緊急災害時に有用な家族間の個人情報共有システムの提案と実装

長谷川 友香　(指導教員：小口 正人)

1　はじめに

東日本大震災のような大きな災害に遭った場合，家族の安否を第一に確認したいと思うであろう．ところが，このような災害時には電話回線が混雑し，家族と直接連絡をとれないことが多い．

そのような状況において，スケジュール情報や移動履歴，メールなどの個人情報を得ることができれば，家族が災害発生時にどこにいたかを特定する大きな手がかりとなり有用である．しかし，家族とはいえ平常時から個人情報が閲覧できてしまうことには抵抗があると思われる．さらに，大規模な災害ではなくても，子供と連絡がつかない場合にはスケジュールだけ閲覧できるようにするなど，緊急の度合いに合わせて閲覧できる情報を制御したいという要望が想定される．

そこで，本研究では家族間で個人情報を共有するシステムを構築し，緊急時のみ閲覧を許可する認証方法を実装する．その認証方法としては階層型相互認証を提案する．階層型であるため，情報の閲覧を許可するかしないかの切り替えではなく，どのレベルの情報まで許可するかを制御可能となる．

2　階層型相互認証

本研究における階層型相互認証とは，ユーザ各々に認証レベルを付与して，ある取り決めに従って認証レベルにより情報の公開制御を行うものである．本研究では認証レベルをユーザの間でそれぞれに対して保持されるものとした．これらの認証レベルは互いに独立しており，例えばユーザ A のユーザ B に対する認証レベルの変更はユーザ C の認証レベルには影響しない．

この認証レベルに個人情報を段階的に割り当てる．本研究では一例として，図 1 にあるようにレベル 0 は閲覧不可，レベル 1，2，3 はそれぞれスケジュール，移動履歴，メールまで閲覧可能という割り当てを行った．

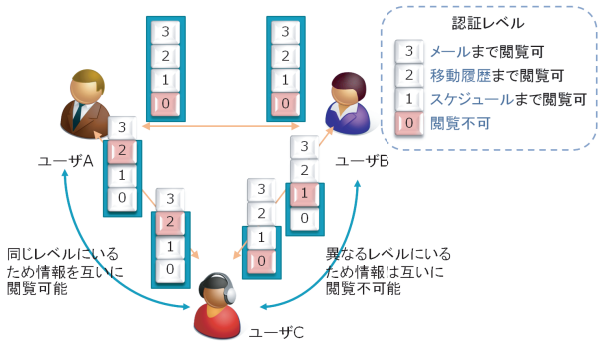


図 1: 階層型相互認証の概要図

自分の認証レベルは基本的に変更可能であり，それによって相手に対して公開する自分の情報を制御することができる．例えばユーザ A とユーザ B が互いに対する認証レベルを 1 とすると，スケジュール情報が互いに閲覧可能となる．何も公開したくない場合には相手に対する認証レベルを 0 とすることで互いに何の情報も閲覧できない状態になる．

このように平常時の情報公開の制御は当人が行えば良いが，緊急時には当人が制御するのは現実的でない場合がある．そこで，緊急時には認証レベルに関して次のような操作を行う．まず，情報を閲覧するユーザ（以下，閲覧ユーザ）が閲覧対象のユーザ（以下，被閲覧ユーザ）の認証レベルを任意のレベルまで上げる．そうすると閲覧ユーザの認証レベルも自動的に同じレベルまで上がり，被閲覧ユーザに対して認証レベル変更を通知するメールが送られる．こうすることで両ユーザの認証レベルが同じになり，情報の閲覧が可能となる．閲覧ユーザの認証レベルが自動的に上がることと，被閲覧ユーザに通知が送られることが通常時にこの操作を行うことへの抑止力になると考えられる．

ユーザ間で上げられる最大の認証レベルを設定可能とした．また，全ての情報を閲覧できてしまう特権ユーザというものが家族内に存在すると，情報を閲覧されたくないという気持ちから，システム自体が使用されなくなる可能性が高いと考えられるため，特権ユーザは存在しないこととした．

3　認証の取り決め

本研究では認証の基本的な取り決めとして以下のものを採用している．

基本的な取り決め
- 相手のレベルを上げると自分も同じレベルまで上がる
- 設定した最大値より上には上げられない
- 自分と相手のレベルの低いほうに該当する情報まで互いに閲覧可能

さらにユーザは，相手の認証レベルを変更していない状態，相手によって自分のレベルを変更された状態，自分が相手のレベルを変更した状態をの 3 状態を持ち，その状態に従って可能な操作が決まる．自分から見た状態遷移の様子を図 2 に示す．

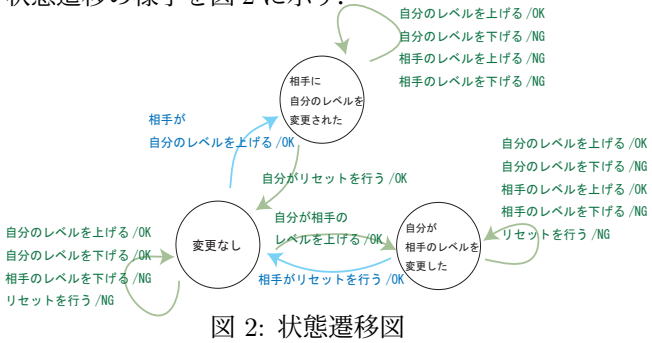


図 2: 状態遷移図

閲覧ユーザが被閲覧ユーザのレベルを上げると両ユーザともレベルを下げられないことになるが，被閲覧ユーザがリセット操作をすることにより変更なし状態に戻り，再び自分のレベルの設定を自由に行えるようになる．この遷移図は自分から見た場合であり，相手側の制御を考える場合には相手と自分を読みかえればよい．

4　開発環境

本研究では Google App Engine と Android 端末を使用してシステムを構築する．システム構成図を図 3 に示す．

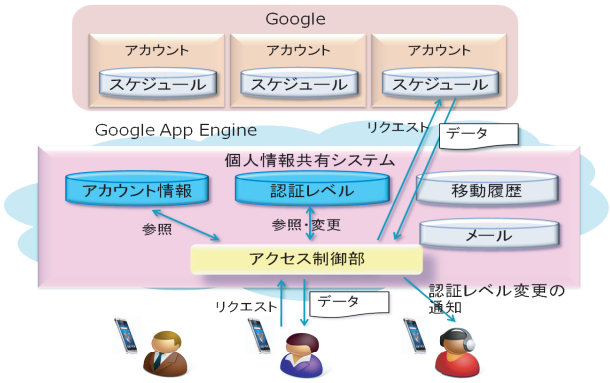


図 3: システム構成

4.1　Google App Engine

Google App Engine とは Google 社が提供するクラウドサービスで，PaaS 形式のアプリケーションプラットフォームである．Python や Java，Go 言語を使用しての開発が可能であり，本研究では Java を用いて開発した．Google App Engine にユーザ情報や認証レベル，移動履歴情報などを保存し，階層型認証の制御を行う．ユーザは Google App Engine 上に作成した Web サイトから情報の閲覧ができる．

4.2　Android

Android は，Google 社を中心とするモバイル機器の共通プラットフォームを推進する組織「OHA（Open Handset Alliance）」によって開発が行われている携帯情報端末向けのプラットフォームである．本システムではこの Android を搭載したスマートフォンを用いて移動履歴の取得と情報の閲覧を行う．

5　個人情報へのアクセス

本研究で扱う個人情報はスケジュール，移動履歴，メールの 3 種類である．通常の利用であればこれらの情報にアクセスするのは本人だけであるが，情報共有のためにはシステムからアクセスする手法が必要になる．それぞれの情報へのアクセスの実装方法を以下に示す．

5.1　スケジュール

スケジュール情報としては，Google Data API を用いて Google カレンダーの情報を取得する．Google Data API とは REST 形式でカレンダーやマップなどの Google のリソースを操作できる API である．本システムでは，この API を用いるに当たり OAuth 認証を行う．OAuth 認証とは，あらかじめ信頼関係を構築したサービス間でユーザの同意のもとにセキュアにユーザの権限を受け渡しする「認可情報の委譲」のための仕様である．本システムの初回利用時にユーザがシステムに対して Google カレンダーへのアクセス権限を委譲し，システムはそれ以降その権限使って個人のカレンダー情報を取得する．

5.2　移動履歴

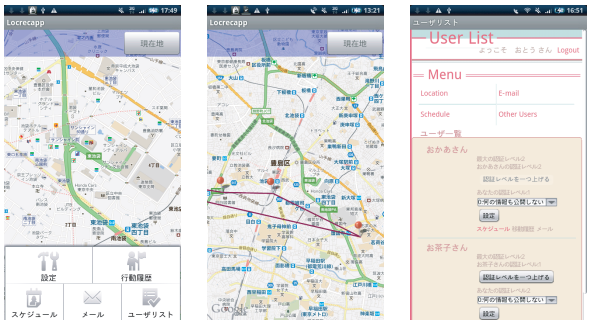
移動履歴は Android 端末で GPS を使って取得する．具体的には，Android 端末で動くアプリケーションを作成し，時間と距離に関する閾値を超えると現在地の緯度と経度を取得してサーバに送信する．

5.3　メール

メールは転送設定を行い，クラウド上のアプリケーションで受信して蓄積する．本システムでは Gmail からの転送を想定して実装している．

6　システム実装

本システムは PC 上のブラウザと Android アプリケーションからアクセスが可能である．ここでは Android アプリケーションから利用する場合を説明する．自分の情報は図 4(a) で示したメニューから選んで自由に閲覧することができる．図 4(b) は移動履歴を閲覧している画面である．



(a) メニュー　(b) 移動履歴　(c) ユーザ一覧

図 4: Android アプリケーション使用画面

認証に関する操作は図 4(c) のユーザ一覧画面から行う．自分の認証レベルはリストボックスから選んで設定する．相手の認証レベルの変更は「認証レベルを一つ上げる」のボタンで行う．このボタンを押すと，相手の認証レベルが一つ上がり，自分の認証レベルも同じところまで上がる．これによりレベルに対応する相手の情報が閲覧可能になる．

7　まとめと今後の課題

緊急時に家族間で個人情報を共有するためのシステムを Google App Engine と Android を用いて構築した．プライバシの問題を考慮し，緊急時のみ当人の許可がなくても情報の閲覧を可能にするために階層型相互認証機構を実装した．

今後の改善点として，ユーザ間で対応するレベルをずらした認証モデルの採用が挙げられる．これは，ユーザ間で同じ情報を公開することで釣り合いが取れるというものではなく，例えばユーザ A がスケジュールを公開するときにユーザ B が移動履歴まで公開して当人同士で平等だと感じるかもしれないということに基づく．この機能については現在実装中である．

参考文献

[1] 長谷川友香，小口正人，階層型相互認証に基づく緊急災害時に有用な家族間の個人情報共有システム，DEIM Forum 2012，2012 年 3 月発表予定