Vol.2017-MBL-82 No.9
Vol.2017-UBI-53 No.9
2017/3/9

コンタクトセンター業務効率化のための WebRTC を用いた多元業務集約管理方式

志賀勇介[†1]　高見一正[†1]

概要：近年，WEB ブラウザ間の P2P 通信技術として WebRTC が注目されており，様々な分野で活用されはじめている．また，インターネット技術が発展する現在において，企業と顧客間で主なコミュニケーション形態となっている電話や E メールから，ビデオ通話や WEB チャットなど，手軽さや利便性を求めた多様化が期待されている．本稿では，コンタクトセンター業務で WebRTC を活用し，電話機・その他業務端末など多岐に渡るツールを一括集約する事で，業務効率の向上を目的としたコンタクトセンターアプリケーションの多元業務集約管理方式を提案する．

キーワード：コンタクトセンター，業務集約管理，WebRTC，P2P 通信，ブラウザ，WEB アプリケーション

A Multi-task Aggregation Management Method using WebRTC for Improving Contact Center Efficiency

YUSUKE SHIGA[†1]　KAZUMASA TAKAMI[†1]

Abstract: In recent years, WebRTC has attracted attention as P2P communication technology between WEB browsers, and it is beginning to be utilized in various fields. Also, as Internet technology evolved, it is expected that diversification is sought for convenience such as video call and web chat from telephone and e-mail, which are the main forms of communication between companies and customers. In this paper, we propose a contact center's multi - task aggregation management method aimed at improving operational efficiency by collectively collecting a wide range of tools such as phones and other business terminals by using WebRTC in contact center business.

Keywords: Contact Center, Task Aggregation Management, WebRTC, P2P Connection, Browser, WEB Application

1. はじめに

近年，WEB ブラウザ間の P2P（Pear to Pear）コネクションを実現する技術として WebRTC（Web Real-Time Communication）が注目されている[1]．Google によりオープンソース化されている．また，World Wide Web Consortium (W3C)によりブラウザ対応 API が[2]，IETF（Internet Engineering Task Force）により関連の通信プロトコルが[3]，それぞれ標準化されている．これにより，専用アプリ等を用いてサーバ仲介によって実現していた複数端末間のビデオチャットを，WEB ブラウザ間の P2P 接続のみで実現する事が可能となり，産業界の様々な活用が期待されている．

一方，既存のコンタクトセンターでは，設置した電話機に加え，PC 等の業務用端末を併用した業務形態が一般的である．更にその業務用端末においても，メールやロギングなど，その内容によって使用するツールが多岐に渡り集約による業務効率化が求められている．

本稿では，WebRTC を活用したコンタクトセンター向け業務アプリケーションを提案する．WebRTC を用いて電話業務を含めたこれらの業務を一元集約することにより，業務効率化を達成する．第 2 章ではコンタクトセンター事業形態と関連研究について説明する．第 3 章で提案システムとそれを実現するための課題を示し，第 4 章で課題解決の提案方式を示す．第 5 章で提案アプリケーションの試作システムを説明する．第 6 章で評価方法を示し，第 7 章で評価結果をまとめる．第 8 章で本稿のまとめと今後の課題を記す．

2. コンタクトセンター事業形態と関連研究

本章では，昨今のコンタクトセンター事業の動向について触れ，本稿で試作するアプリケーションの核となる，WebRTC が持つ技術について示す．また，WebRTC を用いた高齢者のための支援技術について示す．

2.1 昨今のコンタクトセンター事業形態

近年のインターネット技術の進化に伴い，消費者間のコミュニケーション形態は変化してきているが，消費者と企業のコミュニケーションの形態に大きな変化は見られず，現在も従来からの「電話（86.1%）」「Web サイト（79.8%）」「E メール（51.0%）」が中心となっている．その中で，スマートフォンや SNS などの普及によりデジタルコミュニケーションが主流になってきた若い世代の消費者を中心に，「WEB チャット（期待度 36.9%）」「テレビ電話（期待度

†1 創価大学　工学部　情報システム工学科
Soka University

43.1%)」のような，企業とのコミュニケーション形態のデジタル化を求める声が多く見られる．一方で，従来の「電話」「Web サイト」「E メール」についても，期待度は下がるとはいえ一定以上の利用意向があることから，企業にはサービス形態の多様化が求められている[4].

2.2 WebRTC とコンタクトセンター事業への活用と課題

本稿でメインの技術である WebRTC とは，Web ブラウザにプラグインを追加することなく，Web ブラウザ上で UDP/IP を使用した，インタラクティブなリアルタイムコミュニケーションを可能にするオープンフレームワークである[5]．対応しているブラウザは Chrome，Firefox，Opera の 3 つであるが，iOS 上の Chrome では動作しないなど，OS によっては該当のブラウザでも動作しないことがある．この技術を用いて，ブラウザ上にて，端末のカメラ／マイクから取得したストリームデータとテキストデータやバイナリデータの P2P 通信が実現できる．

WebRTC を利用すれば，約十数行程度の JavaScript/HTML の記述と，SSL 対応の Web サーバを構築するだけで，誰でもブラウザ上で動く Web チャットを作成することができる．しかしコンタクトセンターでブラウザとして活用するためには，以下の課題がある．

(1) オペレーションログ記録へのアプローチ

業務利用をする上で，最低限のログ記録が必要となる．ログの用途は，主に顧客とのトラブル時に役立つエビデンスや，サービス品質向上に繋がるデータマイニングのリソースなどである．WebRTC はサーバを経由せず P2P 通信でデータの送受信を行うため，送受信されるデータは機密性が高く，意図して設計しない限りクライアント端末以外で保存されることはない．ログを記録する為には，P2P 通信でやり取りされるデータからログとして適切なもののみをピックアップし，DB サーバに保存する処理を実装する必要がある．

(2) 電話機能の再現

コンタクトセンターとして電話業務を行う上で，通話転送，保留，特定の顧客への発信（ダイヤル）を行うための UI 設計などの機能は不可欠である．これらはアプリケーション側で設計する必要がある．

(3) 問い合わせ回線の再現

コンタクトセンター・コールセンターを設ける際，PBX などを利用し，ひとつの電話番号を複数の電話機で共有する回線を作る．これを再現した設計を，アプリケーション側で組み込む必要がある．

2.3 WebRTC を活用した EC 弱者支援

WebRTC を活用した既存研究には，インターネット利用に不慣れな高齢者などの EC(Electronic Commerce)弱者に対して，遠隔地から EC 利用の支援を行うブラウザ連携システムとして活用する提案がある [6]．この論文で提案されているシステムでは，WebRTC による P2P 通信を通して URL やスクリーンショットなどの情報を送受信し，EC 弱者と支援者間で Web ブラウザの表示状態を同期させる．このブラウザ連携システムは, JavaScript 等によりページ内で動的に内容が細かく書きかわってしまうような Web サイトに対しても，クリックイベントの同期を行うことで対応させている．また，カメラで撮影した映像や，マイクで集音した音声を P2P 通信で送受信する機能も備えており，音声/映像による通信相手の認識や細かい支援を行うことが可能である．さらに HTML5 Canvas を利用し，EC 弱者がブラウザ上のどのポイントに注目すればよいかを明示する為のイメージを描画し，その描画位置やイメージについても EC 弱者のブラウザと同期することができる事から，支援の効果を高めている．

この既存研究と本稿との違いは，EC サイトを表示するブラウザ上での操作や，ページ内での決済手続きの方法について，EC 弱者に細かく支援ができるよう特化させた，ブラウザ状態そのものを WebRTC で連携させる形式のシステムが既存研究であるのに対して，WebRTC を用いた音声/映像通信による擬似的な電話機能を基本とし，さらにコンタクトセンター業務効率化に特化すべく，その他の様々な必要機能一つのアプリケーションに集約するシステムが本稿の提案である．このように，WebRTC を通信ツールとしてどのように利用するか，そのアイデアによって，様々な用途に特化したシステムを生み出す事が可能である．

3. WebRTC を用いた多元業務集約管理方式

本稿では，WebRTC を用いたコンタクトセンター業務アプリケーションを提案し，その機能集約によって業務効率の向上に繋げる事を目的とする．具体的には，図 1 のように，本来は電話機を用いて行われる「音声通話」，メールやクラウド等を用いて行われる「文章データのやり取り」や「ファイル共有」, そして各センターで専用の業務用ツールを用いて行われる事が想定される「対応履歴のロギング」など，コンタクトセンターサービスの展開に必要な機能一式を，ひとつのアプリケーションに集約する事で業務効率化を図ることである．これを達成する為，以下の課題を解決する必要がある．

(1) 業務アプリケーション機能要件の明確化

コンタクトセンター業務に必要な機能を選定し，試作アプリケーションのユースケースと機能要件を明確化する．

(2) WebRTC に基づくアプリケーションの試作

WebRTC を実装するために使用する API や，開発言語，開発環境を選定し，機能要件に沿ったコンタクトセンターアプリケーションの試作を行う．

(3) 実用性の向上

業務用アプリケーションとして開発する上で，その実用性が基準以上である必要がある．試作アプリケーションの実用性を高めるため,「ユーザビリティ」の観点で，設計上

Vol.2017-MBL-82 No.9
Vol.2017-UBI-53 No.9
2017/3/9

の工夫を施す．

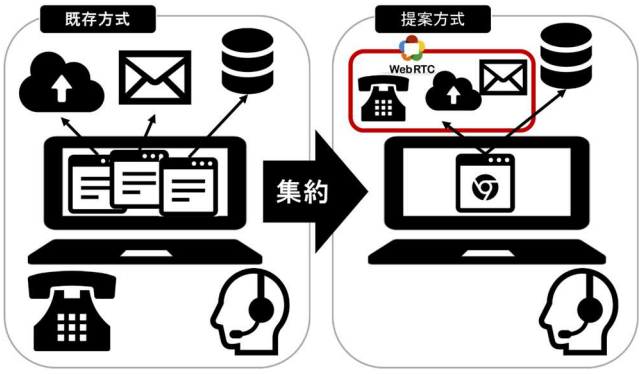

図 1　「WebRTC」を活用した機能集約のイメージ

Figure 1 Image of multi-task aggregation using “WebRTC”.

4. 提案方式

4.1 業務フローの策定

業務アプリケーションは本来電話機を用いて行われる音声通話応対業務を含める為，アプリケーションの中で，問い合わせ回線への入電を再現する必要がある．カスタマが問い合わせる際にアクセスするページと，オペレータが対応業務を行う際にアクセスするページの 2 つを個別に用意することで，これを実現する．

本アプリケーションを利用した場合の基本的な業務フローを次のように策定し，図 2 のアクティビティ図で表す．

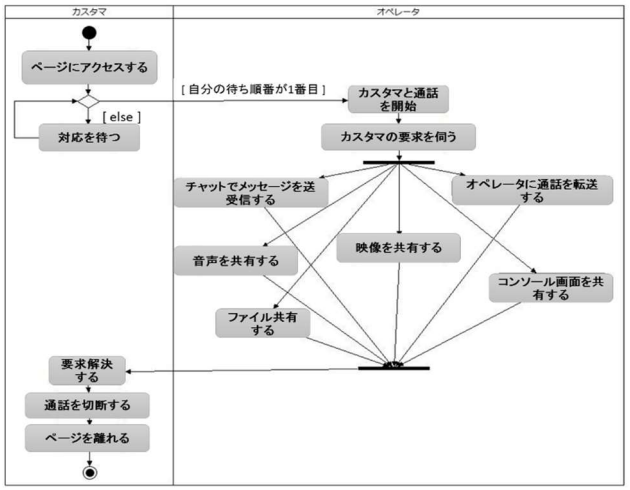

図 2　業務フローのアクティビティ図

Figure 2 UML activity diagram of the workflow.

まずカスタマが，問い合わせ用の URL にアクセスし，オペレータからの通信を待つ．オペレータは専用の URL にすでにアクセスしており，アクセス順に並んだカスタマのリストから，先頭のカスタマへ通信を開始し，アプリケーションの機能を 1 つ以上利用してサポートを実施する．要求が満たされ，カスタマが通話を切断しページを離れる．ここでカスタマ 1 人に対しての対応フローが終了となる．

また，これら要求解決に必要となる「文字/ファイル形式データの送受信」「画面共有」など，既存方式であれば複数のツールを必要とする工程を含め，この業務フロー全ての行程を一つの提案アプリケーションで実行可能であるということが，既存方式の業務フローとの大きな違いである．

4.2 機能要件とユースケース

業務アプリケーションの機能を決定する．コンタクトセンター業務に必要な機能を以下に挙げる．

(1)　コンタクトセンター業務で必要な基本機能

- 音声のリアルタイム共有（≒電話）
- ファイル共有（≒メールや FTP を使った共有）
- チャット（≒メール）

(2)　あると便利なオプション機能

- 映像共有（迅速かつ正確な情報伝達が可能）
- 画面共有（具体的操作方法の視覚的援助）
- 回線混雑状況の可視化（カスタマのストレス減少）

以上の機能を備えた試作アプリケーションのユースケースを図 3 に示す．

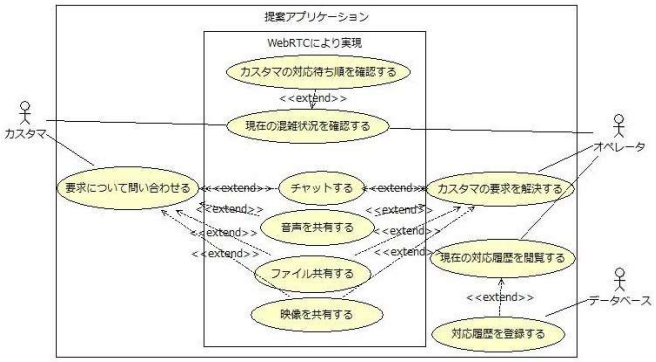

図 3　提案アプリケーションのユースケース図

Figure 3 UML use case diagram of the suggesting application.

5. 提案アプリケーションの試作

5.1 概要

図 4 に試作のファイル構成，表 1 に概要をまとめた．

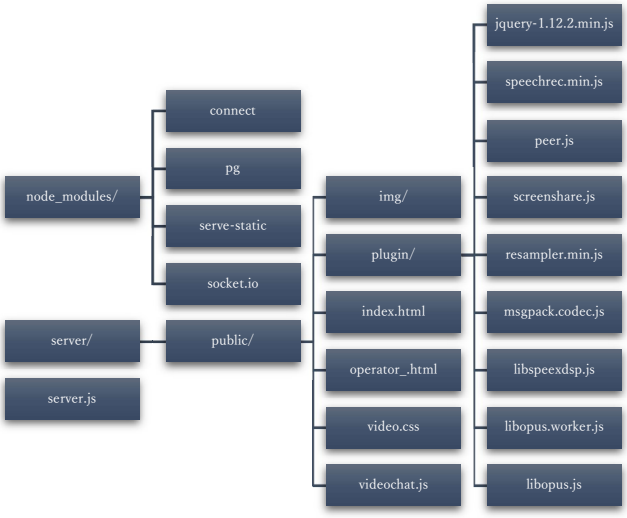

図 4　試作アプリケーションのディレクトリ構成

Figure 4 Directory structure of prototype application.

表 1　試作アプリケーション概要

Table 1 Overview of the application.

アプリケーション名	Web Chat For Enterprise			
動作環境	Windows / Google Chrome version50 以上			
アプリケーションの形式	Node.js アプリケーション			
記述したプログラムファイル	静的リソース	index.html	56 行	
		operator_.html	62 行	
		video.css	821 行	
		videochat.js	2965 行	
	Node.js アプリ実行ファイル	server.js	153 行	
サードパーティプラグインファイル	jQuery		jquery-1.12.2.min.js	
	音声認識 API ※1	リサンプラ	libspeexdsp.js	
			resampler.min.js	
		MessagePack 用ライブラリ	msgpack.codec.js	
		Opus エンコーダ	libopus.js	
		Opus ワーカ (WebWorkers で動かすワーカスレッド)	libopus.worker.js	
		音声認識クライアントライブラリの本体	speechrec.min.js	
	画面共有 API※2		screenshare.js	
			screenshare.min.js	
	Simple peer-to-peer with WebRTC※3		peer.js	

※SkyWay [7] 提供

※1：https://github.com/nttcom/SkyWay-SpeechRec

※2：https://github.com/nttcom/SkyWay-ScreenShare

※3：https://github.com/nttcom/peerjs

HTML ファイル「index.html」はカスタマ向けのページ，「operator_.html」はオペレータ向けのページである．アプリケーションの動作処理は全て「videochat.js」の 1 ファイルに記述しており，スタイルシートも「video.css」の 1 ファイルにまとめている．「index.html」も「operator_.html」もどちらも同じスクリプトファイルとスタイルシートファイルを読み込む形となる．「videochat.js」は，アクセスしたクライアントがオペレータ向けページにアクセスしているか，カスタマ向けページにアクセスしているかを判別し，それぞれに適した動作を行うように設計している．

また，表 1 に示すように SkyWay [7]によって提供されるプラグインファイルを複数導入している．SkyWay は，NTT コミュニケーションズによって提供される WebRTC プラットフォームである．SkyWay では，WebRTC を用いる P2P のストリームデータ通信／テキスト・バイナリ通信を実装し易く設計した，PeerJS API という SDK を提供しているだけでなく，拡張機能として TURN，音声認識，多人数接続，画面共有機能など，それぞれ API を用意し，容易に実装できるようにサポートしている．

5.2 システム構成

試作アプリケーションのシステム構成を図 5 に記す．Web サーバは Cloud Application Platform である heroku[8]に構築した．server.js は Node.js で記述したアプリケーション実行ファイルで，「connect」や「http」モジュールを用いてページリソース（HTML/JS/CSS）の管理と簡易的な HTTP サーバの運用，「pg」(postgreSQL)モジュールを用いてデータベースとのアクセスを行う．

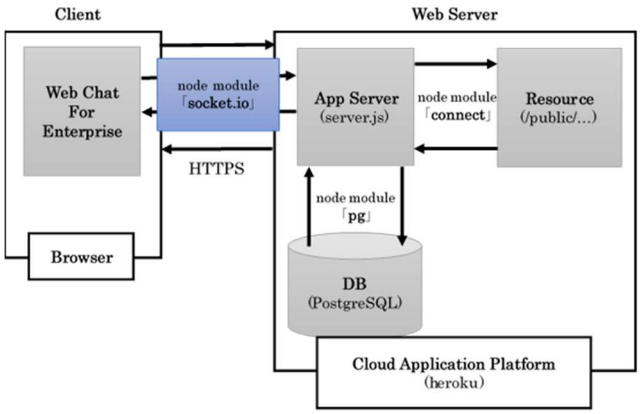

図 5　システム構成

Figure 5 System configuration.

5.3 主な実装機能

試作アプリケーションに実装した主な機能を表 2 に示す．「デスクトップ画面共有」機能の利用については，試作専用に別途作成した Manifest ファイルを含む，Google Chrome Extension ファイルをユーザが予めインストールしておく必要がある．

表 2　試作アプリケーションの主な実装機能

Table 2 Main implementation function of prototype application.

ビデオ通話	マイク・カメラを使用したリアルタイムの音声/映像共有
他オペレータの通話招待	他オペレータ 1 人・カスタマ 1 人を同時に相手にしたビデオ通話
通話転送	他オペレータへのカスタマ通話転送
回線混雑状況の確認	入室中のカスタマ ID･名前の一覧表示と管理
メッセージの送受信	テキスト文章を入力・送信・受信
ファイル共有	アカウント毎に設けられたアップロード領域の共有
デスクトップ画面共有	任意のデスクトップ画面・Window 画面を送信
通話対応のログ自動保存	対応内容を残したアクティビティログ・メッセージのやり取りの履歴を DB へ自動保存

6. 評価手法

試作アプリケーションの総合的な評価目標は「コンタクトセンター業務のための，実用性の高い多機能集約アプリケーション」である．よって，機能集約による業務効率と

ユーザビリティの 2 面で評価を行い，また，別途画面共有機能の有用性の評価も行った．

6.1 業務効率の評価テスト

シナリオ 1「研究室への入会対応」

カスタマは研究室に入会したいと考えているが，いくつかの質問と，手続きの仕方についてわからない．これらを解決するため，窓口に問い合わせる．問い合わせると，オペレータが応対し，電話口で質問事項は解決する．手続きについて確認すると，電子書類を送付され必要事項の入力を求められる．オペレータの指示に従い，その場で入力した書類を返送し，手続きを完了する．

目的

一般的なコールセンター形態（パターン A）と，試作アプリを使った形態(パターン B)で業務効率を比較する．

パターン A

電話機（携帯電話），PC（G メール）を使用する．シナリオに沿ったツール使用と業務の流れを図 6 に示す．

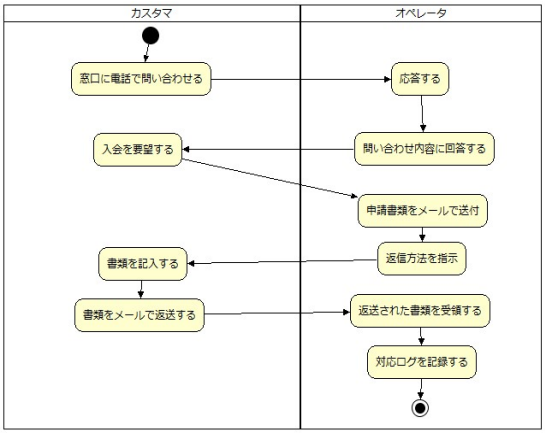

図 6　パターン A のシナリオフロー

Figure 6 Scenario flow of pattern A.

パターン B

試作アプリを使用する．シナリオに沿ったツール使用と業務の流れを図 7 に示す．

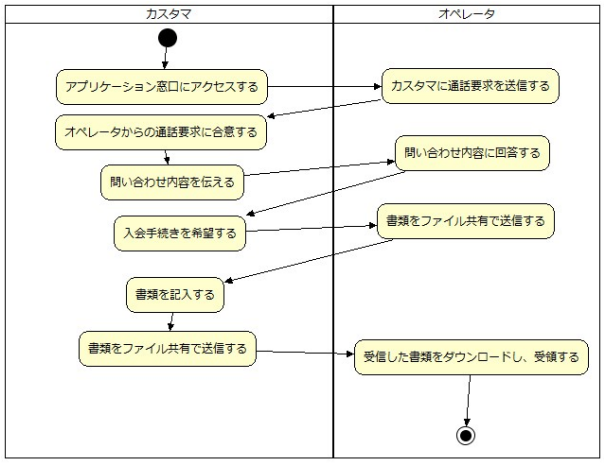

図 7　パターン B のシナリオフロー

Figure 7 Scenario flow of pattern B.

手法

表 3 の被験者に，それぞれのパターンでシナリオを遂行させる．それぞれの達成にかかった時間を図り，その差異から業務効率の変化を確認する．また被験者にはそれぞれ，テスト後にアンケート調査を行い，ユーザビリティの観点でも評価を行う．

表 3　テストシナリオ 1 の被験者の特性

Table 3 Characteristics of subjects in test scenario 1.

	年齢	性別	PC 使用頻度	PC 環境（OS / Memory / CPU）
カスタマ	22	男性	日常生活でよく使う	Windows8.1 / 8GB / Intel Core i5
オペレータ	22	男性	日常生活でよく使う	Windows7 / 8GB / Intel Core i3

6.2 ユーザビリティ評価テスト

シナリオ 2「サーバ保守のコンタクトセンター業務」

カスタマがベンダーから購入したサーバの LED 点灯状況がいつもと違う事に気づき，どうしたらいいか保守窓口に問い合わせる．保守窓口である URL（Web Chat For Enterprise）にアクセスし，まず問い合わせ概要を確認する受付オペレータ A との通話が開始. LED 点灯に異常があることを伝える．オペレータ A は，技術的内容と判断し，通話を保留にする. オペレータ B と別途通話を行い，問い合わせの経緯について簡単に説明し，継続の対応を依頼．オペレータ A はカスタマとの通話をオペレータ B に転送する．転送が無事完了したことを確認したら，オペレータ A はオペレータ B との通話を終了する. 通話が開始されたオペレータ B は点灯状況について映像媒体で詳しい情報を要求する．カスタマはカメラを使って LED 状況（本テストでは紙媒体に印刷された画像）を映像で送信する．オペレータ B は LED の点灯位置から，該当する障害内容を伝え，必要な処置を案内する．以上を図 8 にアクティビティ図として示す．

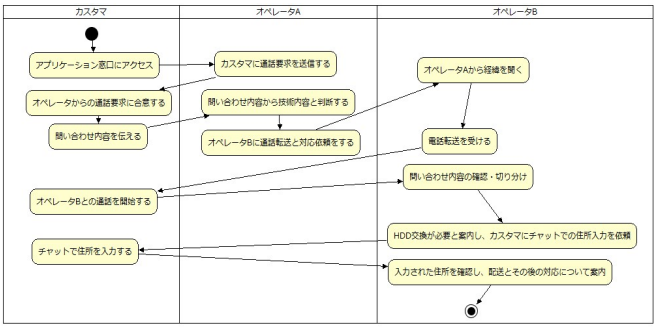

図 8　テストシナリオ 2 のフロー

Figure 8 Flow of test scenario 2.

目的

アプリケーションのユーザビリティを測る．

手法

多機能を利用するテストシナリオ 2 を表 4 の被験者 3 名で行い，それぞれにユーザビリティの観点でアンケート

Vol.2017-MBL-82 No.9
Vol.2017-UBI-53 No.9
2017/3/9

調査を行う．

表 4　テストシナリオ 2 の被験者の特性

Table 4 Characteristics of subjects in test scenario 2.

	年齢	性別	PC 使用頻度	PC 環境（OS / Memory / CPU）
カスタマ	23	男性	日常生活でよく使う	Windows8 / 5GB / Intel Core i5
オペレータ A	23	男性	日常生活でよく使う	Windows7 / 3GB / Intel Pentium
オペレータ B	22	男性	日常生活でよく使う	Windows8 / 5GB / Intel Core i5

6.3 画面共有機能の有用性評価テスト

シナリオ 3「画面共有機能に必要な設定操作の支援」

画面共有機能を利用するために必要な準備操作（Google Chrome Extension のインストール）を，試作アプリケーションを使った通話で支援する．

目的

アプリケーションの画面共有機能の有用性を評価する．

手法

1 つのシナリオを次の 2 つのパターンで，それぞれ操作支援を行い，被験者が画面共有機能を利用可能になるまでにかかった時間と，その行程の効率を比較する．表 5 のように，パターン毎にそれぞれ別の被験者を用意した理由は，いずれか一方で一度支援した被験者は操作方法を理解するため，別のパターンで同じ操作を支援しても有効な評価が得られないからである．

パターン A)　口頭のみで操作支援を受ける

パターン B)　画面共有を利用して操作支援を受ける

表 5　テストシナリオ 3 の被験者の特性

Table 5 Characteristics of subjects in test scenario 3.

	年齢	性別	PC 使用頻度	PC 環境（OS / Memory / CPU）
パターン A	28	男性	コンタクトセンター業務に勤め，仕事で常に使用している	Windows7 / 4GB / Intel Core i5
パターン B	21	男性	日常生活でよく使う	Windows7 / 4GB / Intel Pentium

7. 評価結果

7.1 試作アプリケーションの機能集約率

パターン A（電話機＋PC の従来のコンタクトセンター業務形態）とパターン B（試作アプリケーションを利用した提案の業務形態）を比較したテストシナリオ 1 から得られた，使用ツール数・所要時間の比較を図 9，図 10 のグラフにまとめた．

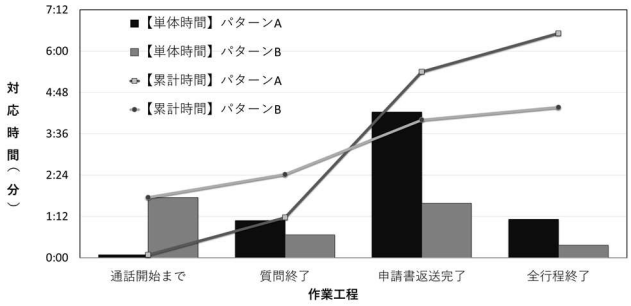

図 9　パターン A・パターン B に対する作業行程と対応所要時間の関係

Figure 9 Relationship between work process and correspondence time for pattern A and pattern B.

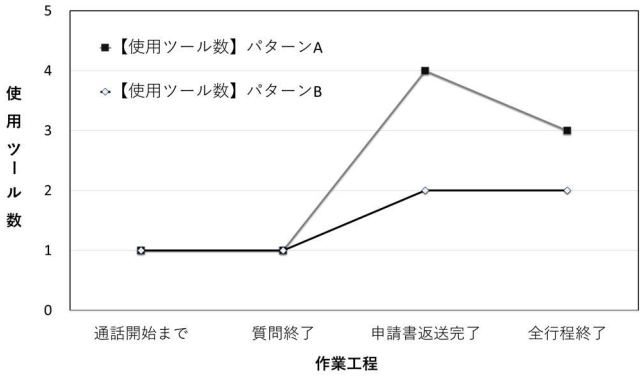

図 10　パターン A・パターン B に対する使用ツール数の関係

Figure 10 Relationship between the number of tools used and the work process for patterns A and B.

パターン B では，WebRTC による P2P 通信確立の工程にかかる時間や，試作アプリケーションの操作に対する不慣れ等により，通話開始までの工程で時間がかかってしまっているが，電話機は誰もが迷うことなく操作できることから，パターン A では 5 秒で通話開始の工程を終えた．この時間浪費をもってしても，パターン B の方が後半の工程で大幅な時間短縮を実現し，全体で 43%短縮できている．また「申請書返送完了」工程において，パターン A では，メール，ロギングツール，電話機，申請書類 Excel ファイルの計 4 ツールの併用が発生するが，パターン B ではそれらが試作アプリケーションで実現できるため，ツール数は試作と申請書類 Excel ファイルの 2 つに抑えられ，これ以上になることはない．このことから，試作アプリケーションは作業効率の向上を十分もたらすと言える．

7.2 ユーザビリティ

シナリオ 1，シナリオ 2 の被験者計 5 名にユーザビリティの数値化方式である SUS(System Usability Scale)算出[9]の為のアンケート調査を行い，平均点数 72 という結果を残した．SUS の平均値と言われる 68 点を上回り，十分なユーザビリティを備えていると言える．表 6 に，それぞれの設問に対する素点と SUS 算出結果を示す．

Vol.2017-MBL-82 No.9
Vol.2017-UBI-53 No.9
2017/3/9

表 6　SUS 算出結果

Table 6 Result of questionnaire for SUS score.

質問	シナリオ 1 オペレータ	シナリオ 1 オペレータ	シナリオ 2 オペレータ A	シナリオ 2 オペレータ B	シナリオ 2 カスタマ
①	5	3	4	4	4
②	4	2	2	3	3
③	3	4	5	3	4
④	2	2	1	2	3
⑤	4	4	5	5	4
⑥	2	2	1	2	2
⑦	4	5	4	4	5
⑧	2	2	1	3	1
⑨	2	3	5	3	3
⑩	4	3	2	3	1
SUS	60	70	90	65	75
平均	72				

また，次の通り，Jakob Nielsen により定義されたユーザビリティ項目[10]に基づき，表 7～表 11 にアンケートの各質問内容に対する回答結果の一部を取り上げる．

(1) **学習しやすさ（Learnability）**

質問内容：このアプリケーションは，マニュアルを見なくても操作できそうでしたか？操作できそう，または操作できたという場合，特にどの操作でそう思いましたか？

表 7　学習しやすさについてのアンケート回答

Table 7 Answers for questionnaire about learnability.

シナリオ 2 オペレータ A	シナリオ 2 オペレータ B	シナリオ 2 カスタマ
アイコンが強調表示されるのでオペレータの説明があれば問題なく操作できそう.	受けるだけだったので，大体の操作はマニュアルを見なくても操作できました.（例えば，電話に応答するときなど）	マニュアルがなくても大丈夫だと思いました.

(2) **効率性（Efficiency）**

質問内容：試作アプリケーションを使用して行った業務と，電話機＋PC の従来のコールセンター形式で行った業務と，どちらが効率的であると感じましたか？

表 8　効率性についてのアンケート回答

Table 8 Answers for questionnaire about efficiency.

シナリオ 1 オペレータ	シナリオ 1 オペレータ
試作アプリケーション 理由：業務中のアクションが少なくて済む為，ミスが減り，結果として時間的効率も動作的効率も良いと感じました.	試作アプリケーション 理由：PC のみで完結する.

また，別の質問で，以下のように機能面において明確な指摘事項を聴取することもできた．

- 初めの入力が面倒. オートコンプリートで一度入力したものは保存されていると良い.
- ファイルをドラッグ＆ドロップでアップロードできない.
- ログが画面の端とかに出ていると通話の状況がわかりやすいかなと思います.
- 「Calling Menu」「Other Menu」はもっとわかりやすくしたほうがいいと思います. 下のスペースが空いているので，文字で「通話終了」とか書いてもいいと思います.
- 応答状況とアイコン関係を良くすると良いと思います.

(3) **記憶しやすさ（Memorability）**

質問内容：このアプリケーションを一度使ってみれば，次からは迷うことなく操作できそうですか？

表 9　記憶しやすさについてのアンケート回答

Table 9 Answers for questionnaire about memorability.

シナリオ 2 オペレータ A	シナリオ 2 オペレータ B	シナリオ 2 カスタマ
できそうです.	他の機能を使わなかったので，一度使っても次の時もマニュアルが必要な気がします.	一度使えば，どこのボタンでどこの操作ができるかつかめるので，大丈夫だと思います.

(4) **エラー（Errors）**

質問内容：アプリケーションを操作する中で，エラーや問題などが発生しましたか？発生した場合，どのようなエラーや問題でしたか？

表 10　エラーについてのアンケート回答

Table 10 Answers for questionnaire about errors.

シナリオ 2 オペレータ A	シナリオ 2 オペレータ B	シナリオ 2 カスタマ
応答が切れた．というか，相手が応答しないことが多かった.	途中，音声が聞こえなくなりました.	特にエラー等はありませんでした.

(5) **主観的満足度（Satisfaction）**

質問内容①：このアプリケーションを操作する中で，楽しい，面白いと思える部分はありましたか？ある場合，具体的にどの場面でしたか？

質問内容②：アプリケーションを操作する中で，不快感を覚えるような場面はありましたか？あった場合，具体的にどの場面で不快感を覚えましたか？

表 11　主観的満足度についてのアンケート回答

Table 11 Answers for questionnaire about satisfaction.

質問	シナリオ 2 オペレータ A	シナリオ 2 オペレータ B	シナリオ 2 カスタマ
①	他のオペレータを通話に参加させた後，自分が退室しても，通話の続いていること	あまり機能を使うことがなかったので，面白いと思える部分は特にありませんでした.	ボタンや機能が多いので，いろいろ使ってみたいと思いました.
②	ありません	特にありませんでした.	オペレータにつなぐ前で，電話がかかっているのか，画面が止まっているのか少し不安になりました.

また，満足した点として，「操作が直感でわかる.」「サイ

トの動作は軽い.」などの回答もあった.

7.3 画面共有機能有用性テストの結果

パターン A（口頭説明のみでの操作支援）と，パターン B（画面共有を利用した操作支援）での，目的達成になるまでにかかった対応累計時間の関係を図 11 に示した.

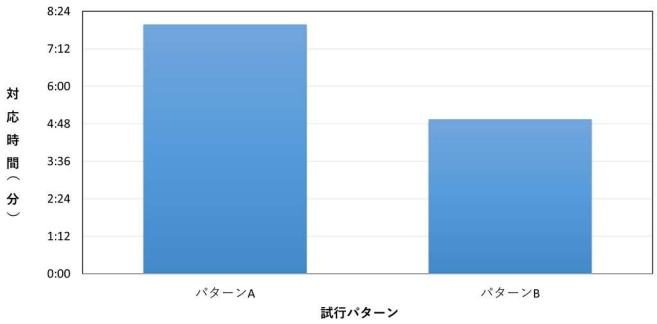

図 11　試行パターンに対する対応時間の関係

Figure 11 Relationship of response time to trial pattern.

累計 7 分 59 秒の時間を要したパターン A に比べ，パターン B では累計時間 4 分 57 秒という結果となり，約 38%の時間短縮ができた．この結果より，画面共有機能の充分な有用性を確認できた.

また，テスト中に効率性を感じた点を以下にまとめた.

- 画面遷移の様子などを明示的に確認させることが可能なため，通話相手の理解が早い
- 共有中にポインタを用いて必要なポイントへ注目を集めることが可能である
- 通話相手が画面共有の動作を追いながら操作を行うことができるため，安心感を得られる

以上の点から，時間短縮を実現でき，顧客満足度の向上も期待できると考えられる．しかし，以下のような問題点も想定される.

- ディスプレイに映る必要以上の情報を相手に与えてしまう
- OS 等，相手の動作環境の相違により，操作方法や表示等に違いが生じた場合，画面情報を共有しても支援の役に立たない可能性がある

これらの問題点に対する解決策として，画面共有範囲限定の徹底や，様々な環境のユーザへのサポートを可能にするため，支援側が動作環境をあらかじめ複数用意する必要がある.

8. まとめ

本稿では，WebRTC を用い，コンタクトセンター業務向けのアプリケーションを試作することで，WebRTC のコンタクトセンター業務への活用を提案した．そして，十分な効率化が図れることを確認した.

WebRTC のリアルタイムビデオ/音声/データのコネクションを利用することで，電話機を使った業務を，アプリケーションを用いた業務の中に包含した．これにより，電話機と PC を併用して行う業務を単純化でき，電話機を設置せず，Web ブラウザのみを使ってコンタクトセンターを設けることができることを確認した. さらに「ファイル共有」や「文字チャット」「画面共有」などの機能を含めたことにより，より様々な種類の業務へのアプローチを図った．これにより，WebRTC を用いたアプリケーションひとつで，基本的なコンタクトセンター業務＋α の業務を一元化することができた．またこれらにより，カスタマに対して電話応対以上のサービスをより手軽に提供できる為，顧客満足度の向上に繋がるだけでなく，電話機設置も不要になる事から運用コスト削減等にも貢献できる事を想定できる.

課題としてまず一点目に，電話機が主流となっている通話機能を，WebRTC を用いてアプリケーションに盛り込んだが，その通信特性として UDP を用いた P2P 形式である故，通信が安定しない事態が多々発生する．これについては予期せぬ切断時にその原因を的確に突き止め，直ちに再接続を行うなどの予防策を考え，実装する必要がある.

また二点目に，WebRTC がサポートされたブラウザ環境に限りがあることが，WebRTC 普及の大きな柵になっていると考えられる．WebRTC を用いた本稿の試作のようなアプリケーションで実際のサービス業務を運用する場合，必然的にサービスを提供できる顧客が限定的になってしまう. iOS，IE ブラウザなど，現在サポートされていない環境についてもサポートが拡大されないことには，十分なサービス提供として WebRTC を利用することは難しくなる．今後 WebRTC が多くの環境でサポートされていくことを願う.

参考文献

[1] “WebRTC”. https://webrtc.org/.
[2] “WebRTC 1.0: Real-time Communication Between Browsers“. https://www.w3.org/TR/webrtc/.
[3] “Rtcweb Status Pages”. https://tools.ietf.org/wg/rtcweb/.
[4] “消費者と企業のコミュニケーション実態調査 2016”. http://www.trans-cosmos.co.jp/data/2016dec/.
[5] Alan B. Johnston (原著), Diniel C. Burnett (原著), 内田直樹 (翻訳). WebRTC ブラウザベースの P2P 技術. リックテレコム, 2014
[6] 本郷 直哉, 山本寛, 山崎克之「WebRTC による EC 弱者向けブラウザ連携システムの開発と評価」電子情報通信学会論文誌 2014/10 Vol. J97-B No.10 p890-902
[7] “SkyWay”. http://nttcom.github.io/skyway/.
[8] “Heroku”. https://www.heroku.com/.
[9] Brooke, J.; SUS: A "quick and dirty" usability scale. In P. W. Jordan, B. Thomas, B. A. Weerdmeester, & A. L. McClelland (Eds.), Usability Evaluation in Industry. London: Taylor and Francis. (1996). http://hell.meiert.org/core/pdf/sus.pdf.
[10] Jakob Nielsen (原著), 篠原 稔和 (翻訳), 三好 かおる (翻訳)「ユーザビリティエンジニアリング原論—ユーザーのためのインタフェースデザイン (情報デザインシリーズ) 」東京電機大学出版局 (出版) 2002