフェース(宅配ボックス)パッケージが持つ利用者側ボックス操作クラスは、宅配ボックスと利用者向けアプリに関連のある操作をまとめており、宅配業者側ボックス操作クラスは、宅配ボックスと宅配業者側向けアプリに関連のある操作をまとめている。操作インタフェース(管理サーバ)パッケージが持つクラスも操作インタフェース(宅配ボックス)パッケージが持つクラスと同様にまとめている。

図 3.2 に記述されてあるシステムの機能について説明をする。

- ・登録

利用者アプリの属性であるシリアルナンバーと宅配ボックスの属性であるボックスナンバーおよびシリアルコードを、宅配ボックスと管理サーバのデータベースで紐づけられると登録完了となる。マスターシリアルコードを持つオーナーが再設定を行った場合、指定したボックスナンバーおよびシリアルコードを持つデータベースに新たなシリアルナンバーが登録されれば設定完了となる。

- ・取り出し

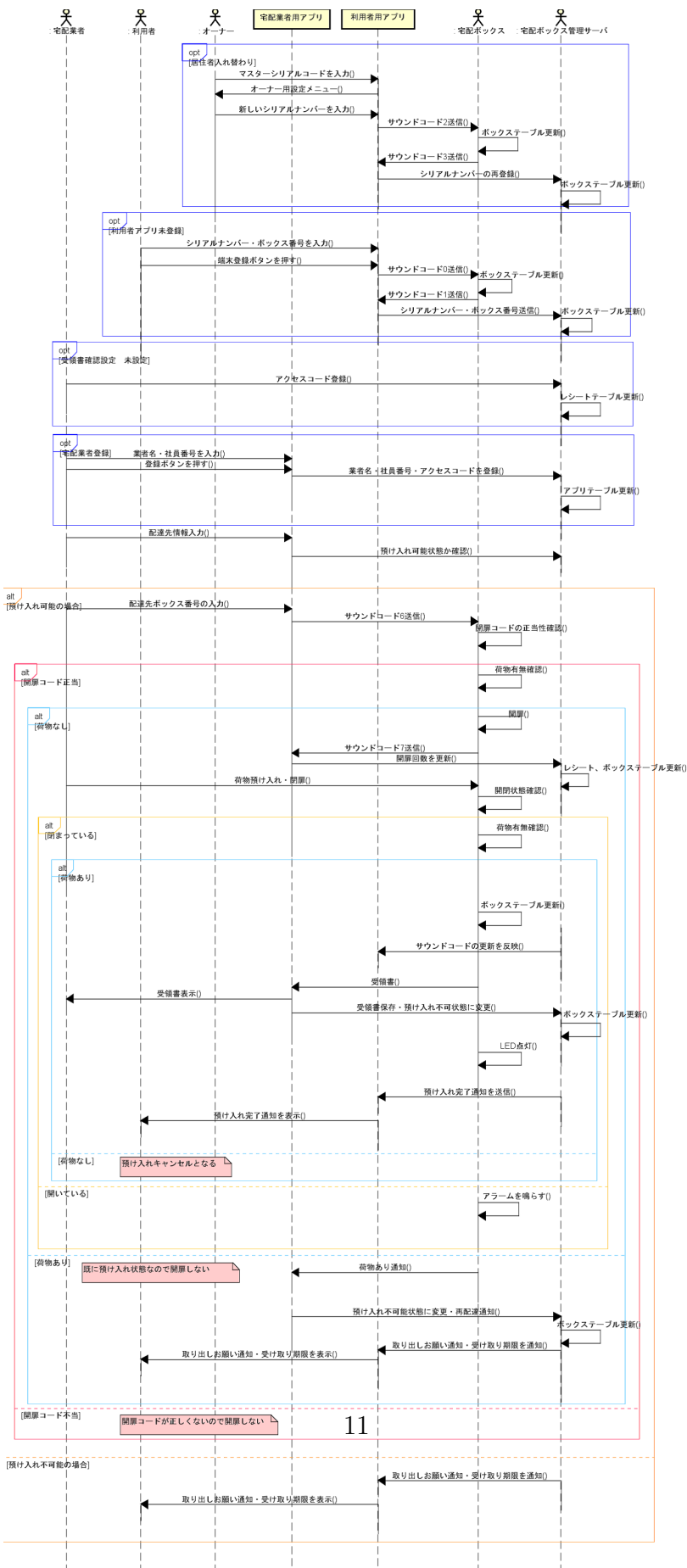
利用者は利用者向けアプリからシリアルナンバーを宅配ボックスに送信すると、宅配ボックスはデータベースに登録されてあるシリアルナンバーと一致したならば扉が開き、荷物の取り出しが可能となる。扉を閉めると宅配ボックスの中の荷物有無の状態がアプリを通じて管理サーバに保存される。また、扉が開いた際には、新しいサウンドコードの生成が必要であるため、宅配ボックスと管理サーバのデータベースにある開扉回数を更新する。これは、預け入れで開扉した場合にも行われる。

- ・預け入れ

宅配業者は宅配業者向けアプリからボックスナンバーを指定すると、そのボックスナンバーを持つ宅配ボックスは荷物が預け入れられてない場合のみ開扉して、荷物を預け入れることができる。扉を閉めると宅配ボックスの中の荷物有無の状態がアプリを通じて管理サーバに保存される。同時に宅配業者向けアプリから受領書が発行され管理サーバに保存される。その後、管理サーバは利用者向けアプリに荷物預け入れ通知を行う。

その他に宅配業者アプリは受領書履歴を確認することができ、指定したボックスナンバーとシリアルコードに対応した宅配ボックスの荷物有無状態を確認できる機能がある。

以下の図 3.3 が宅配ボックスシステムの基本設計となるシーケンス図である。



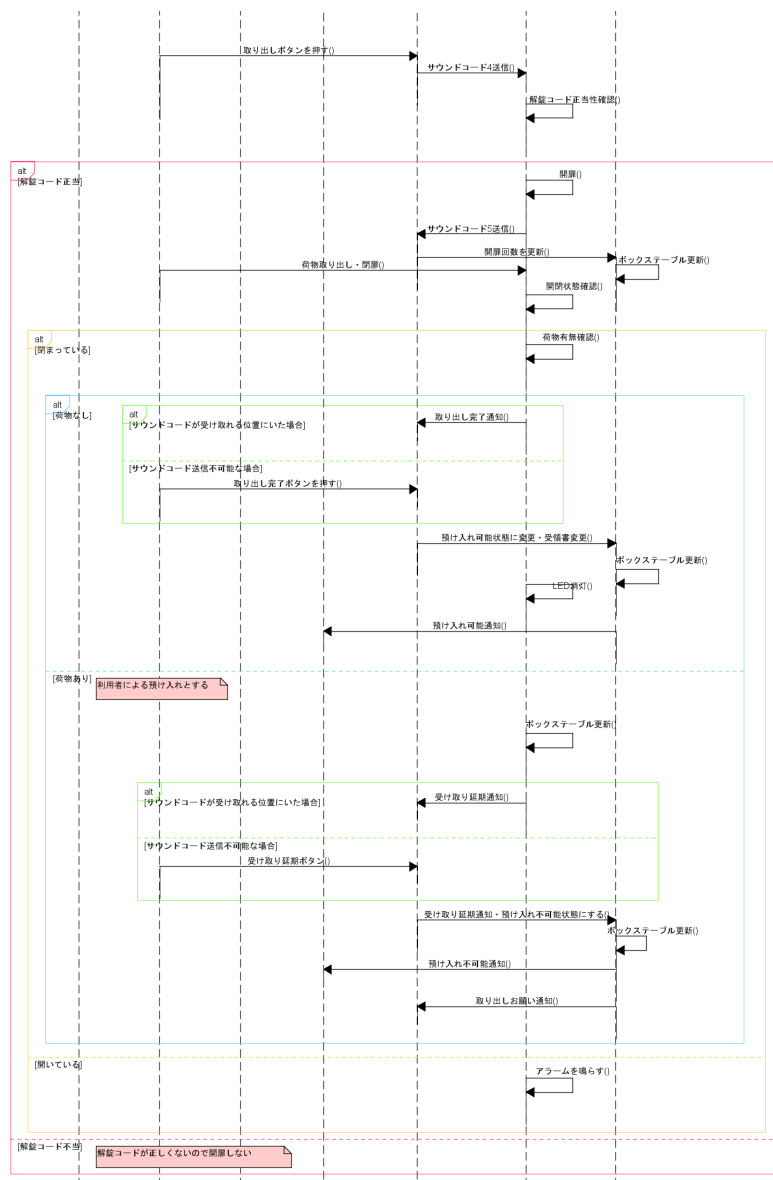


図 3.3. システムのシーケンス図

図 3.3 のシーケンス図は図 3.1 のユーザとシステムと利用者向けアプリ、宅配業者向けアプリのオブジェクトの間で行われる操作を時系列で表している。図 3.3 では複合フラグメントの opt,alt を使用している。opt はガード内で記述された条件が満たされている場合に処理が行われる。alt は処理が点線で区切られており、ガード内で記述された条件が満たされた場合は点線までの処理が行われ、そうでない場合は点線以降の処理が行われる。これにより扉が開かなかった場合やその他のヒューマンエラー時に動作するシステムの処理を明確に定義している。

シーケンス図から読み取れるように宅配ボックスシステムは登録→預け入れ→取り出しの流れで使用される。また、登録、預け入れ、取り出しの操作はすべて、アプリ→ボックス→アプリの順にサウンドコードで通信した後、アプリから管理サーバにデータを送信するという流れになっている。サウンドコード通信やデータ送信の際に各操作ごとに適切なデータを送ることで、宅配ボックスと管理サーバが直接通信を行わないでデータの同期を図ることができる設計となっている。

第 4 章

実装および検証

本章では V 字モデルにおける実装及び検証について述べる。本章の構成は以下の通りである。4.1 節では、設計から実装を行う高優先度の機能を抽出して、それに合わせた実装環境について述べる。4.2 節では、実装したシステムを単体テスト、結合テストおよび総合テストを行うことによりそれぞれ、詳細設計、基本設計および要求定義を検証した結果について述べる。

4.1 実装

3.2 節でのシステム設計では、システムを使う際にヒューマンエラーが発生した場合の処理を記述していたが、今回の実装では考えないものとする。すなわち、登録→預け入れ→取り出しが常に正しく行われるシナリオを想定する。また、開扉回数によるサウンドコードの更新は行われないものとする。このシナリオを満たす機能が高優先度の機能である。本実験では高優先度の機能のみを実装する。高優先度の機能を以下の図 4.1 に示す。

なお、登録の通信方法は、システム設計の段階ではサウンドコード通信だったが、図 4.1 では PHP 通信に変更されてある。

以下の表 4.1 にスマートフォンアプリ、宅配ボックス、管理サーバの実装環境と周辺機器を示す。

本研究の実装ではアプリのユーザからのインタフェースは Web ブラウザとする。

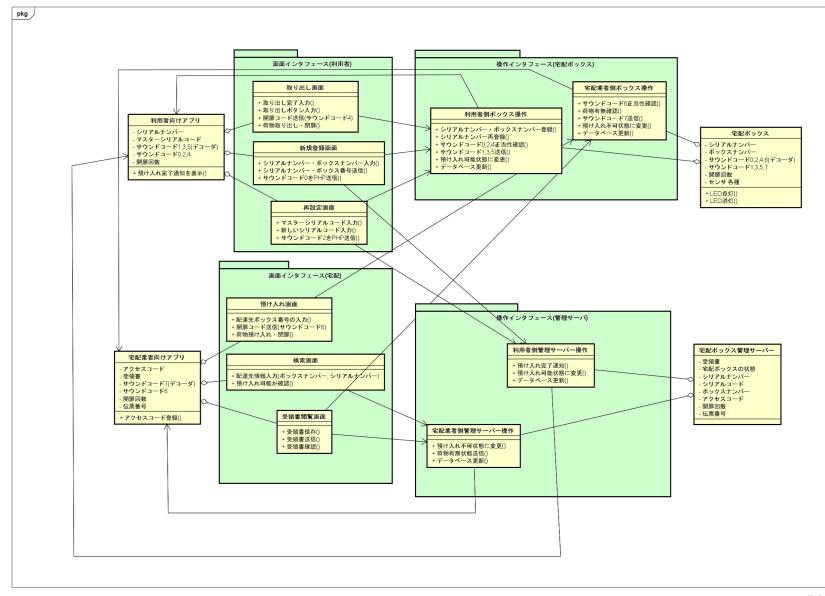


図 4.1. 高優先度の機能項目

表 4.1. 実装環境

	実装環境	周辺機器	開発言語
アプリ	Raspberry Pi 3 Model B	スピーカー, マイク	PHP7, C++
宅配ボックス	Raspberry Pi 3 Model B	スピーカー, マイク, LED, スイッチ	PHP7, MySQL, C++
管理サーバ	Raspberry Pi 3 Model B		PHP5, MySQL

Raspberry Pi にサウンドコードのエンコーダとデコーダを実装することで、周辺機器のスピーカーとマイクからサウンドコードの通信を可能にする。また、PHP によるデータの通信も可能である。

宅配ボックスも同様に Raspberry Pi にサウンドコードのエンコーダとデコーダを実装することで、サウンドコードの通信を可能にする。また、PHP による通信も可能である。スイッチは荷物の有無を切り替えることに使用する。荷物の有無やボックスの開扉などは LED の点灯で確認ができる。

管理サーバは PHP 通信によりアプリからデータを受け取ることでデータベースのテーブルを更新する。

4.2 検証結果

V 字モデルに従い単体テスト、結合テスト、総合テストを行うことで、実装したシステムが機能項目、基本設計、高優先度のシナリオを満たしているかを確認する。各実装対象の単体テスト内容を表 4.2, 表 4.3 および表 4.4 にそれぞれ示す。

表 4.2. 単体テスト (アプリ)

	確認内容
1	ブラウザ上のテキストフォームからサウンドコード0のライセンスキーを作成後 PHPで送信
2	サウンドコード1のライセンスキーを取得後、管理サーバにデータ送信
3	ブラウザ上のテキストフォームからサウンドコード2のライセンスキーを作成後 PHPで送信
4	サウンドコード3のライセンスキーを取得後、管理サーバにデータ送信
5	ブラウザ上の預け入れ入力によりサウンドコード6をエンコードし、音声出力
6	サウンドコード7をデコードできる
7	サウンドコード7をデコードした後、管理サーバにデータ送信
8	ブラウザ上の取り出し入力によりサウンドコード4をエンコードし、音声出力
9	サウンドコード5をデコードできる
10	サウンドコード5をデコードした後、管理サーバにデータ送信
11	設定したアクセスコードの受領書が見れる
12	シリアルコード、ボックスナンバーをPHPで送信後、箱状態変数が見れる
13	宅配業者の情報をPHPで送信しサーバーに登録する

表 4.3. 単体テスト (管理サーバ)

	確認内容
1	サウンドコード0のライセンスキーを取得後、boxテーブルが正しく作成される
2	サウンドコード2のライセンスキーを取得後、boxテーブルのシリアルコード、ボックスナンバーに対応するシリアルナンバーが正しく更新される
3	サウンドコード4のライセンスキー、箱状態変数を取得後、boxテーブルのボックスナンバー、シリアルコードに対応した箱状態変数が正しく更新される
4	サウンドコード6のライセンスキー、箱状態変数、預け入れ時間、伝票番号、シリアルコードを取得後、テスト項目5,6を確認
5	boxテーブルのボックスナンバー、シリアルコードに対応した箱状態変数が正しく更新される
6	receiptテーブルが正しく作成される
7	アクセスコード、宅配会社を取得後、apliテーブルが正しく作成される
8	アクセスコードを受け取り、それに対応したreceiptテーブルを送信する
9	ボックスナンバー、シリアルナンバーを受け取り、boxテーブルからそれらに対応した箱状態変数を送信する
10	テスト項目4の動作の後に利用者アプリに通知をする

表 4.4. 単体テスト (ボックス)

	確認内容
1	サウンドコード0のライセンスキーをPHPで取得後boxテーブルが正しく作成される
2	サウンドコード1をエンコードし、出力する
3	サウンドコード2のライセンスキーをPHPで取得後boxテーブルが正しく更新される
4	サウンドコード3をエンコードし、出力する
6	サウンドコード6のデコードができる
7	サウンドコード6のデコードがされた場合、開扉する
8	LEDが光る
9	スイッチを押すことにより箱状態変数が1となる
10	boxテーブルを更新
11	サウンドコード7をエンコードし、出力する
12	LEDが消える
13	サウンドコード4のデコードができる
14	サウンドコード4のデコードがされた場合、開扉する
15	LEDが光る
16	スイッチを押すことにより箱状態変数が0となる
17	apliテーブルを更新
18	サウンドコード5をエンコードし、出力する
19	LEDが消える

これらの単体テストの内容を確認することにより、各実装対象が図 4.1 の機能項目を満たしていることを確認する。

次に表 4.5 に結合テストの内容を示す。

表 4.5. 結合テスト

	対象	確認内容
1	利用者アプリ→ボックス	サウンドコード0のライセンスキーをPHPで送信
2	ボックス→利用者アプリ	サウンドコード1を音声出力
3	利用者アプリ→管理サーバ	サウンドコード0のライセンスキーをPHPで送信
4	宅配アプリ→ボックス	アクセスコードをPHPで送信
5	宅配アプリ→管理サーバ	アクセスコードをPHPで送信
6	利用者アプリ→ボックス	サウンドコード2のライセンスキーをPHPで送信
7	ボックス→利用者アプリ	サウンドコード3を音声出力
8	利用者アプリ→管理サーバ	サウンドコード2のライセンスキーをPHPで送信
9	宅配アプリ→ボックス	サウンドコード6を音声出力
10	ボックス→宅配アプリ	サウンドコード7を音声出力
11	宅配アプリ→管理サーバ	サウンドコード6のライセンスキー、箱状態変数、預け入れ時間、伝票番号、シリアルコードをPHPで送信
12	管理サーバ→利用者アプリ	荷物が預け入れられたことを通知
12	利用者アプリ→ボックス	サウンドコード4を音声出力
13	ボックス→利用者アプリ	サウンドコード5を音声出力
14	利用者アプリ→管理サーバ	サウンドコード4のライセンスキー、箱状態変数をPHPで送信
15	宅配アプリ→管理サーバ	アクセスコードをPHPで送信
16	管理サーバ→宅配アプリ	受領書(預け入れ時間、ボックスナンバー、シリアルコード)を送信
17	宅配アプリ→管理サーバ	シリアルコード、ボックスナンバーをPHPで送信
18	管理サーバ→宅配アプリ	箱状態変数を送信

結合テストの内容を確認することにより、各実装対象の間で基本設計に従ったデータ通信が行われていることを確認する。

最後に表 4.6 に総合テストの内容を示す。

総合テストでは架空のシリアルナンバー、ボックスナンバー、シリアルコード、アクセスコードを想定して、新規登録→預け入れ→取り出し→再設定→預け入れ→取り出しの操作を行い、正常に動作することを確認した。従って、想定していたシナリオを満たす宅配ボックスシステムを実装することができたことを確認する。

表 4.6. 総合テスト

	対象	確認内容
1	利用者アプリボックス	ブラウザテキスト欄のシリアルコード、シリアルナンバー1、ボックスナンバーからサウンドコード0を作成後、ライセンスキーをPHP送信
2	ボックス	取得したライセンスキーをboxテーブルのボックスナンバー、シリアルコードと参照して、正当ならテスト項目番号2.1へ、そうでないならテスト項目番号2.2へ
2.1	ボックスー利用者アプリ	データベースのboxテーブルを更新し、サウンドコード1(true)を音声出力
2.2	ボックスー利用者アプリ	サウンドコード0(false)を音声出力
3	利用者アプリー管理サーバ	デコードしたサウンドコード1がtrueならサウンドコード0のライセンスキーを管理サーバへPHP送信
4	管理サーバ	データベースにboxテーブルを作成
5	宅配アプリーボックス	アクセスコードをPHPで送信後、ボックス側でapliテーブルを作成
6	宅配アプリー管理サーバ	アクセスコードをPHPで送信後、管理サーバ側でapliテーブルを作成
7	利用者アプリーボックス	ブラウザテキスト欄のシリアルコード、シリアルナンバー2、ボックスナンバーからサウンドコード2を作成後、ライセンスキーをPHP送信
8	ボックス	取得したライセンスキーをboxテーブルのボックスナンバー、シリアルコードと参照して、正当ならテスト項目番号8.1へ、そうでないならテスト項目番号8.2へ
8.1	ボックスー利用者アプリ	データベースのboxテーブルを更新、サウンドコード3(true)を音声出力
8.2	ボックスー利用者アプリ	サウンドコード3(false)を音声出力
9	利用者アプリー管理サーバ	デコードしたサウンドコード3がtrueならサウンドコード2のライセンスキーを管理サーバへPHP送信
10	管理サーバ	データベースのboxテーブルを更新
11	宅配アプリー管理サーバ	検索画面からボックスナンバー、シリアルコードを入力して、預け入れ可能が確認
12	宅配アプリーボックス	ボックスナンバーを入力し、作成したサウンドコード6を音声出力
13	ボックス	デコードしたサウンドコード6が正当でありかつ荷物状態変数が0の場合テスト項目番号13.1へ そうでなければテスト項目番号13.2へ
13.1	ボックスー宅配アプリ	LEDが点灯、スイッチを押すことによりboxテーブルの荷物状態変数1に変更後サウンドコード7を音声出力
13.2	ボックス	LEDが点灯しない
14	宅配アプリー管理サーバ	サウンドコード7をデコード後、サウンドコード6のライセンスキー、荷物状態変数、シリアルナンバー、伝票番号、預け入れ時間を管理サーバに送信
15	管理サーバ	receiptテーブルを作成、apliテーブル、boxテーブルを更新
16	管理サーバー利用者アプリ	預け入れ通知
17	宅配アプリー管理サーバ	検索画面からボックスナンバー、シリアルコードを入力して、預け入れ可能が確認
18	宅配アプリーボックス	サウンドコード6を送信し、黄色のLEDが点灯しないことを確認
19	利用者アプリー管理サーバ	受領書履歴確認
20	利用者アプリーボックス	取り出しボタンからサウンドコード4を生成して、音声出力
21	ボックス	サウンドコード4をデコードして得られたデータとboxテーブルの値を参照し、正当ならテスト項目番号21.1へ、そうでなければテスト項目番号21.2へ
21.1	ボックスー利用者アプリ	LEDが点灯、スイッチを押す場合、boxテーブルの荷物状態変数0に変更後サウンドコード5を音声出力 スイッチを押さない場合、荷物状態変数1のサウンドコード5を音声出力
21.2	ボックス	LEDが点灯しない
22	利用者アプリー管理サーバ	サウンドコード5をデコード後、サウンドコード4のライセンスキー、箱状態変数を管理サーバにPHP送信
23	管理サーバ	boxテーブルを更新

第 5 章

評価と考察

本章では設計段階のシステムを基に実装したシステムを評価することにより、サウンドコード技術を IOT セキュリティシステムに利用する利点と問題点について述べる。

設計段階のシステムを基に実装した宅配ボックスシステムを評価し、実装できていない部分を以下の表 5.1 にまとめる。

表 5.1. システムの相違点

項目番号	相違箇所	実装したシステム	設計段階のシステム
1	アプリのユーザインタフェース	Web ブラウザ	スマートフォンアプリ
2	ヒューマンエラー時の処理	無し	有り
3	開扉回数によるサウンドコードの更新	無し	有り
4	登録時のアプリ→ボックスの通信手段	PHP 通信	サウンドコード通信

サウンドコードは宅配ボックスシステムの中でアプリとボックス間のデータ通信手段として使われ、ボックスの開扉やユーザ情報の登録の役割を担っている。データ通信の際には人の耳には聞こえない音であることと、ボックスの開扉回数によりサウンドコードが更新されることから機密性は高いと言える。また、サウンドコードの作成は容易であるため、管理コストの削減を行うことができる。宅配ボックスシステムにサウンドコード技術を活用するために使用した周辺機器はスピーカーとマイクのみであることから導入コストが低いことがわかる。以上のことから、サウンドコード技術

を利用したセキュリティシステムには従来のセキュリティシステムにはないメリットがあるといえる。

次に、表 5.1 を参照してサウンドコード技術の問題点について考察を述べる。項目番号 4 に着目する。この項目は設計段階で想定していた新規登録及び再登録が、サウンドコードによる通信では実装できなかったことを示している。これは、現状のサウンドコードの通信技術では、送信者しか知りえない情報を受信者に送ることができないからである。今回のシステムでは、利用者が持つシリアルナンバーをサウンドコードに変換し音声出力したところ、宅配ボックスには不明のサウンドコードとして認識されデコードができないということである。現状のサウンドコード通信技術では、設計段階の宅配ボックスシステムを実装することは不可能であり、他の IOT セキュリティシステムに用いることも難しいのではないかと考えられる。

先の考察から、IOT セキュリティシステムにサウンドコード技術を利用する価値はあると言えるが、現状の通信方式のみではシステムに組み込むのは困難であるため、サウンドコードの新たな通信プロトコルの開発が課題となる。

第 6 章

あとがき

本論文ではサウンドコード技術を利用した宅配ボックスシステムのシステム設計および部分的な実装を行い、それらを評価することで IOT セキュリティシステムにおけるサウンドコード技術のメリット、問題点、今後の課題について考察した。宅配ボックスシステムのシステム設計には UML 図を用いており、要求定義をユースケース図、設計をクラス図およびシーケンス図にまとめた。実装は設計の機能項目から決定した高優先度の機能のみを行い、単体テスト、結合テスト、総合テストを通して動作を確認した。設計段階のシステムを基に実装できたシステムを評価することでサウンドコードの通信技術の問題を挙げ、サウンドコードの新たな通信プロトコルの開発を今後の課題とした。また、設計段階の宅配ボックスシステムを考察し、IOT セキュリティシステムにサウンドコード技術を利用することにより、機密性が高く低コストのシステムを設計できることを確認した。

謝辞

本研究を進めるにあたり、懇篤な御指導、御鞭撻を賜りました本学高橋寛教授に深く御礼申し上げます。

本論文の作成に関し、詳細なるご検討、貴重な御教示を頂きました本学樋上喜信准教授に深く御礼申し上げます。

また、審査頂いた本学岡野大准教授ならびに宇戸寿幸准教授に深く御礼申し上げます。

最後に、多大な御協力と貴重な御助言を頂いた本学工学部情報工学科情報システム工学講座高橋研究室の諸氏に厚く御礼申し上げます。

参考文献

- [1] 宅配の再配達が発生による 社会的損失の試算について - 国土交通省
<http://www.mlit.go.jp/common/001102289.pdf>

- [2] オージス総研『かんたん UML』株式会社 翔泳社 2007 年

- [3] 阪田史郎、高田広章『組込みシステム』株式会社 オーム社 2006 年