

スマートフォンデバイスを活用した 市町村防災行政無線支援システムの開発

佐々木 喜一郎

- はじめに
- 第一章 スマートフォンデバイスの変遷
 - 第一節 スマートフォンデバイスの普及
 - 第二節 スマートフォンデバイスの通知情報
 - 第三節 Android デバイスの通知機能
 - 第四節 iOS デバイスの通知機能
- 第二章 支援システムの開発
 - 第一節 Android アプリケーションの実装
 - 第二節 iOS アプリケーションの実装
- 第三章 支援システムの評価
- おわりに

はじめに

市町村では、防災、応急救助、災害復旧に関する情報伝達的手段として、市町村防災行政無線を使用している。市町村防災行政無線は、前述の使用方法の他に、平常時の一般行政事務に使用できる無線局である。市町村防災行政無線の種類は、同報系防災行政無線と移動系防災行政無線の2種類に大別される。同報系防災行政無線は、市町村役場から住民に対して屋外拡声器を利用して情報を伝えるシステムである。移動系防災行政無線は、行政機関内の通信手段として車載型や携帯型の移動局と市町村役場が情報通信を行うシステムである。

本研究は、市町村防災行政無線の同報系防災行政無線に関する諸問題の解決について、岐阜県本巣郡北方町から相談があったことから開始された。市町村防災行政無線の同報系防災行政無線は、設置されている屋外拡声器との距離や住宅の気密性の違いにより、音声情報が音量不足や音量過剰となり、住民へ適切に情報が伝達できていない。つまり、音量不足の状態では音声情報が聞こえない、音量過剰の状態では騒音となっている。また、屋外拡声器による情報音声の発信は、風向きや天候に左右されやすく、台風や豪雨などの防災情報の発信に不向きである。さらに、複数個所に設置された屋外拡声器の環境は、音声情報の旋律が間隔でずれるため聞き取りにくいという課題もある。[1][2][3][4][5] ゆえに、本研究では、市町村防災行政無線の同報系防災行政無線に関する諸問題を明らかにし、ICTを活用した市町村防災行政無線の同報系防災行政無線を支援

する新しい仕組みについて提案する。

本研究の意義は、危機的な自然現象や人為的な原因による社会基盤や人命に受ける被害の未然防止や拡大防止の充実かつ強化が可能なことである。本研究の新規性は、著しい増加傾向にあるスマートフォンに着目して、新しい情報発信の仕組みを考案し、実証したことである。また、ICTの技術的な革新とネットワーク経営の関係性について明らかにしたことである。

第一章 スマートフォンデバイスの変遷

第一節 スマートフォンデバイスの普及

本節は、スマートフォン普及の変遷について述べる。スマートフォンアプリケーションの要件定義を行うためには、スマートフォンデバイスの普及状況について深く理解し、今後の状況についても予測が必要である。なぜなら、デバイスやユーザの総数を考慮し、適切な対象を決定しなければならぬからである。はじめに、スマートフォンの爆発的な普及の要因を明らかにする。

はじまりは、世界に初めて携帯電話を使用したインターネットサービスが日本において提供された事である。なお、1999年1月に旧エヌ・ティ・ティ移動通信網株式会社（現株式会社NTTドコモ）がiモード、同年4月に旧DDIセルラーグループのIDO（現KDDI株式会社）がEZwebを開始した。このように、独自のインターネット接続環境を提供し、高機能なカメラやワンセグによるテレビ視聴機能、近距離無線通信による決済機能といった高度で特徴的な機能を搭載したフィーチャーフォンが登場した。ところが、フィーチャーフォンは、高度で特徴的な機能を実現するあまり、独自の操作性となり、取り扱いが難しくなっていた。また、ウェブサイトがそのまま閲覧が不可能で、専用のタグ言語で開発されたウェブサイトが必要であった。そこで、2001年5月に旧株式会社エヌ・ティ・ティ・ドコモ（現株式会社NTTドコモ）が、日本において世界初の三代携帯電話商用サービスである3Gサービスを開始した。その結果、フィーチャーフォンとパーソナルコンピュータを接続して、従来と比べて高速なデータ通信が行えるようになった。ところが、フィーチャーフォンの他に、ノート型のパーソナルコンピュータが必要であるため一般的に普及しなかった。

一方、米国では3Gサービスの普及とともに、小型の携帯情報デバイスであるPersonal Digital Assistant（以下、PDAとする。）が通信専門のデバイスとして普及した。PDAは、電子メールの送受信やブラウザによるウェブサイトの閲覧、ビジネスアプリケーションの利用、静止画の閲覧や動画再生が可能であった。これが、スマートフォンの登場と普及の土台を形成したのは疑う余地がない。そして、アップル社から2007年6月29日にアメリカ合衆国で初代iPhoneシリーズが発売された。なお、各国におけるiPhoneの発売は、オーストラリア、オーストリア、ベルギー、カナダ、デンマーク、フィンランド、フランス、ドイツ、香港、アイルランド、イタリア、日本、

メキシコ、オランダ、ニュージーランド、ノルウェー、ポルトガル、スペイン、スウェーデン、スイス、英国、米国の22カ国で2008年7月11日に発売されたiPhone 3Gである。これが、スマートフォン普及のはじまりである。また、黎明期におけるスマートフォン普及に影響を与えた。というのは、ガートナーのスマートフォンマーケットに関するプレスリリースより作成した世界のスマートフォンの売上台数推移（表1）から確認できる。

表1. スマートフォンデバイスのOS別出荷台数（数千台）

四半期	Android	iOS	Windows	BlackBerry	Symbian	その他	スマートフォン 総出荷数	携帯電話 総出荷数	備考
2007年第1四半期	-	-	2,931	2,080	15,844	4,087	24,943	-	[6]
2007年第2四半期	-	270	3,212	2,471	18,273	3,628	27,855	272,604	[7]
2007年第3四半期	-	1,104	4,180	3,192	20,664	3,612	32,752	291,142	[8]
2007年第4四半期	-	1,928	4,374	4,025	22,903	3,536	36,766	330,055	[9]
2008年第1四半期	-	1,726	3,858	4,312	18,400	4,113	32,408	294,283	[10]
2008年第2四半期	-	893	3,874	5,594	18,405	3,456	32,221	304,722	[11]
2008年第3四半期	-	4,720	4,053	5,800	18,179	3,763	36,515	308,532	[12]
2008年第4四半期	639	4,079	4,714	7,443	17,949	3,319	38,143	314,708	[13]
2009年第1四半期	575	3,848	3,739	7,534	17,825	2,986	36,507	269,120	[14]
2009年第2四半期	756	5,325	3,830	7,782	20,881	2,398	40,972	286,122	[15]
2009年第3四半期	1,425	7,040	3,260	8,523	18,315	2,531	41,093	308,895	[16]
2009年第4四半期	4,043	8,676	4,203	10,508	23,857	2,517	53,804	347,103	[17]
2010年第1四半期	5,227	8,360	3,696	10,753	24,068	2,403	54,506	359,605	[18]
2010年第2四半期	10,653	8,743	3,059	11,629	25,387	2,588	62,058	367,987	[19]
2010年第3四半期	20,544	13,484	2,204	12,508	29,480	2,912	81,133	417,086	[20]
2010年第4四半期	30,801	16,011	3,419	14,762	32,642	3,515	101,150	452,037	[21]
2011年第1四半期	36,350	16,883	2,582	13,004	27,599	3,357	99,775	427,846	[22]
2011年第2四半期	46,776	19,629	1,724	12,652	23,853	3,107	107,740	428,661	[23]
2011年第3四半期	60,490	17,295	1,702	12,701	19,500	3,497	115,185	440,502	[24]
2011年第4四半期	75,906	35,456	2,759	13,185	17,458	4,278	149,042	476,555	[25]

それでは、ガートナーのスマートフォンマーケットに関するプレスリリースによって、2007年以降の世界におけるスマートフォンの売上台数推移を確認する。（表1）を参照すると、2007年第1四半期の24,943,000台から2018年第1四半期の383,503,000台へ急速に拡大している。その大半がシェア率を占めるのは、Androidデバイスである。2010年第1四半期の5,227,000台から2011年第1四半期の36,350,000台へ急速に拡大している。この第1次の爆発的な普及は、Android 2.2 Froyoに対応した機種が発売された事が要因である。[26] 具体的には、Samsung社のGalaxy S

シリーズ、HTC社のDesireシリーズの世界的なヒットが大きな要因である。また、Android 2.2 Froyoに関係する技術的な要因が5つある。はじめに、Androidプラットフォームで採用されていたレジスタベースの仮想マシンDalvik VMにJITコンパイラを搭載し、システムやアプリケーションに関する処理が2倍から5倍に高速化できた事である。次に、JavaScript実行エンジンであるGoogle V8 JavaScript Engineを搭載し、JavaScriptで記述されたウェブサイトの処理が2倍から3倍に高速化できた事である。次に、Kernel Memory Management Boostに対応し、アプリケーションの切り替え速度の高速化が実現できた事である。次に、Adobe社が開発した動画やゲームを扱うための規格Adobe Flashに対応し、マルチメディアコンテンツの閲覧や利用ができた事である。最後に、クラウドとAndroidデバイス間のメッセージングを可能とする双方向プッシュ同期機能が実現し、デバイス通知及びユーザ通知を活用できた事である。つまり、デバイス間のメッセージングを可能とする双方向プッシュ同期機能は、スマートフォンデバイス向けに開発されたアプリケーションの発展に大きく貢献した。そこで、本研究では、デバイス通知及びユーザ通知を活用することにより、市町村防災行政無線の同報系防災行政無線を支援するシステムの開発をする方針とした。

第二節 スマートフォンデバイスの通知情報

本節は、AndroidやiOSが搭載されたデバイスであるスマートフォンやタブレットデバイスの通知情報について明らかにする。[27] 前述におけるデバイスの通知情報は、大別するとユーザ通知とデバイス通知の2種類に分けられる。ユーザ通知は、AndroidやiOSが搭載されたデバイスであるスマートフォンやタブレットデバイスから、その利用者であるユーザに通知情報を伝える方式である。この機能は、AndroidにおけるダイアログやiOSにおけるアラートとして実装される。また、ユーザへアプリケーションの情報を伝え、ユーザへ実行して欲しい行動を促す手段として活用される。つまり、ユーザ通知を活用する事により、アプリケーションに関わる情報を意図的に伝え、アプリケーションの利便性を向上させる事ができる。デバイス通知は、サーバからAndroidやiOSが搭載されたデバイスであるスマートフォンやタブレットデバイスに通知情報を伝える方式である。この機能は、通知情報のクラウドサービスと通知情報ライブラリーを利用して実装される。また、ユーザへアプリケーションの情報を伝え、ユーザへ実行して欲しい行動を促す手段だけでなく、継続してアプリケーションを活用してもらう仕掛けとして利用される。つまり、デバイス通知を活用する事により、アプリケーションサービスの提供側に関わる情報を意図的に伝え、様々なクラウドサービスと連携した結果を通知する拡張的な機能を提供する事ができる。ゆえに、ユーザ通知とデバイス通知を駆使したアプリケーションが数多く存在する。

これらのアプリケーションのデバイス通知の実装方法は、大別するとプッシュ通知とフェッチ通知の2種類に分けられる。プッシュ通知は、サーバからの通知情報をユーザのデバイスがいつも待ち受けている状態で即座に受信する方法である。フェッチ通知は、サーバからの通知情報を

ユーザのデバイスから設定された時間毎に受信する方法である。これらの2種類の通知情報の違いについて、メッセージングアプリケーションを具体例として比較する。プッシュ通知は、サーバからメッセージの受信をリアルタイムに通知してくれる。しかし、常にメッセージの受信を監視する必要がある為、バッテリーの消耗がフェッチ通知と比べると多い。フェッチ通知は、メッセージの受信を定期的に確認して通知してくれる。ユーザの時間設定による受信であるため、リアルタイム性に欠ける。しかし、スマートフォンのバッテリーの消耗がプッシュ通知と比べると少ない。これらのデバイス通知の実装方法は、アプリケーションの用途やアプリケーションサービスプロバイダの事業計画により、採用される方法が変わる。なお、両方の方法が採用される場合もある。

第三節 Android デバイスの通知機能

本節では、Android デバイスの通知機能について明らかにする。Android デバイスは、アプリケーションストアから提供されるサードパーティ製スマートフォンアプリケーションから、デバイス通信方式のプッシュ通知が実現された2番目のモバイルデバイスである。まず、2010年5月21日にGoogle社からAndroid 2.2 FroyoのSDK2.2がリリースされ、2010年6月23日にソースコードが開示された事により、デバイス通信方式のプッシュ通知アプリケーションの開発が可能になった。次に、2010年5月20日にGoogle Cloud Messaging（以後、GCMとする。）の前身であるAndroid Cloud to Device Messaging（以後、C2DMとする。）が開始され、アプリケーションサービスプロバイダ（以後、ASPとする。）のサーバからデバイス通知を送信するサービスが開始されたためである。なお、GCMは、2012年7月9日から開始され、2012年6月27日にGoogle社からリリースされたAndroid 4.1 Jelly Beanで通知情報の受信が可能になった。すなわち、Android 2.2 FroyoのSDK2.2とC2DMにより、デバイス通知で受信したデータをユーザ通知で表示する機能が実現された。また、次期仕様では、Android 4.1 Jelly BeanとGCMにより、デバイス通知で受信したデータをユーザ通知で表示する機能が実現された。この時のユーザ通知とは、トースト、ダイアログ、ノーティフィケーション、音声読み上げの4種類の表現方法を送信側で決定し、アプリケーション毎に任意で選択する事、もしくはユーザ通知そのものを任意で無しに設定する事も可能である。

トーストは、スクリーンにASPから送信された文字列データを表示させる。このトーストには、ASPから送信された文字列データを表示し、一定時間が経過すると自動的に表示が終了される。当時の具体的な実装方法は、Toast Notificationが使用された。

ダイアログは、アプリケーションのスクリーンにダイアログボックスを表示させる。ダイアログボックスには、ASPから送信された文字列データを表示し、イベントをユーザへ通知して必要に応じてユーザに応答を求める。この時に、ユーザがダイアログボックスのボタンをタップするアクションをするまでは表示が継続される。当時の具体的な実装方法は、ProgressDialogやProgressBarが使用された。なお、リスト形式やラジオボタン形式、チェックボックス形式など複数の候

補を表示し、ユーザが任意に選択する表示も可能である。

ノーティフィケーションは、画面上部のステータスバーの領域にメッセージを表示させて、画面が表示されていないアクティビティやバックグラウンドで動作しているサービスが、ユーザに対して情報を伝えたい場合に利用される。当時の具体的な実装方法は、`NotificationManager` が使用された。

音声読み上げは、テキストから音声を合成し、サウンドファイルを作成して再生させる。なお、Android デバイスに標準で搭載されている音声合成エンジン以外にも、他のアプリケーションと連携させて、様々な言語を再生させることが可能である。当時の具体的な実装方法は、`TextToSpeech` が使用された。これらのユーザ通知のパケット通信において、パケットに含まれるヘッダやトレーラなどの付加的情報を除いたデータそのものであるペイロードデータの上限は、受信する側の Android のバージョンにより異なる。

第四節 iOS デバイスの通知機能

本節では、iOS デバイスの通知機能について明らかにする。iOS デバイスは、アプリケーションストアから提供されるサードパーティ製スマートフォンアプリケーションから、デバイス通信方式のプッシュ通知が実現された初めてのモバイルデバイスである。これは、2009年6月17日に Apple Inc. から提供された2つのサービスにより実現化された。まず、iOS3.0 がリリースされ、デバイス通信方式のプッシュ通知が iOS3.0 を導入した iPhone で通知情報の受信が可能になったためである。なお、この時の iPhone の対応機種は、発売されていた機種としては初代の iPhone シリーズと iPhone 3G、発売時に搭載された機種としては iPhone 3GS である。次に、Apple Push Notification Service (以後、APNS とする。) が開始され、ASP のサーバからデバイス通知を送信するサービスが開始されたことである。すなわち、iOS3.0 と APNS により、デバイス通知で受信したデータをユーザ通知で表示する機能を実現された。この時のユーザ通知とは、バナー、ダイアログ、サウンド、バッチによる4種類の表現方法である。ただし、ユーザは4種類の表現方法を任意で選択する事、もしくはユーザ通知そのものを任意で無しに設定する事も可能である。

バナーは、スクリーンの上部に ASP から送信された文字列データを表示させる。このバナーには、SP から送信された文字列データを表示し、一定時間が経過すると自動的に表示が終了される。当時の具体的な実装方法は、`PushNotification` が使用された。なお、アプリケーションが起動していない場合、アプリケーションがフォアグラウンドで起動している場合、アプリケーションがバックグラウンドで起動している場合を想定した実装が必要である。

ダイアログは、アプリケーションのスクリーンにダイアログボックスを表示させる。ダイアログボックスには、ASP から送信された文字列データを表示し、イベントをユーザへ通知して必要に応じてユーザに応答を求める。この時に、ユーザがダイアログボックスのボタンをタップするアクションをするまでは表示が継続される。当時の具体的な実装方法は、アラートダイアログボック

ス表示に `UIAlertView`, ボタンのタップ制御に `UIAlertViewDelegate` が使用された。サウンドは、システムやアプリケーションのリソースに準備されているサウンドリソースを再生して、BGM や効果音、音声によりユーザへ通知情報の受信を知らせる。当時の具体的な実装方法は、システムサウンドの制御に `AudioServicesPlaySystemSound`, アラートサウンドの制御に `AudioServicesPlayAlertSound` が使用された。なお、サウンド制御だけでなく各サウンドと連動したパイプレーションの制御も可能である。

バッジは、ホームスクリーンのアプリケーションアイコンに、ユーザに通知するイベントの数を数値アイコンにより通知する。すなわち、通知情報があるアプリケーションアイコンの右上に、通知情報の数が表示される。これにより、視覚的に通知情報があったことを再確認できる。当時の具体的な実装方法は、`UIApplication` が使用された。なお、`UIApplication sharedApplication` で取得したプロパティ `applicationIconBadgeNumber` に整数値をセットする事で利用できる。なお、これらのユーザ通知のパケット通信において、パケットに含まれるヘッダやトレーラなどの付加的情報を除いたデータそのものであるペイロードデータの上限は、受信する側の iOS のバージョンにより異なる。

第二章 支援システムの開発

第一節 Android アプリケーションの実装

GCM から Android デバイスへプッシュ送信するアルゴリズムについて述べる。はじめに、Android デバイスへプッシュ送信をするには、事前準備として Android デバイスを GCM サーバへ登録する必要がある。GCM サーバを使用するには Google アカウントが必要である。Google アカウントでログイン後に GCM コンソールへアクセスし、Android アプリケーションパッケージ名を登録してプロダクト ID を取得する。また、プッシュ送信サーバからのアクセスを許可する `GOOGLE_API_KEY` を取得する。次に、Android デバイスでのデバッグ起動が必要である。プッシュ送信を受信する Android アプリケーションの設定は、Android アプリケーション Project の `AndroidManifest.xml` にネットワークとパイプレーションの利用、スリープからの復帰とプッシュ送信の受信を許可する構文を記述する。また、プッシュ送信のデータを解析するためのレシーバを登録する構文を記述する。プロダクト ID は Android アプリケーションの Java プログラム内に記述する。プッシュ送信の受信に必要な `registration id` の取得は `Activity` の `onCreate` 関数内に記述する。また、`registration id` を取得する際にプロダクト ID を GCM サーバへ送信し、`registration id` はアプリケーション内のローカルストレージに保存する。さらに、プッシュ送信サーバへ `registration id` を送信する。なお、プッシュ送信を受信するには実機の Android デバイスが必要である。つまり、アプリケーションを実機にインストールして実行する。実行後の Android デ

バイスを GCM サーバへ登録する流れは、(図 1) のフローチャートに示す。

Android デバイスへプッシュ送信するアルゴリズムについて述べる。プッシュ送信サーバのウェブページへログインし、災害情報入力画面を表示させて、災害情報入力画面から本文と災害情報を入力し、入力情報確認画面を表示させる。なお、入力されたデータは、プッシュ送信サーバ内で GCM サーバへ送信するために JSON データを作成する。送信する registration id はデータベースから配列データとして取得する。送信するために作成する JSON データの内容は、to に registration id を配列で記述する。また、notification 内に message タグをつけてその中に本文と災害情報を区切り文字で繋いだメタデータを記述する。GCM サーバへプッシュ送信をするには、HTTP header に GOOGLE_API_KEY と Content-Type を設定し、作成した JSON データを POST データとして送信する。プッシュ送信サーバがプッシュ送信するデータを GCM サーバへ送信する流れについては、(図 2) のフローチャートに示す。プッシュ送信サーバからデータを受信した GCM サーバは、データ内にある registration id の数を計測し、その数分だけ送信を繰り返して送信結果を記録する。送信結果の内容は、正常に送信できた success と送信できないまたは受信が拒否された failure と GCM サーバで管理されている multicast_id と message_id である。GCM サーバから Android デバイスへ送信する流れは (図 3) のフローチャートに示す。

Android デバイスがプッシュ送信を受信するアルゴリズムについて述べる。まず、プッシュ送信サーバからプッシュ送信を受信した Android デバイスはインストールされているアプリケーション全てへプッシュデータをブロードキャスト送信する。ブロードキャストデータ受信したアプリケーションは GcmBroadcastReceiver の onReceive 関数でプッシュデータの解析を行う。GcmBroadcastReceiver の onReceive 関数で解析する内容は、プッシュデータの構造に間違いがあるかと GCM サーバ側でデータが削除されたかを解析する。プッシュデータの構造に間違いがなければ、その中から message タグを解析し、紐付けられたメタデータを取得する。取得したメタデータを区切り文字で文字配列へ分解し通知領域に表示する本文と災害情報を抽出する。通知領域に表示するアイコンとバイブレーションパターンの設定と通知音であるサウンドの設定をして、通知領域に表示する。通知領域に表示された通知からアプリケーションを起動したときに通知内容のウェブページを表示する動作を登録する。Android デバイスがプッシュ通知を受信して処理する流れは、(図 4) のフローチャートに示す。

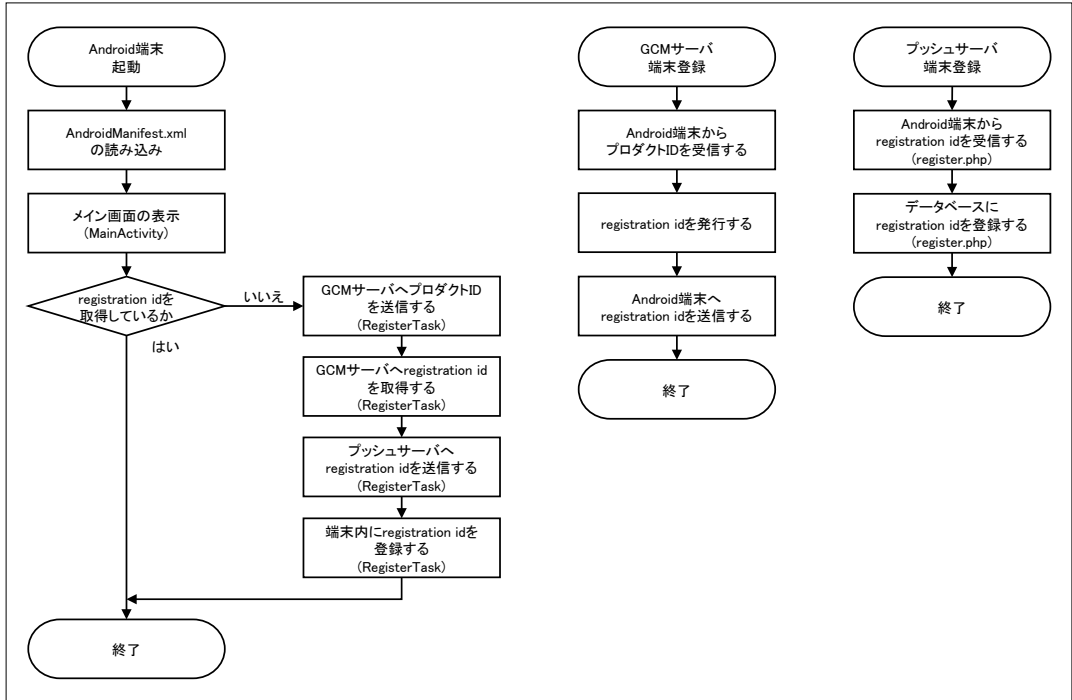


図 1. GCM 版プッシュ送信受信デバイス登録処理のフローチャート

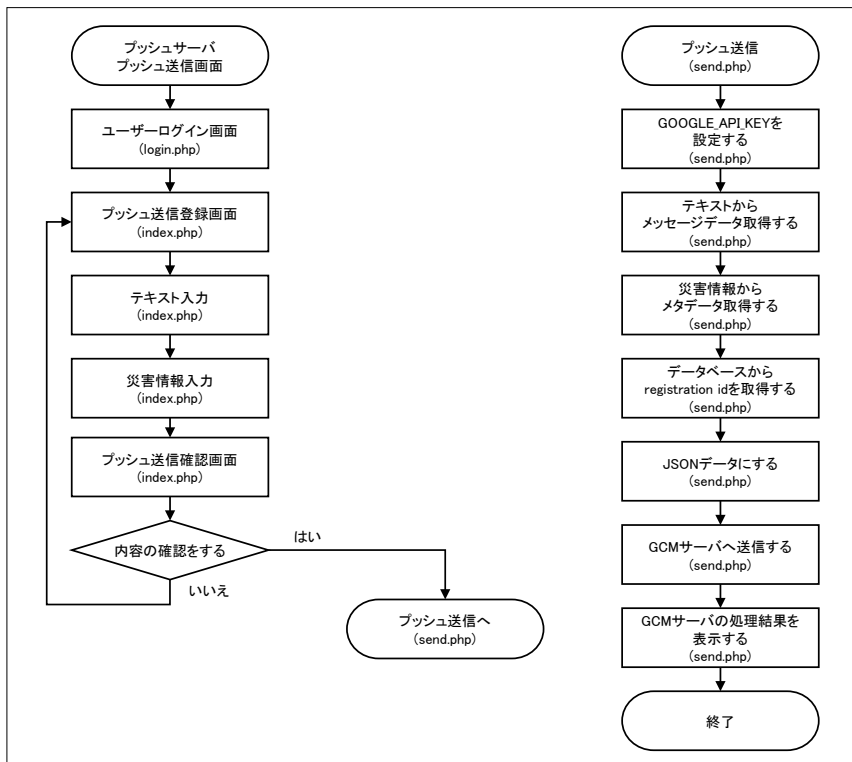


図 2. GCM 版プッシュ送信プッシュサーバのフローチャート

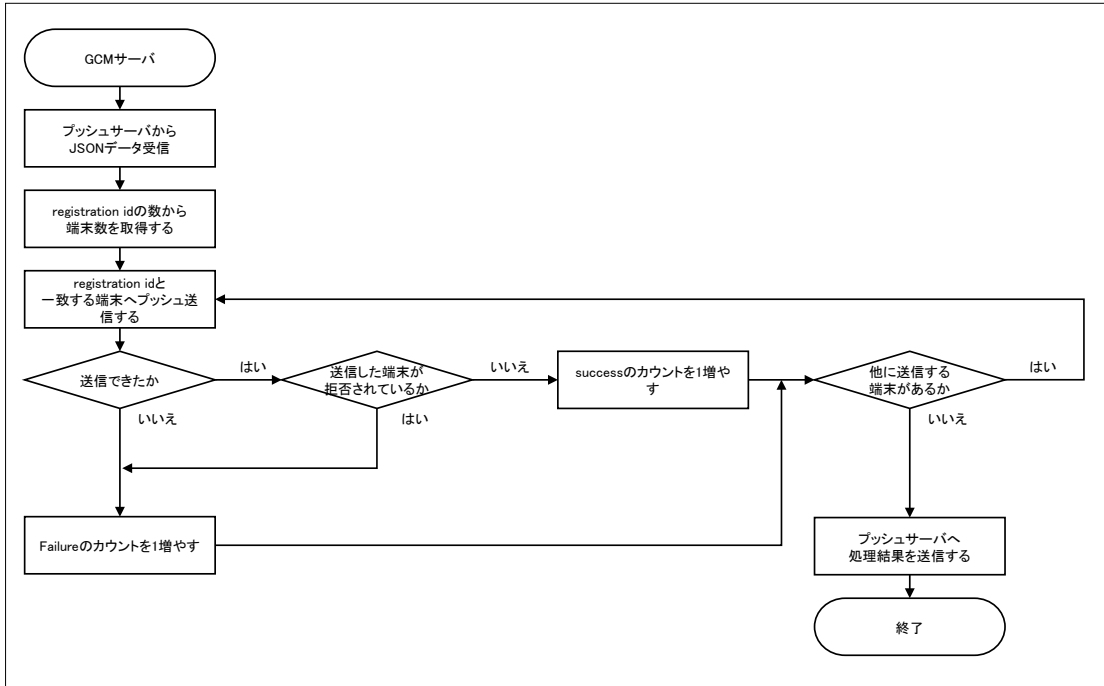


図 3. GCM 版プッシュ送信 GCM サーバのフローチャート

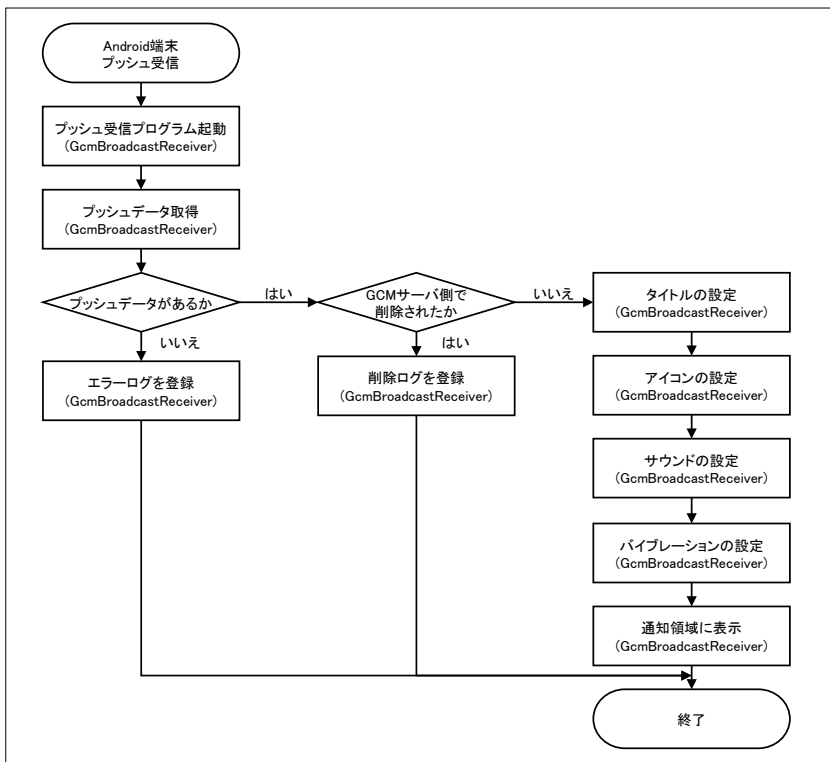


図 4. GCM 版プッシュ送信受信デバイスのフローチャート

第二節 iOS アプリケーションの実装

APNS から iOS デバイスへプッシュ送信するアルゴリズムについて述べる。はじめに、iOS デバイスへプッシュ送信をするには、事前準備として iOS デバイスを APNS サーバへ登録する必要がある。APNS サーバを使用するには Apple ID が必要である。Apple ID でログインをして有料の iOS Developer Program に登録する。iOS でプッシュ送信を行うには Apple ID の証明書を発行する必要がある。証明書は、MacOS のキーチェーンアプリから認証局に Apple ID を登録して開発用証明書を発行する。プッシュ送信を受信する iOS アプリケーションを iOS Developer Program へ登録する。なお、iOS Developer Program コンソールの App ID へアクセスし iOS アプリケーションの Bundle ID を登録し、App Services の Push Notifications を有効にすることで登録ができる。プッシュ送信サーバが APNS サーバへアクセスを許可するための証明書を発行する。APNS 用証明書の発行は、iOS Developer Program コンソールの Certificates へアクセスし、開発用証明書を使用して APNS サーバへアクセスする APNS 用証明書を発行する。APNS 用証明書は開発版の Apple Push Notification service SSL の Sandbox 証明書と製品版の Apple Push Notification service SSL の Production 証明書の 2 つを取得する必要がある。この 2 つの証明書は、プッシュ送信サーバが APNS に接続する際に、開発デバイスに送信する場合や製品アプリケーションへ送信する場合によって、証明書の切り替えができるようになっている。また、この 2 つの証明書は、そのままプッシュ送信サーバから APNS サーバへのアクセスには使用できない。ゆえに、MacOS のキーチェーンアプリを使用して登録し、p12 形式のファイルに出力し openssl を使用して pem 形式のファイルに変換する。そして、プッシュ送信サーバへ pem ファイルを送信する。iOS アプリケーションの設定は、Xcode の iOS アプリケーション Project の Target から App Service の PushNotification を有効にする必要がある。

iOS デバイスでのデバッグ起動は、AppDelegate.m ファイルの didFinishLaunchingWithOptions 関数へ、APNS サーバへのアクセスとアプリケーションの使用者にプッシュ送信を受信する許可を求めるダイアログを表示するプログラムを記述する。また、プッシュ送信を受信した際にプログラム側でバッジ表示の許可と通知音のサウンド設定の許可と表示領域の設定の許可をするプログラムを記述する。プッシュ送信の受信に必要な deviceToken の取得は、AppDelegate.m ファイルの didRegisterForRemoteNotificationsWithDeviceToken 関数内に記述する。deviceToken は、アプリケーション内のローカルストレージに保存し、プッシュ送信サーバへ deviceToken を送信する。プッシュ送信を受信するには実機の iOS デバイスが必要であるため、アプリケーションを実機にインストールして実行する。実行後の iOS デバイスを APNS サーバへ登録する流れについては（図 5）のフローチャートに示す。

iOS デバイスへプッシュ送信するアルゴリズムについて述べる。プッシュ送信サーバのウェブページへログインし、災害情報入力画面を表示し、災害情報入力画面から本文と災害情報を入力

すると入力情報確認画面が表示される。この際、入力されたデータは、プッシュ送信サーバ内で APNS サーバへ送信するために JSON データを作成する。なお、送信する DeviceToken は、データベースから配列データとして取得する。また、送信するために作成する JSON データの内容は `aps` に `alert` タグの中に `body` タグをつけてその中に本文と災害情報を区切り文字で繋いだメタデータを記述する。また、`alert` タグの中に `title` タグをつけて本文を記述する。APNS サーバへプッシュ送信をするには、デバッグでテストをするのであれば開発版の Sandbox 用証明書を使用し、インストールされたアプリケーションに送信するのであれば製品版の Production 用証明書を使用し、作成した JSON データを SSL 経由の APNS のポート 2195 へ送信する。また、開発版と製品版では、送信する URL が別々にあり、開発版では `gateway.sandbox.push.apple.com` を使用し、製品版では `gateway.push.apple.com` を使用する。プッシュ送信サーバがプッシュ送信するデータを APNS サーバへ送信する流れについては (図 6) のフローチャートに示す。

プッシュ送信サーバからデータを受信した APNS サーバは、データ内にある DeviceToken の数を計測し、その数分だけ送信を繰り返して送信失敗があればそれ以降の送信を停止させる。送信結果は、送信失敗があればエラー内容をプッシュ送信サーバへ送信する。なお、APNS サーバでは、複数の iOS デバイスに対して一括送信ができず、途中で送信できないデバイスがあると、それ以降の送信を停止させる設定になっている。また、送信失敗を繰り返すとプッシュ送信を一定期間受け取らなくなる。APNS サーバから iOS デバイスへ送信する流れは (図 7) のフローチャートに示す。

iOS デバイスがプッシュ送信を受信するアルゴリズムについて述べる。プッシュ送信サーバからプッシュ送信を受信した iOS デバイスは iOS がデータを処理する。iOS は、デバイスがスリープしているかどうか判断し、スリープしていれば iOS 内でプッシュデータを保存して通知領域に表示をする。また、スリープ状態から通知領域経由でアプリケーションを起動した場合のみアプリケーションへプッシュデータを送信する。iOS デバイスがスリープしていなければ、iOS からアプリケーションへプッシュデータが送信されアプリケーション内で処理が実行され通知領域に表示する。iOS デバイスがプッシュ通知を受信して処理する流れは (図 8) のフローチャートに示す。iOS で受信した場合は、4 種類の表示形式である (図 9) (図 10) (図 11) (図 12)。

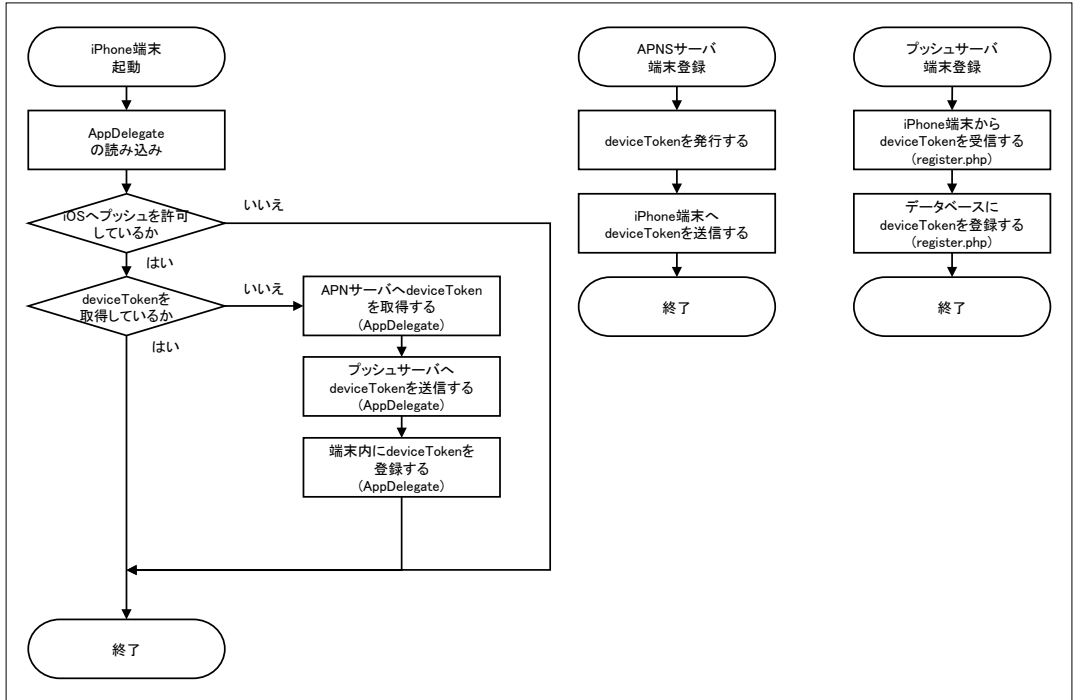


図 5. APNS 版プッシュ送信受信デバイス登録処理のフローチャート

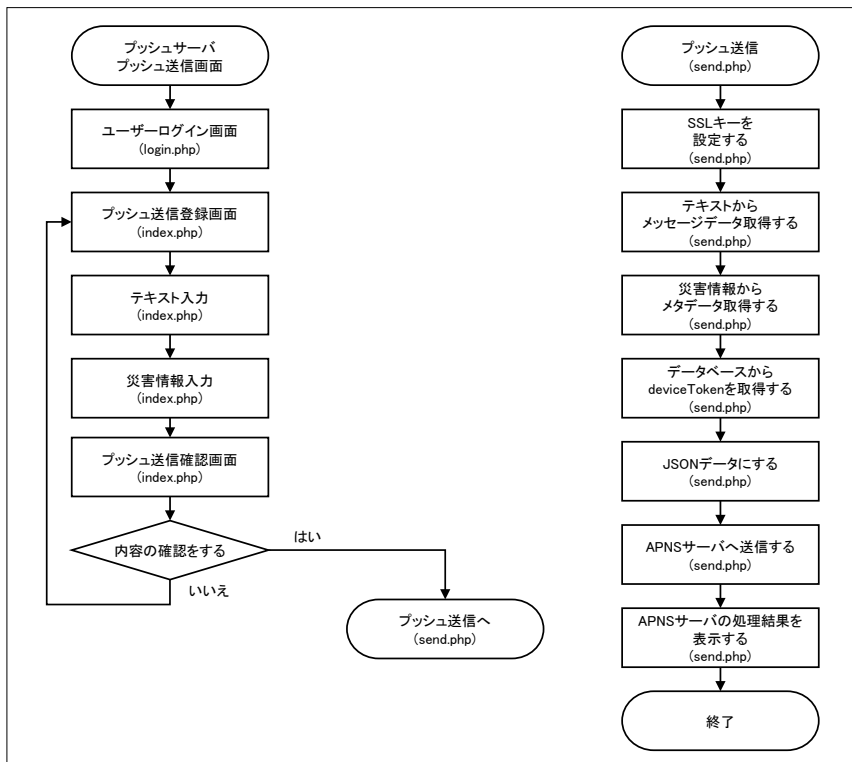


図 6. APNS 版プッシュ送信プッシュサーバのフローチャート

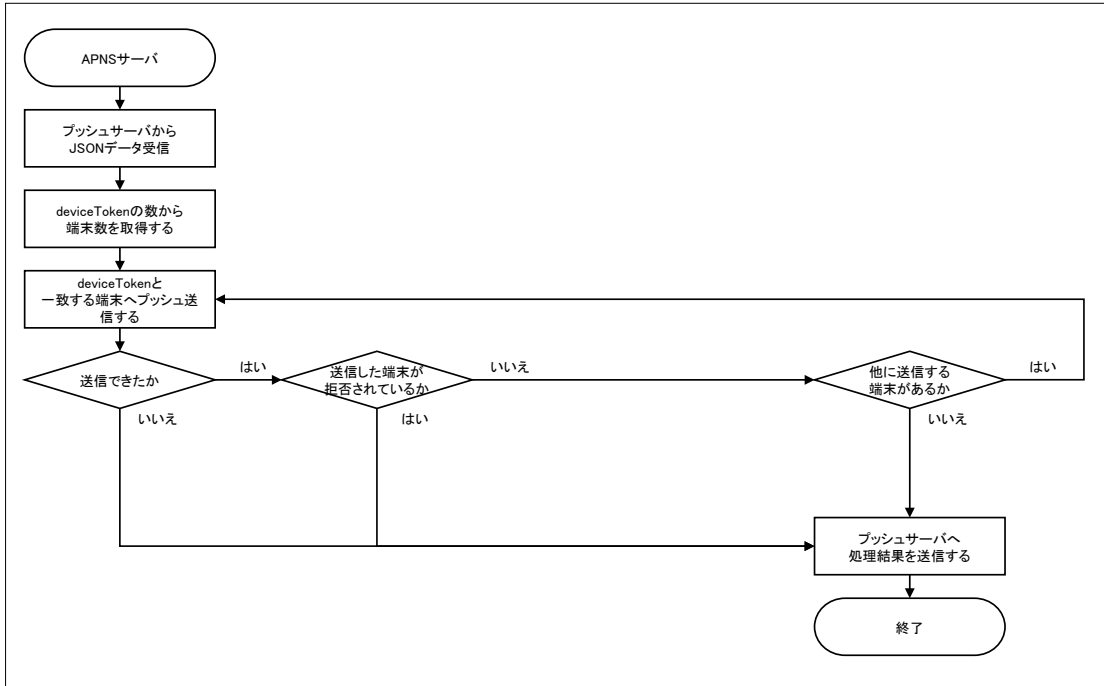


図 7. APNS 版プッシュ送信 APNS のフローチャート

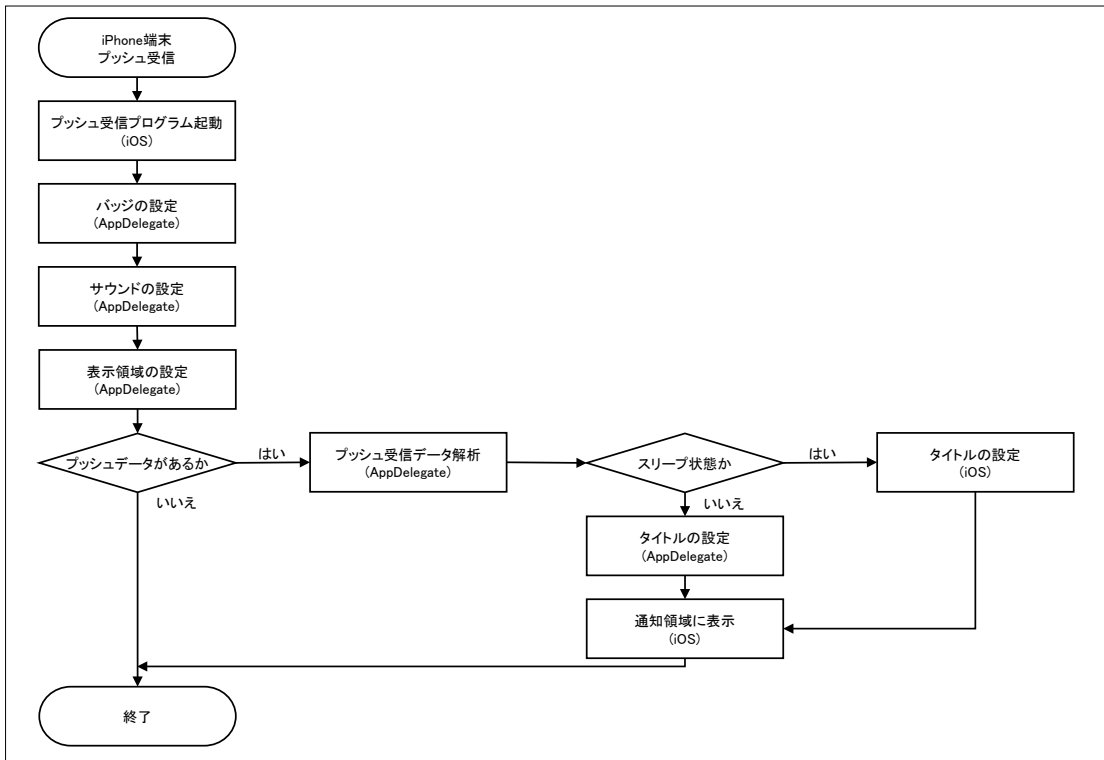


図 8. APNS 版プッシュ送信受信デバイスのフローチャート



図 9. バナーによる通知情報



図 10. ダイアログボックスによる通知情報



図 11. 通知センターによる通知情報



図 12. ロック画面による通知情報

第三章 支援システムの評価

本システムの評価を行うため、岐阜県本巣郡北方町において実証実験を実施した。なお、市町村防災行政無線を使用する職員 12 名に対して、アンケート調査とインタビュー調査を実施した。アンケートの内容は、①アンケート対象者の属性について、②災害情報送信画面フォームのログイン画面について、③災害情報送信画面フォームの情報送信画面について、④災害情報送信画面フォームの送信内容確認画面について、⑤災害情報送信画面フォームの送信完了画面について、⑥災害情報受信画面について、⑦全体的な御意見と御感想についての 7 条目である。各条目は、対象項目に対しての 5 段階評価と記述式で構成した。そのアンケート結果について、各条項の項目別に述べる。この際、5 段階中 3 の評価や未回答、判別不可能な回答は、未評価に分類している。

②災害情報送信画面フォームのログイン画面についてのアンケート結果は以下となった。2-1「ログイン画面全体は見やすいですか」の結果は、プラス評価 8 名、マイナス評価 2 名、未評価 2 名であった。理由の記述欄は、「ID とパスワードを忘れた際の措置について表記されているといい」、「シンプル」であった。2-2「文字のサイズは適切ですか」の結果は、プラス評価 6 名、マイナス評価 1 名、未評価 5 名であった。理由の記述欄は、「字が薄く小さく感じる」、「もう少し大きいほうがいい」、「大きくて良い」であった。2-3「ボタンのサイズは適切ですか」の結果は、プラス評価 7 名、マイナス評価 0 名、未評価 4 名である。理由の記述欄は、「字が薄く小さく感じる」であった。2-4「ボタンの位置は適切ですか」の結果は、プラス評価 9 名、マイナス評価 1 名、未評価 2 名であった。理由の記述欄は、「リセットよりログインが左だと嬉しい」、「逆だと良い」であった。2-5「ボタンの色は適切ですか」の結果は、プラス評価 6 名、マイナス評価 0 名、未評価 6 名であった。理由の記述欄は、「パスワードも緑色にした方が良い」、「入力が必要な個所は色付きが良い」であった。以上により、概ねプラス評価であることから、問題なく利用可能である事が示唆された。しかし、画面構成やオブジェクトの表現方法に課題があり、ユーザインタフェースの改善が必要である事が示唆された。

③災害情報送信画面フォームの情報送信画面についてのアンケート結果は以下となった。3-1「送信画面全体は見やすいですか」の結果は、プラス評価 8 名、マイナス評価 0 名、未評価 4 名であった。理由の記述欄は、「スッキリしていて使いやすいがスッキリしすぎているかもしれない」であった。3-2「文字のサイズは適切ですか」の結果は、プラス評価 6 名、マイナス評価 1 名、未評価 5 名であった。理由の記述欄は、「薄く小さい気がする」、「簡単な説明文も入れてほしい」、「大きくて良い」であった。3-3「ボタンのサイズは適切ですか」の結果は、プラス評価 5 名、マイナス評価 0 名、未評価 7 名であった。理由の記述欄は、「大きくて良い」であった。3-4「ボタンの位置は適切ですか」の結果は、プラス評価 9 名、マイナス評価 1 名、未評価 2 名であった。理由の記述欄は、「送信と戻るの位置が逆の気がする」であった。3-5「ボタンの色は適切ですか」の結果は、プラス評価 6 名、マイナス評価 1 名、未評価 5 名であった。理由の記述欄は、「両方同じ色では操作を誤りやす

い」であった。3-6「送信する情報内容の分類、項目で必要でないものはありますか」の結果は、「火山に関する情報」、「津波情報」、「土砂災害」であった。3-7「送信する情報内容の分類、項目で必要だと思うものはありますか」の結果は、「公共施設や医療情報」であった。以上により、概ねプラス評価であることから、問題なく利用可能である事が示唆された。しかし、オブジェクトの表現方法に課題があり、必要不可欠な選択肢が必要であることから、構造の改善が必要である事が示唆された。

④災害情報送信画面フォームの送信内容確認画面についてのアンケート結果は以下となった。4-1「送信画面全体は見やすいですか」の結果は、プラス評価7名、マイナス評価1名、未評価4名であった。理由の記述欄は、「シンプル」、「内容」、「20文字いっぱい入力した場合改行後の頭文字位置が悪い」であった。4-2「文字のサイズは適切ですか」の結果は、プラス評価7名、マイナス評価1名、未評価4名であった。理由の記述欄は、「小さい」、「大きくても良い」であった。4-3「ボタンのサイズは適切ですか」の結果は、プラス評価5名、マイナス評価0名、未評価7名であった。理由の記述欄は、「大きくても良い」であった。4-4「ボタンの位置は適切ですか」の結果は、プラス評価9名、マイナス評価1名、未評価2名であった。理由の記述欄は、「位置が逆の気がする」であった。4-5「ボタンの色は適切ですか」の結果は、プラス評価7名、マイナス評価1名、未評価4名であった。理由の記述欄は、「同色より色を変えた方が良い」、「本当に送信することを明らかにするために色を変えた方が良い」であった。以上により、概ねプラス評価であることから、問題なく利用可能である事が示唆された。しかし、画面構成やオブジェクトの表現方法に課題があり、デザインの改善が必要である事が示唆された。

⑤災害情報送信画面フォームの送信完了画面についてのアンケート結果は以下となった。5-1「送信画面全体は見やすいですか」の結果は、プラス評価8名、マイナス評価0名、未評価4名であった。理由の記述欄は、「シンプル」であった。5-2「文字のサイズは適切ですか」の結果は、プラス評価6名、マイナス評価0名、未評価6名であった。理由の記述欄は、「大きくても良い」であった。5-3「ボタンのサイズは適切ですか」の結果は、プラス評価6名、マイナス評価0名、未評価6名であった。理由の記述欄は、「大きくても良い」であった。5-4「ボタンの位置は適切ですか」の結果は、プラス評価9名、マイナス評価0名、未評価3名であった。5-5「ボタンの色は適切ですか」の結果は、プラス評価8名、マイナス評価0名、未評価4名であった。以上により、概ねプラス評価であることから、問題なく利用可能である事が示唆された。

⑥災害情報受信画面についてのアンケート結果は以下となった。6-1「送信画面全体は見やすいですか」の結果は、プラス評価10名、マイナス評価1名、未評価1名であった。理由の記述欄は、「表示が暗く感じる」、「災害時の背景色とイベント時の背景色が違うといい」、「POPUPの文字も大きくした方が老人に見やすい」、「目が良いから」であった。6-2「文字のサイズは適切ですか」の結果は、プラス評価4名、マイナス評価3名、未評価5名であった。理由の記述欄は、「振り込め詐欺の表示がすべて表示されなかった」、「下の文字がもう少し大きい方がよい」、「高齢者の人には小さい」、「小さい」であった。6-3「ボタンのサイズは適切ですか」の結果は、プラス評価6名、

マイナス評価 0 名、未評価 6 名であった。理由の記述欄は、「ボタン内の文字が小さく感じる」、「小さい」であった。6-4「ボタンの位置は適切ですか」の結果は、プラス評価 11 名、マイナス評価 1 名、未評価 0 名であった。理由の記述欄は、「音の停止がわからない」、「アラームの停止がわかりにくい」であった。6-5「ボタンの色は適切ですか」の結果は、プラス評価 6 名、マイナス評価 3 名、未評価 3 名であった。理由の記述欄は、「やや暗い」、「明るい方が良い」、「わかりづらい」、「確認と閉じるで色分け」、「イワカンはない」であった。6-6「イラストの種類は適切ですか」の結果は、プラス評価 5 名、マイナス評価 2 名、未評価 5 名であった。理由の記述欄は、「わからなかった」、「自由に使えるイラストが多くあるといい」、「ネットから取得できるようにしたらどうか」、「ビックリマークはよいがイラストはわかりづらい」、「イラストがかわいくて緊急時のピリピリとした気持ちと和らげてくれる」であった。6-7「イラストのサイズは適切ですか」の結果は、プラス評価 4 名、マイナス評価 3 名、未評価 5 名であった。理由の記述欄は、「見えなかった」、「目立たせた方がよい」、「もう少し大きい方がわかりやすい」、「災害時はもう少しイラストが大きい方がよいと思う」、「適切と思います」であった。6-8「イラストの位置は適切ですか」の結果は、プラス評価 9 名、マイナス評価 2 名、未評価 1 名であった。理由の記述欄は、「気づかなかった」、「中より目立たせた方がよい」であった。6-9「受信音の種類は適切ですか」の結果は、プラス評価 4 名、マイナス評価 0 名、未評価 8 名であった。理由の記述欄は、「受信音は自動で切れるようにした方がよいと思う」、「音は機種によって違うようでしたが…」、「緊急性の度合いによっては種類や音量が変えられるといい」、「緊急っぽくはないかもしれない」であった。6-10「バイブレーションの種類は適切ですか」の結果は、プラス評価 7 名、マイナス評価 0 名、未評価 5 名であった。6-11「現在マナーモード中には音やバイブレーションは作動しませんかどう思いますか」の結果は、「マナーモードでも音やバイブは作動した方がよい」、「緊急性をチェックして、高いものは強制的に作動させるといいと思う」、「情報のレベルによっては鳴らしてほしい」、「強制的にバイブレーションだけは作動するようになると良い」、「バイブレーションは緊急情報のときは必要だと思う」、「マナーモード中でもバイブは動いた方がよい」、「作動させなければ気づかないので自動で作動するようにしては」、「作動していいと思う」、「普段マナーモードが多いので、気づかないかもしれない」、「実際マナーにしたら気づかなかった」、「マナーモード中でも作動するようにした方がよいと思う」であった。以上により、概ねプラス評価であることから、問題なく利用可能である事が示唆された。しかし、画面構成やオブジェクトの表現方法に課題があり、イラストも賛否両論である事から改善の検討が必要ある事が示唆された。

⑦全体についてのアンケート結果は以下となった。7-1「全体を通して御意見、御感想等ありましたらお願いします」の結果は、「9月に住民の意見を聞き（自主防災訓練づくり ws 2 回目）、10月に示せるといい（自主防災訓練当日）」、「受信画面は見やすいが確認画面は見にくいと感じた」、「閉じてしまった後に再度確認できるようにした方がよいと思う」、「ホームページの表示が PC 表示なので見にくい」、「ボタンの表示を「確認」ではなく「詳細」にしたほうがいいのでは?」、「左上のタイトルも情報によって表示が変わるようにした方がよい」、「必要な情報を登録して、登

録者だけが受信できるようになるといい」、「台風時の絵は必要?」、「アプリの容量はできるだけ小さくしてほしい」、「他アプリとのケンカにも注意をお願いします」、「受信したデータを再度確認できると良い」、「アイコンが緑のため背景画像が緑だと見えない」、「前の情報画面が出たまま確認ボタンを押さない状態で新しい情報が発生すると確認できない」、「アプリを立ち上げた初期状態では画面が見切れてしまう」、「高齢者にもわかりやすくしてほしい」であった。以上により、利用者の実情に合わせたインタフェースが必要である事が明らかとなった。今後、アプリケーションのユーザインタフェースの改善を実施したい。

おわりに

本研究は、ICTを活用した市町村防災行政無線の同報系防災行政無線を支援する新しい仕組みについて提案するため、課題の明確化、スマートフォンに関する動向調査、支援システムの要件定義、支援システムの設計及び開発、ユーザによる支援システムの評価を行った。

はじめに、同報系防災行政無線に関する諸問題を明らかにするために、関連研究から調査して分析を行った。その結果、従来の機器やシステムは一長一短があり、普及が拡大しているスマートフォンに対応した新しい仕組みが必要である事を明らかにした。次に、スマートフォンに対応した新しい仕組みを検討するために、スマートフォン普及の変遷について、経営的及び技術的な観点で調査して分析を行った。その結果、クラウドコンピューティングやスマートフォンに関する技術的な革新、それにとまなう新しい企業間連携サービスの進展が密接に関係している事が明らかとなった。つまり、ICTの革新とネットワーク経営の進展が密接不可分の関係である事が明確となった。特に今回の研究では、様々な企業間及びユーザ間のメッセージングを実現するクラウドサービスに注目して活用する事とした。次に、スマートフォンで利用できるメッセージングクラウドサービスの調査を行った。その結果、AndroidにはGCMを利用する事が有効であること、iOSにはAPNSを利用する事が有効である事を解明した。そして、メッセージングクラウドサービスと連携したシステム全体の設計を実施した。また、送信する側はウェブアプリケーションとして設計を実施した。さらに、受信する側はAndroidとiOSに対応するスマートフォンアプリケーションとして設計を実施した。次に、幾つかの留意点を明確にして実装を行った。具体的には、送信する側のウェブアプリケーションからGCMやAPNSに送信するペイロードデータが不正確であると、ブラックリストに登録されてサービスが利用できなくなる恐れがあった。特に、APNSは、ペイロードデータの送信を対象のスマートフォン毎に送信する必要があるため、サービスの予想を上回るタイミングで連続送信されても、前述と同様の事態を招く恐れがあった。そこで、ペイロードデータを厳密に生成するアルゴリズムを考案して対応した。また、ペイロードデータを送信するタイミングを制御するアルゴリズムを考案して対応した。その結果、ユーザによる評価実験の際に問題なく、どのようなデバイスにも通知情報を送信できるようになった。最後に、ユーザによ

る支援システムの評価を岐阜県本巣郡北方町の役場で実施した。その結果、スマートフォンの通知機能を活用した市町村防災行政無線の支援が有効的である事を明らかにできた。ところが、受信する側のスマートフォンアプリケーションの画面構成や画面遷移に関する課題、操作性の課題、通知情報の表現手法に関する課題が残った。

本研究は、ICTを活用した市町村防災行政無線の同報系防災行政無線を支援する新しい仕組みの実現にあたり、スマートフォンアプリケーションを支援する新しいクラウドコンピューティング基盤を活用し、多岐にわたるオープンAPIを駆使した。ゆえに、多種多様な組織が展開する各種サービスを結集したネットワーク経営が形成されているからこそ実現できたシステムである。

〔参考文献〕

- [1] 蓮井昭夫(他)『安曇野市におけるモバイル WiMAX システムを用いたネットワーク & サービス実証実験結果-2』電子情報通信学会技術研究報告, 110(369),257-262,2011年1月13日
- [2] 小野寺昌隆, 山崎規史, 阿部健太『市町村同報系防災行政無線システム』NEC 技報 66(1), 46-49,2013年8月
- [3] 伊村真『無線通信システム技術の変遷と今後の展望』三菱電機技報 88(9), 552-555, 2014年9月
- [4] 山口良昌, 大泉明弘, 立川徹(他)『音声データハイディングを用いた防災無線システムのための携帯端末用アプリケーションの試作』電子情報通信学会技術研究報告, 信学技報 113(480), 27-32,2014年3月7日
- [5] 大野美憂, 佐々木 喜一郎, 安田 孝美『スマートフォンを活用した防災無線支援サービスの検討』第77回全国大会講演論文集, 733-734, 2015年3月17日
- [6] 『Gartner Says Worldwide Smartphone Sales Reached Its Lowest Growth Rate With 3.7 Per Cent Increase in Fourth Quarter of 2008』Gartner Press Releases, 2009年3月11日
- [7] 『Gartner Says Worldwide Smartphone Sales Grew 16 Per Cent in Second Quarter of 2008』Gartner Press Releases, 2008年9月8日
- [8] 『Gartner Says Worldwide Smartphone Sales Reached Its Lowest Growth Rate With 11.5 Per Cent Increase in Third Quarter of 2008』Gartner Press Releases, 2008年12月4日
- [9] 『Gartner Says Worldwide Smartphone Sales Reached Its Lowest Growth Rate With 3.7 Per Cent Increase in Fourth Quarter of 2008』Gartner Press Releases, 2009年3月11日
- [10] 『Gartner Says Worldwide Smartphone Sales Reached Its Lowest Growth Rate With 3.7 Per Cent Increase in Fourth Quarter of 2008』Gartner Press Releases, 2009年3月11日
- [11] 『Gartner Says Worldwide Smartphone Sales Grew 16 Per Cent in Second Quarter of 2008』Gartner Press Releases, 2008年9月8日
- [12] 『Gartner Says Worldwide Smartphone Sales Reached Its Lowest Growth Rate With 11.5 Per Cent Increase in Third Quarter of 2008』Gartner Press Releases, 2008年12月4日
- [13] 『Gartner Says Worldwide Smartphone Sales Reached Its Lowest Growth Rate With 3.7 Per Cent Increase in Fourth Quarter of 2008』Gartner Press Releases, 2009年3月11日
- [14] 『Gartner Says Worldwide Mobile Phone Sales Grew 17 Per Cent in First Quarter 2010』Gartner Press Releases, 2010年5月19日
- [15] 『Gartner Says Worldwide Mobile Device Sales Grew 13.8 Percent in Second Quarter of 2010, But Competition Drove Prices Down』Gartner Press Releases, 2010年8月12日
- [16] 『Gartner Says Worldwide Mobile Phone Sales Grew 35 Percent in Third Quarter 2010; Smartphone Sales Increased 96 Percent』Gartner Press Releases, 2010年11月10日

- [17] 『Gartner Says Worldwide Mobile Device Sales to End Users Reached 1.6 Billion Units in 2010; Smartphone Sales Grew 72 Percent in 2010』 Gartner Press Releases, 2011年2月9日
- [18] 『Gartner Says 428 Million Mobile Communication Devices Sold Worldwide in First Quarter 2011, a 19 Percent Increase Year-on-Year』 Gartner Press Releases, 2011年5月19日
- [19] 『Gartner Says Sales of Mobile Devices in Second Quarter of 2011 Grew 16.5 Percent Year-on-Year; Smartphone Sales Grew 74 Percent』 Gartner Press Releases, 2011年8月11日
- [20] 『Gartner Says Sales of Mobile Devices Grew 5.6 Percent in Third Quarter of 2011; Smartphone Sales Increased 42 Percent』 Gartner Press Releases, 2011年11月15日
- [21] 『Gartner Says Worldwide Smartphone Sales Soared in Fourth Quarter of 2011 With 47 Percent Growth』 Gartner Press Releases, 2012年2月15日
- [22] 『Gartner Says 428 Million Mobile Communication Devices Sold Worldwide in First Quarter 2011, a 19 Percent Increase Year-on-Year』 Gartner Press Releases, 2011年5月19日
- [23] 『Gartner Says Sales of Mobile Devices in Second Quarter of 2011 Grew 16.5 Percent Year-on-Year; Smartphone Sales Grew 74 Percent』 Gartner Press Releases, 2011年8月11日
- [24] 『Gartner Says Sales of Mobile Devices Grew 5.6 Percent in Third Quarter of 2011; Smartphone Sales Increased 42 Percent』 Gartner Press Releases, 2011年11月15日
- [25] 『Gartner Says Worldwide Smartphone Sales Soared in Fourth Quarter of 2011 With 47 Percent Growth』 Gartner Press Releases, 2012年2月15日
- [26] Android Developers 『Android 2.2 Platform Highlights』
<https://developer.android.com/about/versions/android-2.2-highlights.html#camera-and-gallery>, 2011年5月10日
- [27] 神原健一, 高橋和秀, 山田 暁通 『iPhone/Android 「通知機能」 プログラミング徹底ガイド』 マイナビ, 2013年12月7日

