緊急時判断に基づく 状況に応じた個人情報へのアクセス制御

長谷川友香† 小口 正人†

† お茶の水女子大学 〒 112–8610 東京都文京区大塚 2-1-1
E-mail: †yuka@ogl.is.ocha.ac.jp, ††oguchi@computer.org

あらまし 緊急災害時に家族間で個人情報を共有できれば，安否の確認や捜索などの面で大いに助けになる．しかし，平常時から当人の許可なしに個人情報を閲覧できるのは，プライバシの観点から問題である．本研究では実世界の状況に合わせて情報へのアクセス権を変化させることのできるアクセス制御手法を提案する．提案手法では，実世界において平常時か緊急時かの判断を行うためにシステム外部から情報を取り入れ，さらにユーザの判断をシステムに反映させる．そのようなアクセス制御手法を取り入れたシステムの例として，家族間情報共有システム FISS(Family Information Sharing System) を構築したため，これを紹介する．

キーワード 情報共有，災害，階層型認証，クラウド，Android，ライフログ

An Access Control Method which depends on the Situation based on a Judgement of Emergency

Yuka HASEGAWA† and Masato OGUCHI†

† Ochanomizu University
2-1-1 Otsuka, Bunkyo, Tokyo, 112–8610 Japan
E-mail: †yuka@ogl.is.ocha.ac.jp, ††oguchi@computer.org

1. はじめに

東日本大震災のような災害発生時には，多数のユーザが同時にネットワークを利用するため，回線が輻輳を起こし，急激に電話やメールがつながりにくくなるという事態が発生する．このような場合には，電話やメールなどを用いた家族の安否確認は非常に困難である．被災地住民に対する調査 [2] によると，東日本大震災の被災地で電話が使えず困った人は調査対象全体の 8 割ほど存在し，そのうちで家族・親戚・友人の安否確認がとれず不安に感じた人の割合が 8 割以上であった．このことから電話通信がほぼ不可能な状況において安否の手掛かりになるような情報が入手できれば，不安が軽減されることが予想される．さらに精神面だけでなく，行方不明となった人の捜索の面でも役立つといえるであろう．

そこで，平常時から安否確認の手掛かりとなるような情報であるライフログを貯めておき，緊急時にそれらの情報を見ることのできるシステムがあれば有用であると考えられる．そのような仕組みがあれば災害の発生した地域にいるユーザがメールや電話などに反応できない場合であっても，また災害発生地域の回線が混雑していてつながりにくい場合でも，全く別の場所にあるサーバからデータを取得し，家族の状況を把握する手掛かりが得られる．

しかし，このシステムにおいて考慮すべき点として，緊急時にのみ個人情報へアクセスできるようにしなければいけない．そのためには，実世界の状況に応じた情報へのアクセス制御手法が必要である．本論文ではそのような制御手法を取り入れたシステムの一例として，複数の情報を用いて緊急時判断を行い，その結果に応じてアクセス制御を行うシステムを提案する．

本論文では，まずシステムの概要について紹介し，システムの持つ機能について説明したのち，それらの機能がどのようにアクセス制御に用いられるかを示す．そののち各機能の実装方法についてそれぞれ説明していく．

2. 家族間情報共有システム

2.1 概 要

本研究では，家族間での緊急時における情報共有を目的としたシステムである FISS(Family Information Sharing System) を構築した．そのシステム構成を図 1 に示す．

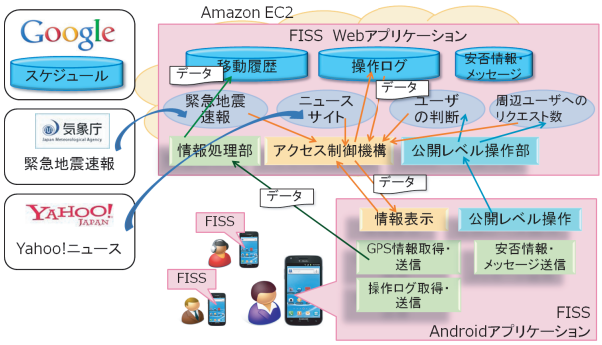

図 1　システム構成図

FISS はサーバ側とクライアント側に分かれており，サーバ側は汎用クラウドサービスである Amazon EC2 上に，クライアント側は Android 端末で動くアプリケーションとして構築している．ユーザは Android アプリケーションからシステムを利用する．

2.2　システムの機能

システムの主な機能として以下の 4 つを取り上げ，それぞれの概要について説明する．

2.2.1　個人情報の蓄積

FISS の第一の機能は個人情報の蓄積である．この個人情報とは，家族と連絡が取れなくなった際に家族の居場所を特定したり，安否を確認したりするために有用な情報を指す．そのような情報としてさまざまなものが考えられるが，例として移動履歴，Android 端末の操作ログ，スケジュールの情報を用いることとした．各情報が，災害発生時にどこにいたか，いつまで携帯端末を操作できる状態にあったか，誰と何をしていたかの手掛かりになると考えられるためである．さらに，災害が起きたと判断された際にユーザに対して安否確認を行い，その際にユーザから送信される安否情報やメッセージを保存する．この安否確認にユーザが反応したかどうかも内部情報として緊急時判断に用いる．

移動履歴は Android 端末の GPS 機能を用いて定期的に取得し，サーバ側に送信して蓄積する．操作ログは Android 端末で操作が行われるとどのような操作がなされたかを取得し，定期的にサーバに送信して蓄積する．スケジュールは Google カレンダーの情報を，Google 社の提供する「Calendar API」[4] を用いて取得する．各ユーザは OAuth 認証を用いて Google カレンダーへのアクセス権をあらかじめシステムに委譲しておく．そうすることによってユーザ ID とパスワードをシステム側で保存しておく必要なしに，システムがデータを取り出せるようになる．

2.2.2　外部情報の取得

第二の機能は，緊急時であるという判断材料となる外部からの情報を取得することである．詳細は後述するが，緊急時であるという判断には複数の情報を用い，より信頼性の高い情報のアクセス管理を実現する．緊急時判断の材料となる情報は，これも多く考えられるが，例として緊急時になりうる状況の一つである地震に着目し，緊急地震速報とニュースサイトの記事を取り込むこととした．これらは公開されている API を用いて取得している．

2.2.3　内部情報の処理

第三の機能は，ユーザの Android アプリケーションの操作によって得られる，システム内部の情報を緊急時判断とアクセス制御に用いるために処理することである．ここでいう「ユーザの Android アプリケーションの操作」とは，自分の家族内のあるユーザの個人情報を見たいというリクエストを送ることである．このリクエストも緊急時判断の情報源となりうるが，リクエストがあったら常に緊急時であると判断するように用いると個人の独断で情報を見ることのできるシステムになってしまう．そのような状況を避けるために，この情報は直接ではなく何らかの制御方式に従って用いるべきであると考えられる．本論文ではその制御方式として「階層型相互認証」と「周辺ユーザへのリクエスト割合による判断」の二種類を提案する．FISS 中ではそれらを併用することとした．

2.2.4　緊急時判断とアクセス制御

第四の機能は，第二の機能で取り上げた外部情報と第一，第三の機能で取り上げた内部情報を用いて緊急時判断を行い，アクセス制御を行うことである．ここでいうアクセス制御には，情報を見せるか見せないかという制御だけでなく，どの情報を見せるかという制御も含まれる．具体的には，外部情報である緊急地震速報とニュース記事，また内部情報のうち「周辺ユーザへのリクエスト割合」,「安否情報の有無」を用いて緊急時判断を行い,「階層型相互認証」を用いてどの情報を見せるか決めるアクセス制御を行う．

3.　緊急時判断とアクセス制御

本章では緊急時判断とアクセス制御の詳細について説明する．処理の流れを図 2 に示す．

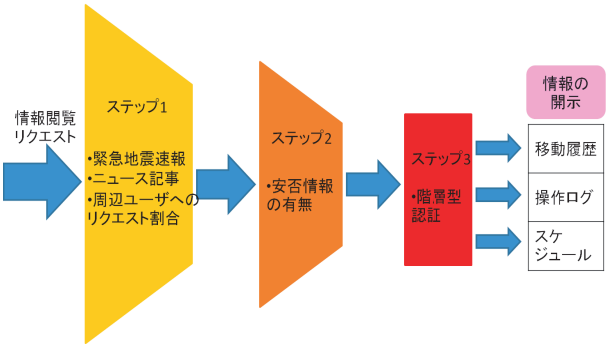

図 2　緊急時判断とアクセス制御の流れ

あるユーザの情報を見たいというリクエストがあったときにまずステップ 1 として緊急地震速報，ニュース記事あるいは周辺ユーザへ情報公開リクエストが送られた割合を用いて情報閲覧をリクエストされているユーザが危険地域にいるかを判断する．ここでいう危険地域とは災害の被害が及ぶと推定される地域のことである．ステップ 2 として，ユーザが危険地域に居た場合はそのユーザから安否情報が通知されているか調べる．安否情報が通知されていない場合に，そのユーザは危険地域に

おり，かつ安否がわからないという可能性が高いので，家族はそのユーザの情報を見ることができるものとする．しかし，あくまでも可能性が高いというだけであり，実際には何の被害も被っておらず，ただ携帯端末を見ていなかったということも十分にあり得る．それゆえ，該当ユーザの個人情報を一気に全て公開してしまうことは適切でないと考えられる．そこでステップ 3 として階層型相互認証を用いる．この階層型相互認証を用いることで先に述べたような状況で情報が見られることを防ぎ，さらに段階的な情報の公開が可能となる．

4. 外部情報の利用

外部情報の取得からユーザへの安否確認の送信までの流れを図 3 に示す．

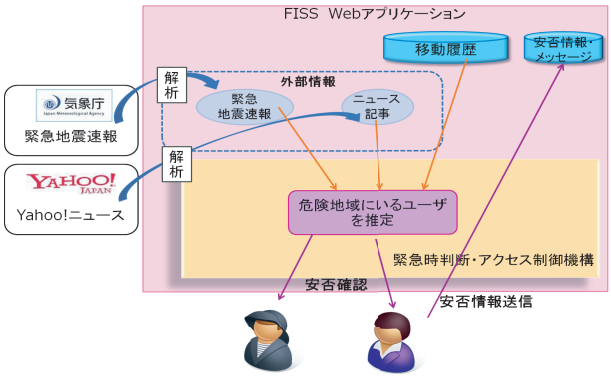

図 3　外部情報の取得とユーザの安否確認

緊急地震速報は揺れが感知された時点で発せられ，実際に地震の波が届く前に警告を与えることを目的としている．それに対し，ニュース記事は実際に観測された地震とその被害状況に基づいて発信される．FISS では災害の被害の及ぶ可能性にあるユーザに対して Android アプリケーション上で安否確認を行うこととしており，できるだけ早く危険地域にいるユーザを見つけたい．しかし，緊急地震速報の発表されない場合や，発表されたものよりも実際の地震の方が規模が大きいことは十分にあり得る．そこでニュース記事の情報も併せて用いることで，危険地域にいるユーザの検出漏れを極力減らす．

4.1 緊急地震速報

気象庁から直接緊急地震速報を取得するには「予報業務許可」が必要であり，法人でなければならない．そのため Twitter 上で発表される緊急地震速報のツイートを取得することで間接的に緊急地震速報を利用した．Twitter 社の提供する「Streaming API」[5] を利用し，緊急地震速報を流しているアカウントのツイートをリアルタイムで取得する．そのツイート内容から地震の発生位置や規模を解析し，危険地域にいると判断したユーザに対して安否確認行う．

緊急地震速報を Twitter 上で流しているアカウントは多数あるが，今回は一例として「@eewbot」というアカウントのツイートを用いた．このアカウントは高度利用者向け地震速報を csv 形式で配信しており，ここから地震発生位置（緯度，経度，深さ）と地震規模（マグニチュード）を取得する．FISS 内部に保存してある各ユーザの現在地を用いてユーザ位置での大まかな震度を予測する．予測のための計算式は [8] や [9] に紹介されているものを用いた．

4.2 ニュース記事

さまざまなニュースサイトが存在するが，一例として Yahoo!ニュースを取り上げ，その記事が発行される様子を監視することとした．Yahoo!社が提供する「トピックス見出しアーカイブ API」[6] を利用し，地震に関連するニュース記事が発行されたかどうかを一定の間隔で確認する．発行された場合はニュース記事の見出しから，地震の発生地域と震度情報を取得する．ユーザの現在地は緯度経度情報で保存されているので，発生地域を緯度経度に換算する必要がある．そこで Google 社の提供する「Geocoding API」[7] を利用する．この API は地名をパラメタとして GET リクエストを送ると，該当する地域の範囲を緯度経度で返すものである．ただし，「関東」や「東北」などの地方名には対応していないため，それらの場合は FISS 内部で変換を行う．変換後の範囲内にいるユーザに対して緊急地震速報の場合と同様にユーザに安否確認を行う．

ここで注意すべきなのはニュース記事の取得はリアルタイムではないということである．こちら側からリクエストを送って定期的に記事発行の有無を確認するという方式であり，記事が発行された瞬間に緊急時判断を行うことは不可能である．

4.3 外部情報による緊急時検知

前節までに述べた方法を用いて取得した約 4 週間分のデータを図 4 に示す．このグラフは，Twitter を通して取得した緊急地震速報と Yahoo!ニュース記事の関係を示すものである．

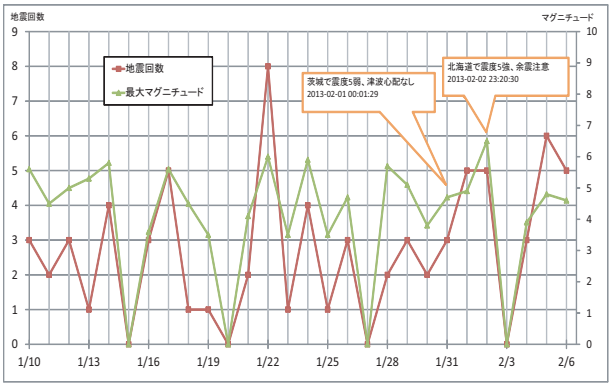

図 4　緊急地震速報と Yahoo!ニュース記事の関係

ここで，「最大マグニチュード」とはその日に発生した地震の中で最もマグニチュードの大きかった地震のその値を指す．緊急地震速報が発表された地震はデータ取得期間中に 76 回であったのに対し，Yahoo!ニュースで地震の記事が発行されたのは 2 回であった．Yahoo!ニュース記事は 2 月 1 日と 2 日に一つずつ発行されている．2 月 2 日に発生した地震は Yahoo!ニュースによると北海道で震度 5 強という比較的大きなもので，緊急地震速報によるとマグニチュードは 6.5 であった．

2 月 1 日のニュース記事に注目すると，茨城県で震度 5 弱の地震が発生したことが伝えられているが，この地震は日をまたぐ直前に発生したものであり，実際に発生した日付は 1 月 31

日である．緊急地震速報からサーバがこの地震を検知した時刻は 1 月 31 日 23 時 53 分であり，ニュース記事が発行される 8 分ほど前である．先ほど取り上げた北海道での地震における緊急地震速報と記事発行の時間間隔は 2 分間程であった．ニュース記事は確認された事実が書かれているため正確な震度や被害状況を知ることが出来るが，その分即時性は下がり，地震発生後どれくらいで記事が発行されるかは状況により異なるということが分かる．

ニュース記事が出ていない他の日でマグニチュードが大きい地震が起こっている様子が見られるが，これらの地震は陸地から離れたところで発生しており，人の住む地域に大きな影響がなかったため記事にならなかったと言える．

次に，ユーザが日本の各地にいると仮定してそれらのユーザが危険地域にいると判断されたかどうかを示す．仮想的なユーザの配置場所を図 5 に示す．

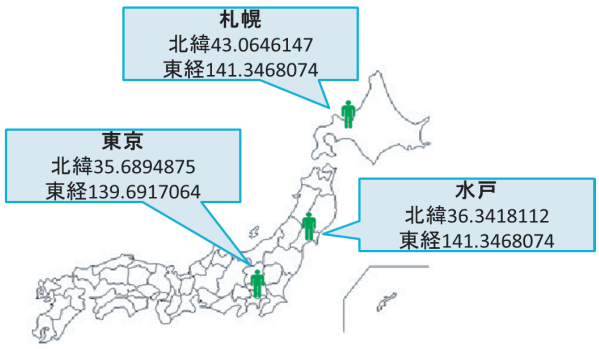

図 5　ユーザの配置

ユーザが札幌，茨城，東京の県庁所在地にいるものと仮定した．

図 4 で取り上げた北海道の地震の際の結果を表 1 に示す．これは北緯 42.6 度，東経 143.3 度，深さ 120km の地点で発生したマグニチュード 6.5 の地震である．気象庁が後日発表した各地の震度を比較のために表に加えた．

表 1　2 月 2 日に発生した北海道での地震によるユーザへの影響

ユーザの場所	緊急地震速報から算出の震度	Yahoo!ニュース	気象庁発表の震度
札幌	3.3 ± 0.7	地域内（震度 5 強）	3
水戸	0.5 ± 0.7	地域外	2
東京	0.1 ± 0.7	地域外	1

同様に茨城での地震の際の結果を表 2 に示す．これは北緯 36.7 度，東経 140.6 度，深さ 10km の地点で発生したマグニチュード 4.7 の地震である．

表 2　1 月 31 日に発生した茨城での地震によるユーザへの影響

ユーザの場所	緊急地震速報から算出の震度	Yahoo!ニュース	気象庁発表の震度
札幌	-1.8 ± 0.7	地域外	なし
水戸	2.5 ± 0.7	地域内（震度 5 弱）	2
東京	1.3 ± 0.7	地域外	なし

緊急地震速報を用いたユーザ位置での震度予測では地震発生場所に近いユーザほど大きな震度が算出されており，妥当な結果であると言える．ニュース記事からも地震の発生地域にいるユーザを危険であると正しく判断できた．

ニュース記事に記載されている震度はその地域で最大のものであり，例えば北海道の地震ならば十勝での震度である．札幌で実際に観測された震度は 3 であり，この場合は緊急地震速報から得たデータで適切に予測が出来ている．茨城の地震の場合は同じ県内でありながら震度 5 弱と 2 の地域が存在するものであったが，これは震源が 10km の深さと比較的浅いところで発生した地震であるため，揺れが広範囲に広がらず，局所的に大きな揺れが発生したことによる．

システムを実運用する際には，算出した震度に対して閾値を設けて，それより大きな震度が予測されたらユーザに安否確認をするべきであるが，算出した震度には幅があり，精度はあまり良くない．算出される値と実施の震度が大きく違う可能性もあるので，閾値は低めにとり，本当の緊急時を取りこぼさないようにすべきだと考えられる．

以上から，緊急地震速報を用いた震度推定と Yahoo!ニュース記事の解析はそれぞれ考慮すべき点はあるものの地震から起因する緊急時の判断に用いることのできる手法であると言える．さらに，それぞれに利点と欠点が存在するため，単独ではなく組み合わせて用いることが有用であると推察される．

5.　内部情報の処理

2 章で述べたように，あるユーザの情報を見たいというリクエストは「周辺ユーザへのリクエスト割合による判断」,「階層型認証」の二つの方式で処理して緊急時判断・アクセス制御に用いる．

5.1　周辺ユーザへのリクエスト割合による判断

この方式は，一人の判断では信用度が低いが，同一の場面で多くの人が同様の判断をしたら信頼度は高いものという考え方に基づいている．この概要図を図 6 に示す．情報を公開するようリクエストが送られたユーザ A の周りにいるユーザを探索し，それらのユーザにも同様にリクエストが送られているか調べる．

ユーザ A の周りのユーザにもリクエストが送られている場合は，その地域は災害の被害を受ける可能性が高いとそれぞれのユーザの家族が判断しているということである．そこでユーザ A は危険地域にいると判断する．

この方式のメリットは，当事者の判断ではなく第三者の判断のみを参考にするため，客観的な基準で情報の公開・非公開が決まるということである．そのため，ユーザ個々の判断を情報源として使用しているものの，一人のユーザが家族の情報を見ようとした場合に簡単に見ることができるという状況は防ぐことができる．

この方式ではどのくらいの数のユーザを探索し，どのくらいの割合でリクエストが来ていたら危険地域にいるという判断をすべきか検討する必要がある．これらの値は人口密度などによって地域ごとに変化すると考えられる．

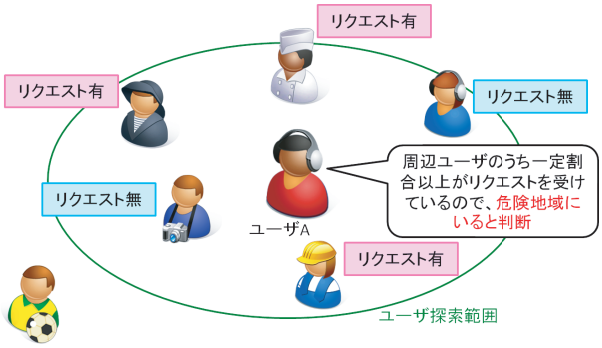

図 6　周辺ユーザへのリクエストを用いた判断の概要

5.2　階層型相互認証

本論文では階層型相互認証とは，ユーザがそれぞれある階層にいるとし，同じ階層にいるユーザ同士が認証されたとして情報を互いに閲覧できる認証モデルである．その概要図を図 7 に示す．

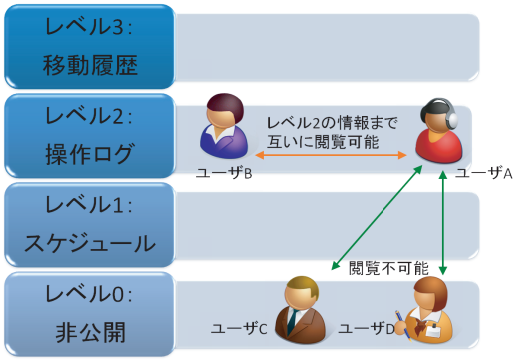

図 7　階層型相互認証の概要

各階層には個人情報が割り当てられており，同階層のユーザでその情報を互いに閲覧できる．通常ユーザはレベル 0 の非公開層にいると考える．緊急時のユーザの階層移動を，ユーザ B がユーザ A の操作ログを見ようとすることを例にとって考える．ユーザ B はユーザ A をレベル 2 の層まで上げる．するとユーザ B も自動的にレベル 2 の層まで上がる．これによりレベル 2 で同じ層であるので相互に操作ログの情報を見られるようになる．

ここで，相手のレベルを上位層に上げると自分も強制的に上位層に上がるので，これによりむやみに高い階層の情報を見ないよう抑止力として働くと考えられる．相手の情報を見るためには自分も同じレベルの情報を相手に公開しなければならないということである．

6.　Android アプリケーション

ここまで説明してきた制御方法等は全てサーバ側の実装である．本章では UI 部分である Android アプリケーションの実装を紹介する．

図 8(a) がアプリケーションのメイン画面である．ユーザはここから自分の各種情報を確認できる．

(a) 自分のレベルを変更中　　(b) 自分のレベルを変更後

(c) 相手のレベルを変更中　　(d) 相手のレベルを変更後

図 8　レベル変更画面

図 8(b) はユーザに対して安否確認を行う画面である．サーバ側で危険地域にいると判断されたユーザに対して，PUSH 通知によってサーバから Android 端末にその情報を通知する．PUSH 通知とはサーバ側からクライアント側へデータを送信する仕組みのことである．通常ならばクライアントからのリクエストがあった時にサーバからデータを返すという流れになるが，この PUSH 通知を用いることで，サーバから任意のタイミングでクライアントにデータを送信することができる．Android でこの仕組みを利用するために Google 社の提供する「Google Cloud Messaging for Android」というサービスを利用した．

図 8(c) は安否確認済みのユーザの情報画面である．このユーザは安否確認を済ませているので，安否情報とメッセージが表

示される．安否情報がわかるため，このユーザは緊急状態にはないと判断され，個人情報を見るために階層レベルの変更をすることは不可能である．

図**??**は安否確認が取れていないユーザの階層レベルを変更した画面である．このとき自分も同じレベルの階層に上がり，他の人から情報を見られる可能性のある状態になる．この画面の状態に至るのは，該当ユーザが危険地域にいると判断され，そのユーザから安否確認が取れず，さらにあるユーザが自分の情報を他の人に見せてでもそのユーザの情報を見なければいけないと判断した場合のみである．そのようなアクセス制御を行ったうえで，非常事態に巻き込まれた可能性の高いユーザの情報を，その家族が見ることのできる仕組みが提供されている．

7. まとめと今後の課題

実世界の状況に合わせてアクセス権の変化するアクセス制御手法を提案し，その手法に基づいて個人情報を管理する家族間情報共有システム (FISS) を構築した．これにより，平常時には個人情報は誰からも見られないが，緊急時のみ家族から情報を見られるシステムを実現した．

今後の課題としては，外部情報から実際に地震を検知し，ユーザを危険地域にいると判断する機能のテストを行う．その際，どのくらいの震度で危険地域にいると判断すべきかという点や，地震発生後どれくらいの時間で判断ができるかという点などを考察する．さらに，地震以外の災害を検知する方法を考察する．

また今後は周辺ユーザのリクエスト割合によって緊急時を検知する方法において，シミュレータを作成して，どのくらいの割合で緊急時とするのが最適かという点やどのくらいの周辺ユーザを探索するべきかという点などを考察する．人口密度の違いなどによって最適なパラメタがどのように変化するか調査する．また，周辺ユーザ探索において，速度を向上させる方法を検討する．

文　献

[1] Takeshi Sakaki, Makoto Ozaki, and Yutaka Matsuo: "Earthquake Shakes Twitter Users: Real-time Event Detection by Social Sensors ", Proc.of the 19th international conference on World Wide Web 2010

[2] 千葉直子，山本太郎，関良明，高橋克己，小笠原盛浩，関谷直也，中村功，橋元良明: "被災地住民の情報通信利用の実態と心理－東日本大震災の被災地住民への訪問留置調査－ ", DICOMO 2012，1G-4，2012 年 7 月．

[3] 山本太郎，千葉直子，関良明，高橋克己，小笠原盛浩，関谷直也，中村功，橋元良明，"被災地住民のインターネット利用における安心と不安－東日本大震災の被災地住民への訪問留置調査－ "，DICOMO 2012，1G-5，2012 年 7 月．

[4] Google Calenar API, http://developers.google.com/google-apps/calendar/

[5] Twitter Streaming API, http://dev.twitter.com/docs/streaming-apis

[6] Yahoo! トピックス見出しアーカイブ API, http://developer.yahoo.co.jp/webapi/news/news/v1/heading.html

[7] Google Geocoding API, http://developers.google.com/maps/documentation/geocoding/

[8] 気象庁地震火山部，"緊急地震速報の概要や処理手法に関する技術的参考資料 "，http://www.seisvol.kishou.go.jp/eq/EEW/kaisetsu/Whats_EEW/reference.pdf

[9] 独立行政法人防災科学技術研究所，"地震動の評価モデル "，http://www.j-map.bosai.go.jp/j-map/result/tn_275/html/html/2-7.html