

Twitter API を用いたセンサー情報のオープン化 Open sensor information with Twitter API

竹田拓馬[†] 武田 利浩[†] 平中 幸雄[†]
Takuma Takeda Toshihiro Taketa Yukio Hiranaka

1. はじめに

現在、センサネットワークの普及が進んでおり、様々なアプリケーションの登場が予想される。しかし、現在のシステムは、個々の目的の為に垂直統合的に設計と実装がなされ、センサ情報はシステム内で閉じている。システム内で利用されるセンサ情報は他のアプリケーションにとっても有意義な情報が多く含まれることが考えられ、センサ情報のオープン化が期待される。

そこで本研究では、現在、情報発信のインフラとしての性格を持つ Twitter 上でのセンサ情報のオープン化を行う。本稿では、Twitter 上でのデータフォーマットの提案と Twitter API を用いたオープン化の手法、及びそのシステムの実装について述べる。

2. Twitter の普及と情報インフラ化

センサ端末から取得できるセンサ情報をオープン化するためには、利用者が直接センサ端末にアクセスするのではなく、センサ情報を一度開かれた場所に置き、誰でも、どこからでも、いつでも利用者がセンサ情報を取得できるようにする必要がある。そこで WEB 上にセンサ情報を公開する。本研究では、WEB サービスである Twitter[1]を用いる事とした。Twitter は、ツイートと呼ばれる 140 文字のメッセージから成り立つ情報ネットワークで、興味のあるテーマに関する最新情報を発見するのに役に立つ WEB サービスである。友人同士の気軽なコミュニケーションツール、またはマイクロブログとして利用のされ方が一般的であるが、API が公開されており、ツイートを自動で行う Bot と呼ばれるアカウントや、Twitter のアカウント情報を用いた第三者 WEB アプリケーションなども多く存在している。これにより、Twitter には様々な情報が溢れ、情報の利用が多くなされている。そのため Twitter は利用者が有意義な情報を入手する際に便利なサービスとなっている。このような Twitter の性質から、地震速報をツイートする Bot アカウント earthquake_jp [2] に代表するような情報を公開する Bot アカウントが多数存在し、Twitter は、情報公開のインフラとなりつつある。そこで、本研究では、センサ情報をオープン化の際の手法として Twitter API を使用する。

3. オープン化の手法

3.1 データフォーマット

Twitter のツイート文にセンサ情報を提示するため、機械と人間の両方が理解しやすいセンサ情報のデータフォーマットを考える必要がある。そこで、以下のようなデータフォーマットとする。以下に例を示す。

[場所] 平中研究室学生室 [温度] 26℃ [加速度] 0.22f,0.225,0.318 [時刻] Thu Feb 10 01:22:45 JST 2011[ID]SS1_01 #SS1_01

このフォーマットでは '[' と ']' によってくくられた項目名の後にデータ内容が続く単純な記述とする。[ID] の項目は、ユーザが任意で決めるセンサ端末を識別するユニークな値を ID とする。#(ハッシュタグ)は検索を容易にするためにあるものであるため、センサ ID や、使用するアプリケーション、まとめたいセンサ端末群など、取得の際に利用しやすいように、任意で付加できるものとする。

3.2 センサ端末と Twitter の連結

図 1 にセンサ端末と Twitter の連結を示す。図示のように、センサ端末と Twitter との連携は、センサ端末にアクセスし、Twitter に自動でセンサ情報をツイートを行う「Bot」アプリケーションを作成する。Bot アプリケーションの主な機能としては、以下の 4 つの機能をつける。

- 1) センサ端末へのアクセス
- 2) 取得したセンサ情報のフォーマット変換
- 3) Twitter API を用いて Twitter へツイート
- 4) 一定時間毎の自動ツイート

図を用いて表すと以下の図 1 のようになる。

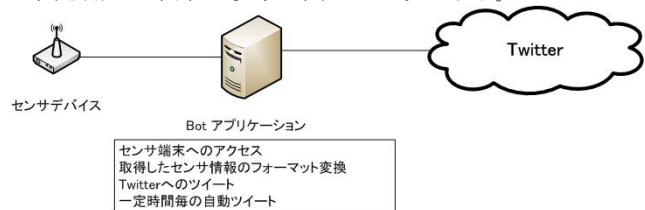


図 1 : センサ端末と Twitter の連結

3.3 Twitter からのセンサ情報の利用

Twitter 上にデータが上がるため、センサ情報の利用は二つの状態が考えられる。

- 1) センサ情報が Twitter 内で完結したタイムラインから得られる情報として利用する。
 - 2) センサ情報を Twitter から取得し、第 3 者によるアプリケーションが利用する。
- 1) に関しては、Twitter のアカウントをもつユーザーが Bot のアカウントをフォローすることで、センサ情報をモニタリングできる。「有益な情報の閲覧」という一般的な Twitter の利用である。
- 2) の利用では、センサ情報を利用するアプリケーションが、公開されている Twitter API を利用しデータを取得する。センサ情報は、Twitter 上に存在するため、ひとつの第 3 者によるアプリケーションがセンサ情報を独占することなく、複数の第 3 者によるアプリケーションがセンサ情報を共有することが可能となる。

4. 実装

4.1 システム全体図とセンサ情報のフロー

システムの全体図とセンサ情報のフローを図 2 に示す。

[†] 山形大学 Yamagata University

図示の(1)ではセンシングを行う環境に置いたセンサデバイスが PC からの要求により、センサ情報を PC へと渡す。(2)では、Bot プログラムが取得したセンサ情報を TwitterAPI を用いて Tweet する。(3)では、Twitter 上でオープン化されたセンサ情報が、TwitterAPI を用いて、アプリケーションへとセンサ情報が渡される。

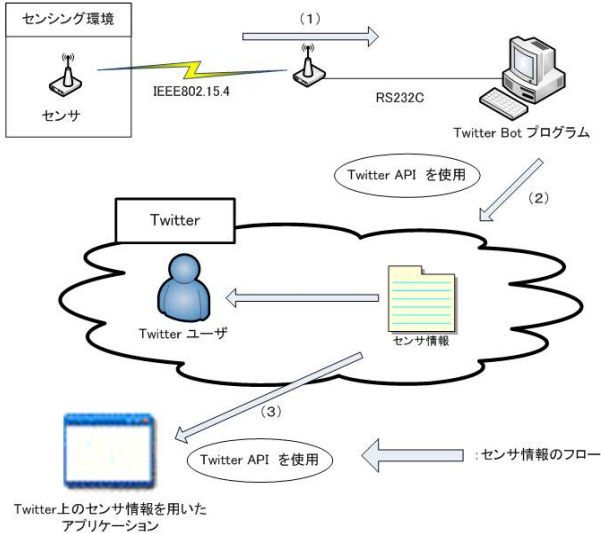


図2：システム全体図とセンサ情報のフロー

4.2 センサノード

使用するセンサデバイスは、SilentSystem社[3]のSS-1とSSIO-1を用いる。SS-1はIEEE802.15.4に準拠した2.4GHz帯無線通信モジュールで、SSIO-1はSS-1とRS-232C端子、や電源端子、3軸加速度センサ、温度センサなどを搭載したSS-1のIOボードである。SS-1とSSIO-1を組み合わせ、筐体に入れたものが図3である。SS-1にはセンシングを行うDeviceモードとPCとの接続を行うMasterモードがあり、Deviceモードは温度と加速度値をセンシングし、Masterへ送信する。MasterモードはPCからの取得リクエストを受け、センサ情報をPCへと送信する。



図3:SS-1とSSIO-1

4.3 Twitter Bot プログラム

開発言語はJavaを用いた。PCとセンサ端末SS-1(SSIO-1)とは、RS-232Cで接続するため、JavaのシリアルポートアクセスのライブラリであるRXTXライブラリ[4]を使用する。また、ツイートの際には、Twitter APIを使用するため、Twitter APIのjavaラップであるtwitter4j[5]を用いて作成した。

4.4 Twitter 上のセンサ情報を用いたアプリケーション

Twitter API を用いてセンサ情報を Twitter から取得し、

二つの環境の温度データのモニタリングするアプリケーションを作成した。開発言語はJavaを使用し、Botアプリケーション同様に、twitter4jを使用した。機能としては以下の3つの機能を持つ

- 1)Twitter API を用いてセンサ情報の取得
- 2)GUI に取得したセンサ情報を描画
- 3)更新ボタンによるイベント処理

5. 動作実験

5.1 センサ端末と Twitter の連結

図4に示すように、センサ情報を Twitter に入れ込むことができた。



図4：Botプログラムの実行結果

5.2 Twitter からのセンサ情報の利用

図5は今回作成したTwitter上のセンサ情報を用いたアプリケーションである、モニタアプリケーションであるこれにより、アプリケーションがセンサ情報をTwitterから取得できた。



図5：モニタアプリケーションの実行結果

6. まとめと課題

本研究では、Twitter上に公開するセンサ情報のフォーマットを定義し、Twitter APIを用いてセンサ情報をWebに取り込みと、取り出しの手法を考案し、センサ情報のオープン化のシステム構築と動作実験を行った。

今後の課題としては、画像や音声といったセンサ情報への対応や、アプリケーションの要求に適したセンサ情報を提供するための要求マッチングの機能の実現が必要であると考えられる。

参考文献

- [1]Twitter, <http://twitter.com>
- [2]地震速報 Bot earthquake_jp, http://twitter.cpm/earthquake_jp/
- [3]SilentSystem社, <http://www.silentsystem.jp>
- [4]RXTX, <http://users.frii.com/jarvi/rxtx/>
- [5]twitter4j, <http://twitter4j.org/ja/index.html>