

2022 年度 卒業研究

題目

ビデオ会議を軸にした実践コミュニティ形成アプリ
ケーションの開発と有効性の検証

氏 名 熊沢 律紀

学籍番号 1J17F048

指導教員 菊池 英明

早稲田大学 人間科学部

目次

第1章 はじめに.....	2
1.1 研究背景.....	2
1.1.1 ビデオ会議.....	2
1.1.2 実践コミュニティ.....	2
1.1.3 実践コミュニティの定義 (HIP, 2018.06.29).....	3
1.2 研究目的.....	3
1.3 本研究のアプローチ.....	3
1.4 本論文の構成.....	4
第2章 システム概要.....	4
2.1 アプリケーション内容.....	4
2.2 実装.....	5
2.3 システム概要.....	5
2.3.1 Flutter.....	5
2.3.2 設計.....	5
2.3.3 開発手法.....	6
2.3.4 サーバー.....	6
2.3.5 AgoraWebRTC.....	6
2.3.6 ツリー構造.....	6
第3章 システム有効性の検証.....	8
3.1 目的.....	8
3.2 検証方法.....	8
3.2.1 事前アンケート.....	8
3.2.2 実験.....	9
3.2.3 事後アンケート.....	9
3.3 実験環境.....	9
3.4 実験における注意事項.....	9
3.5 結果.....	9
3.5.1 七段階評価.....	9
3.5.2 メタ評価.....	9
第4章 考察.....	9
第5章 結論.....	9
引用文献.....	10

図 1 実践コミュニティの構成要素.....	3
図 2 アプリ内のユーザーの動き	5

第 1 章 はじめに

1.1 研究背景

1.1.1 ビデオ会議

Covid-19 の蔓延により遠隔でのコミュニケーションが活発化している。例えば、Zoom や Microsoft Teams などのリモートアプリケーションがある。リモートワークに関してパーソナル総合研究所の新型コロナウイルスによるテレワークへの影響に関する調査によるとコロナが収束した後のリモートワーク継続希望率が 80.9%[1]で今後もリモート環境が続いていくと考えられる。[パーソナル総合研究所シンクタンク本部; , 2022/08/30]ビデオ会議を利用することで遠隔でのコミュニケーションが可能になる。ビジネスパーソンを探す、知識の交換を行うなど目的を持った遠隔でのコミュニティの形成を容易にするツールが必要だが現在、ビデオ会議を利用した遠隔コミュニティ形成支援ツールは存在しない。

1.1.2 実践コミュニティ

実践コミュニティの概念が大企業を中心に導入されている。事例として、富士ゼロックスの VHP がある[2](HIP, 2018.06.29)。この VHP は組織から独立して自主学習をしていく特徴がある。これによって社員の自主性を持って新規サービスを生み出していくことができるため、自己成長や働きがい社員に体感させることができる。他にも大企業を中心に実践コ

コミュニティを導入する企業が増えている。

1.1.3 実践コミュニティの定義 (HIP, 2018.06.29)

Harvard Business School[2] (Etienne, Wenger; Richard, McDermott; William M., Snyder; 櫻井, 祐子; 野村, 恭彦, 2002)によると実践コミュニティの構成要素は領域、コミュニティ、実践の3要素で構成されている。本研究では実践コミュニティはメンバーに共通する課題や問題を影響を与え合いながら学習する集団で知識の交換と活動を行うことを実践コミュニティの定義とする。

実践コミュニティの構成要素



図 1 実践コミュニティの構成要素

1.2 研究目的

以上の背景から、ビデオ会議を軸とした実践コミュニティ形成アプリケーションを提案する。ビデオ会議の情報伝達効率が高い特性を利用し、アプリケーション内に実践コミュニティ形成をサポートする仕組みを作ることとする。

1.3 本研究のアプローチ

本研究のアプローチは以下に順序立てて述べる。

1. アプリケーションの開発

実践コミュニティ形成アプリケーションの開発を行う。アプリケーションの内部構造はビデオ会議プログラムと SNS プログラムで組み合わせて開発を行う。

2. 評価尺度の選定

実践コミュニティの定義からユーザーが実践コミュニティ形成を行うことが出来るのか七段階評価尺度を選定する。

3. アンケートの作成

七段階評価尺度からアンケートを作成する。

4. システム有効性の検証

システム有効性の検証を行う。

1.4 本論文の構成

本論文は全 5 章で構成されている。

第 1 章では、本研究の背景、研究目的、本研究のアプローチ、本論文の構成について述べる。

第 2 章では、システム概要について述べる。

第 3 章では、システム評価について述べる。

第 4 章では、考察について述べる。

第 5 章では、結論について述べる。

第 2 章 システム概要

2.1 アプリケーション内容

全体的な構想としてユーザーにコミュニティ形成を活発化させるためのアプリ内の動きを重要視している。ユーザーは目的とするコミュニティを探すため検索やソートする。次にコミュニティ内にあるルーム一覧から目的が合致する、信頼できるユーザーが存在するルームに参加しビデオ会議を行う。ビデオ会議時間は 15 分間で終了し、次のルームを選び参加する。15 分間の短い時間を設けることでマッチング回数を増やし目的に合致するユーザーと繋がることで目標と課題の合致した実践コミュニティ形成を促進する。

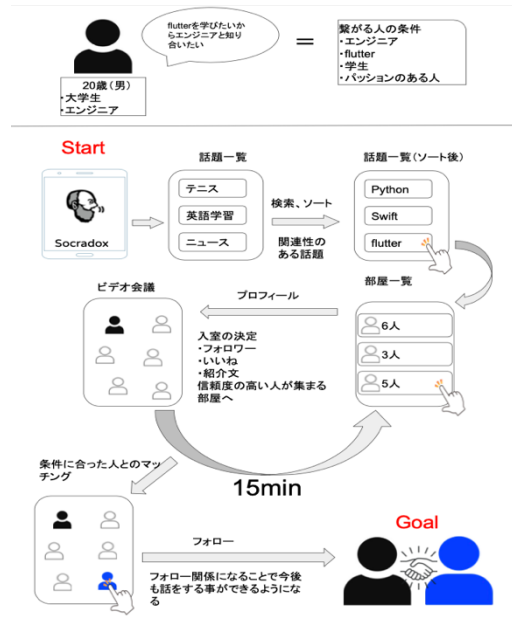


図 2 アプリ内のユーザーの動き

2.2 実装

本アプリケーションは SNS アプリケーションと WebRTC (Agora WebRTC) で構成されている。フレームワークは Flutter で、言語は Dart で構築されている。サーバーは Google サーバー (Firebase) でクロスプラットフォームに開発を行った。

2.3 システム概要

2.3.1 Flutter

マルチプラットフォーム開発用フレームワーク。iOS、Android、MAC、Windows、Linux、Web アプリケーション上で同じ挙動のアプリケーションを開発可能。現在使用可能な言語はオブジェクト指向言語の Dart である。アプリケーション開発においてプラットフォーム毎に異なる言語で記述する必要がなく、Dart 言語 1 つで開発が可能であるため本アプリケーションの開発に Flutter を利用した。

2.3.2 設計

本アプリケーションでは MVVM (Model、View、View Model) の設計思想で開発を行なった。MVVM は Model、View、View Model の略語で、Model は UI 以外の処理、データの管理、サーバーとの入出力、View は UI の処理、View Model で View と Model の変更を監視し、変更が必要である場合に通知を行う。MVVM による利点として、コードの可読性を

高めることが可能になっている。

2.3.3 開発手法

Flutter における状態管理を Riverpod+freezed で実装した。Riverpod+freezed による状態管理は Flutter でアプリケーション開発において人気の高い状態管理手法であり、MVVM モデルでシングルトンに開発を行うことが可能なため、コード変更によるエラーを事前に回避することが出来るなど安全に開発を行うことが出来る。

2.3.4 サーバー

Firebase を使用した。Firebase とは Google が提供しているクラウドサーバーを含むプラットフォームで開発を支援する機能や、高いセキュリティ性、サーバー維持など様々な利点がある。Firebase は Flutter にプラグインを提供しているため、本アプリケーションでは Firebase のクラウドサーバーでバックエンドを実装した。

2.3.5 AgoraWebRTC

本アプリケーションでは AgoraWebRTC でビデオ会議の実装を行なった。初めに、WebRTC とは Web Real Time Communication の略語で、ビデオ会議などのリアルタイム通信機能を提供する API である。AgoraWebRTC は Agora.io が開発した WebRTC で低遅延なリアルタイム通信をアプリケーションに実装することが出来る。本アプリケーションでは低遅延、可能な同時接続人数が 7 人必要であるため AgoraWebRTC を採用した。

2.3.6 ツリー構造

本アプリのツリー構造を以下に記述する。以下のツリー構造からアプリケーションのプログラム全体の構造と機能を把握する。

- └── config 設定ファイル
- └── controllers アプリ全体における状態管理
 - └── schedule スケジュール関連
 - └── theme_change ホワイモード、ダークモードの実装
- └── data データ
 - └── model モデル
 - └── comment システムエラー等の通知
 - └── db データベースとの接続

- | | └── topic 話題モデル
- | | └── user ユーザーモデル
- | └── repository レポジトリ
- └── generated ジェネレートディレクトリ
- | └── intl 多言語変換支援
- └── helpers 検索バー等ウィジェット格納
- └── l10n 多言語変換用ディレクトリ
- └── route_widgets ページ遷移時のウィジェット
- └── routes ページ遷移設定
- | └── nav_controller ナビゲーションバー状態管理
- | └── nav_widget ナビゲーションバー
- └── ui UIディレクトリ
- | └── add_topic 話題作成ディレクトリ
- | | └── add_topic_state 話題作成状態
- | | └── add_topic_state_controller 話題作成状態管理
- | | └── add_topioc_view 話題作成ページ
- | | └── components 構成
- | | └── screens 話題作成スクリーン
- | └── agora AgoraWebRTCディレクトリ
- | | └── controllers 状態管理
- | | └── models モデル
- | | └── src ソース
- | | └── buttons ボタン
- | | └── layout レイアウト
- | | └── widgets ウィジェット一覧
- └── common 画面構成
- | | └── components 画面構成ファイル格納
- | | └── widget 画面構成ウィジェット格納
- └── community コミュニティ画面ディレクトリ
- | | └── community_view コミュニティ画面
- | | └── community_controller コミュニティ状態管理
- | | └── community_state コミュニティ状態
- └── event_room イベントディレクトリ
- | | └── event_room_controller イベント状態管理
- | | └── event_room_state イベント状態
- | | └── event_room_view イベント画面
- └── hooks hooks パッケージによる状態管理

- | |—— login ログインディレクトリ
- | | |—— auth_state 認証状態
- | | |—— auth_state_controll 認証状態管理
- | | |—— auth_view 認証画面
- | |—— profile プロフィールディレクトリ
- | | |—— profile_state プロフィール状態
- | | |—— profile_state_controller プロフィール状態管理
- | | |—— profile_view プロフィール画面
- | | |—— sub プロフィール画面ウィジェット
- | | |—— who_cares_me フォローフォロワーディレクトリ
- | | |—— who_cares_me_state フォローフォロワー状態
- | | |—— who_cares_me_state_controller フォローフォロワー状態管理
- | | |—— who_cares_me_view フォローフォロワー画面
- | |—— room_list_view ルーム一覧画面
- | |—— widgets ウィジェット
- | |—— start 起動画面
- | |—— topic トピックディレクトリ
 - | |—— components トピックディレクトリ構成ウィジェット
 - | |—— topic_state トピック状態
 - | |—— topic_state_controll トピック状態管理
 - | |—— topic_view トピック画面
- |—— widgets ウィジェット一覧

第3章 システム有効性の検証

3.1 目的

実験コミュニティの構成要素である領域、コミュニティ、実践から提案システムが実践コミュニティ形成に効果的であるか検証を行う。

3.2 検証方法

3.2.1 事前アンケート

被験者の目的や課題を調査する。調査結果から目的や課題のジャンルに近い4人をグループ化する。(グループ人数は被験者数から判断する。例：20人の場合5×4、16人の場合4×4、全グループの人数を同数に調整する)

記述形式のアンケートを実施する

アンケート内容：現在抱えている自身の課題や目標を記述してください

3.2.2 実験

被験者に本アプリケーションを利用したワークショップに参加してもらう。

3.2.3 事後アンケート

本アプリケーション内での活動を通して実践コミュニティが形成されたのかを評価する。
評価方法は七段階評価で実践コミュニティ尺度と併せてアンケートに回答してもらう。

3.3 実験環境

対面での実施。

3.4 実験における注意事項

3.5 結果

3.5.1 七段階評価

(実験後に記述予定、事後アンケート結果を提示する)

3.5.2 メタ評価

(実験後に記述予定、サーバー情報からの分析結果を提示する)

第4章 考察

(実験後に記述予定)

第5章 結論

(システム有効性の検証から研究目的を達成できたのか簡潔にまとめる。)

引用文献

- Etienne, Wenger; Richard, McDermott; William M., Snyder; 櫻井, 祐子; 野村, 恭彦.
(2002). コミュニティ・オブ・プラクティス. 翔泳社.
- HIP. (2018.06.29). 富士ゼロックスで 20 年続くイノベーション文化。キーマンが語る秘訣
とは? https://hiptokyo.jp/hiptalk/fujixerox_virtualhollywoodplatform/#.
- パーソナル総合研究所シンクタンク本部;. (2022/08/30). 第七回・新型コロナウイルス対策
によるテレワークへの影響に関する緊急調査（速報版）. <https://rc.persol-group.co.jp/thinktank/data/telework-survey7.html>.