社会センサデータ生成・共有基盤システムの設計と実装

中嶋 奎介[1] 谷山 雄基[1] 横山 正浩[1] 義久 智樹[2] 原 隆浩[1] 西尾 章治郎[3]

概要：近年，SNS（Social Networking Service）が広く利用されている．SNS に投稿された短文や写真から，単語の出現頻度や写真撮影地点を解析することで，実社会の情報を取得できる．例えば，Twitter のツイートを形態素解析し，地震に関する単語が多く含まれていれば地震の発生が考えられる．本研究では，SNS への投稿等を解析して得られるデータを社会センサデータと呼ぶ．社会センサデータや，社会センサデータを生成するためのプログラムを共有することで，新たな社会センサを開発するために再利用することが可能となる．そこで本研究では，社会センサデータを生成・共有するための基盤システムの設計および実装を行った．さらに本研究では，提案システムと既存システムとの機能比較を行い，提案システムは既存システムと比べて社会センサデータの生成，共有に適していることを確認した．

1. はじめに

近年，様々な人々や地域の自治体等により SNS（Social Networking Service）が広く利用されている．例えば，Twitter[*1] や Facebook[*2] のようなマイクロブログや，Flickr[*3] や Instagram[*4] のような動画像投稿・共有サービスが利用されている．これらのサービスにおけるユーザの投稿を解析することで，実社会の情報を取得できる．本研究では，SNS への投稿等を解析して実社会の情報などを得ることを，**社会センサ**によって社会をセンシングすると捉え，得られたデータを**社会センサデータ**と呼ぶ．例えば，Sakaki らは，Twitter のツイートに対して教師あり機械学習手法の SVM（Support Vector Machine）を用いて解析することで，地震に関するキーワードを含むツイートから，地震を検出する手法を提案している [1]．この解析結果は，ある地域，ある時刻に地震が発生したことを示す社会センサデータと捉えられる．上述した例のほか，SNS への投稿等を解析して，社会センサデータとみなせるデータを生成する様々な研究が行われている [2], [3], [4], [5]．これらの社会センサデータを共有することで，共有したデータを他のアプリケーションに適用したり，異なる解析を行うための入力データにするなど，様々な応用に再利用できる．例えば，杉谷らは，位置情報を持たないツイートに対して，SVM を用いて位置情報を付与している [2]．文献 [1] および [3] で言及されているように，位置情報付きツイートを利用した研究が多数存在する一方で，全世界のツイートに対する位置情報付きツイートの割合は 0.77%（2012 年 7 月 30 日現在 [*5]）と非常に低い [6]．杉谷らの研究によって生成された位置情報付きツイート（社会センサデータ）を共有することで，ツイートの位置情報とテキストからイベントを検出するなど，他の応用に再利用できる．

一方，解析プログラムなどの社会センサデータの生成に関する記述も，新たな社会センサを開発する際に有用である．例えば，上述した Sakaki らの地震検出のための関数を参考にして，観測対象を地震のような災害だけでなく，交通渋滞や有名人の突発的なライブといった，様々なイベントを検出する社会センサデータの生成に再利用できる．現在，センサデータを共有する幾つかのシステム [7], [8] が研究開発されているが，データとデータ生成のためのプログラム等を関連付けて共有できなかった．社会センサデータの生成・共有を目的としたシステムを実現することで，社会センサデータを様々な応用に再利用しやすくなる．

そこで本研究では，社会センサデータを生成・共有するための基盤システムの設計および実装を行った．提案したシステムでは，社会センサデータを生成するユーザは，社会センサデータを生成する関数群，社会センサデータの型を定義するファイル，社会センサデータの出力方法に関する設定ファイルの三つのファイルを作成する．提案システムは，それぞれのファイルを構文解析し，社会センサデータの生成プログラムを作成，コンパイル，実行して社会セ

1 大阪大学大学院情報科学研究科
2 大阪大学サイバーメディアセンター
3 大阪大学

*1 https://www.twitter.com/
*2 https://www.facebook.com/
*3 https://www.flickr.com/
*4 https://www.instagram.com/
*5 http://semiocast.com/en/publications/2012 07 30Twitter reaches half a billion accounts 140m in the US

ンサデータを生成し，生成した社会センサデータをデータベースに格納する．社会センサデータの生成に関する記述をこれらのファイルに分けることで，他のユーザは，新たな社会センサの開発に有用な部分のみを再利用できる．また提案システムでは，作成された社会センサデータ生成プログラムの実行時刻を指定して，社会センサデータを定期的に更新できる．ユーザは Web インタフェースを介して，システムのデータベースを検索し，社会センサデータやこれらのファイルを共有できる．

以下では，第 2 章で関連研究を紹介し，第 3 章で提案システムの設計について述べ，第 4 章で提案システムの実装および社会センサ登録の具体例について述べる．第 5 章で提案システムに関する議論を行い，最後に第 6 章で本研究のまとめと今後の課題について述べる．

2. 関連研究

本章では，まず SNS へとの投稿等の解析に関する研究について紹介した後，データおよびプログラムソースの共有に関する既存システムを紹介する．

2.1 SNS への投稿等の解析に関する研究

Zafeiris らは，Twitter のツイートを用いた交通状況の推測を行っている [4]．Zafeiris らの手法では，Twitter の提供する API（Application Programming Interface）を用いて収集したツイートを，交通状況が良好であることを示すツイート集合，交通渋滞が発生していることを示すツイート集合，交通状況について言及されていないツイート集合に分類する．これらを教師データとして SVM で分類し，交通状況が良好であることを示すツイート集合と交通渋滞が発生していることを示すツイート集合の位置情報を用いて，交通状況を推測している．Zafeiris らの研究で生成されるデータは，実社会に関する交通状況のデータであり，社会センサデータといえる．

Kim らは，Twitter のツイートを用いて，発生したイベントの時空間的動向を解析して可視化する手法を提案している [5]．Kim らの手法では，Twitter の提供する API を用いて位置情報付きツイートを収集し，事前に入力していた複数のイベント情報と比較解析し，グループ化する．グループ化されたツイートを 3D ビューワを用いて可視化することにより，イベントの時空間的動向を分析している．このようなイベント情報によってグループ化されたイベントの時空間的動向は，実社会のイベントをセンシングして得られたデータであり，社会センサデータといえる．

この研究で生成された社会センサデータは，独自のシステムやアプリケーションとして開発されているが，様々な応用に再利用できるものである．そこで，本研究で提案したシステムは，このような社会センサデータの生成・共有を目的としている．

2.2 データおよびプログラムソースの共有に関する研究

佐崎らは，モバイルセンサデータの共有のためのシステム SeRAVi を提案している [9]．モバイルセンサデータは，スマートフォンやタブレット，ウェアラブルデバイスなどのモバイルデバイスから収集される位置情報付きのセンサデータである．SeRAVi では，モバイルセンサデータを地図インタフェース上に表示し，モバイルセンサデータの分布を可視化することで，モバイルセンサデータの検索支援機構を提供している．SeRAVi では，システムが収集したセンサデータを共有する機能は提供しているが，SNS の投稿等から生成されるデータを共有する機能を提供していない点で提案システムとは異なっている．

また，Github[*6] などプログラムソース共有サイトがある．これらのサイトでは，ユーザ間でプログラムソースを共有でき，その共有したプログラムソースからさらに新しいプログラムソースを作成し，共有できる．プログラムソースの共有の利点は，文献 [10] で言及されている．しかし，Github などのプログラムソース共有サイトは，プログラムソースの共有しか行えず，社会センサデータの生成，共有は行えない点で提案システムとは異なっている．

3. 社会センサデータ生成・共有基盤システムの設計

本章では，社会センサデータ生成・共有基盤システムの設計について述べる．

3.1 機能要件および設計方針

第 1 章で述べた通り，社会センサデータや社会センサデータの生成に関する記述は，新たな社会センサの開発にも有用であり，広く共有することで，様々な応用に再利用できる．以下に，提案システムの機能要件および設計方針を述べる．

3.1.1 社会センサデータの生成および更新

解析対象となる SNS の投稿等は時々刻々と生成されている．解析対象のデータ発生に追従して，社会センサデータを生成するために，提案システムでは登録された社会センサデータ生成プログラムを定期実行して，社会センサデータを更新する機能を提供する．

3.1.2 社会センサデータの共有

生成された社会センサデータを共有するために，システムに蓄積された社会センサデータを検索および取得するための機能を提供する．具体的には，本システムでは生成された社会センサデータに対し，グループごとのアクセス制御を提供する．

3.1.3 社会センサデータ生成に関する記述の共有

第 1 章で述べたように，既存の社会センサデータ生成に

*6 https://github.com/

関する記述は，新たな社会センサの開発に有用である．既存の社会センサデータ生成に関する記述の有用な部分のみ再利用しやすくするために，本システムにおいてユーザは，SNS のデータを解析するコードが記述されたソースファイル，出力される社会センサデータの型定義ファイル，出力される社会センサデータをデータベースに格納する際の設定を記述したファイルの三つのファイルを作成，入力し，これらのファイルをシステム内のデータベースに格納する．

本研究では，これらのファイルをそれぞれ SSFD（Social Sensor Function Description），SSTD（Social Sensor Type Definition），SSOC（Social Sensor Output Configuration）と呼ぶ．これらのファイルへの要求事項については 3.3 節で説明する．このように複数のファイルに分けることで，各ファイルをそのまま，あるいは一部の記述を変更したうえで再利用できる．これらのファイルはシステム内のデータベースに格納され，ユーザは検索および取得できる．システムで実行される社会センサデータ生成プログラムは，ユーザが入力したこれらの三つのファイルを用いて作成される．

3.2　システム構成

提案システムの構成を図 1 に示す．システムは二つのデータベースをもち，SNS データベースは，解析対象となる SNS から得られた未加工のデータを格納するデータベースである．社会センサデータベースは，SNS データベースのデータを入力として生成された社会センサデータ，およびユーザが入力したファイルを格納するデータベースである．他の各構成要素を，以下に説明する．

社会センサデータベース作成受入部

ユーザが入力したファイル群を受け取り，それぞれのファイルを構文解析する．構文解析結果をもとに，社会センサデータ生成プログラムの作成，およびジョブスケジューラの編集を行う．

社会センサデータ生成部

ジョブスケジューラから社会センサデータ生成プログラム実行のトリガを受けると，SNS への投稿等を入力として社会センサデータ生成プログラムを実行し，社会センサデータを生成する．

Web インタフェース

ユーザと社会センサデータ生成・共有基盤システムの間のインタフェースを提供する．

ジョブスケジューラ

社会センサデータ生成プログラムの定期実行を管理する．

3.3　システムへの入力ファイル

本項では，3.1.2 項で述べたユーザの入力ファイルである SSFD，SSTD，SSOC について，要求事項と記述例を示す．これら三つのファイルは，入力された際にシステム内の社会センサデータベースに格納される．実際に社会センサを登録する際に用いる SSFD，SSTD，SSOC については第 4 章で説明する．

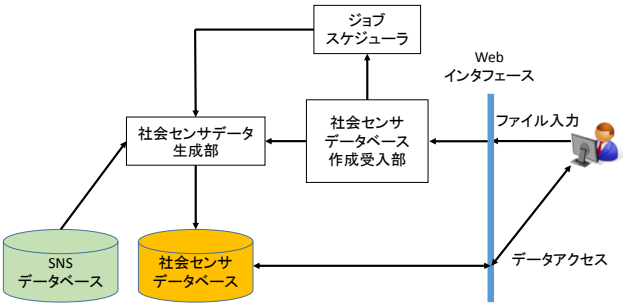

図 1　提案システムの構成

```
所属パッケージの宣言；
利用するクラスの宣言；
public class SNSServiceInterface(){
        ArrayList<Object> dataList = null;          ← SNSへの投稿を読み込む領域
        public class SNSServiceInterface(){         ← SNSへの投稿を取得するメソッド
                                                      （コンストラクタ）
            データベースアクセス；
            SQL発行；
            dataListの取得；
        }
        public ArrayList<Object> getDataList(){     ← SNSへの投稿のgetter
                return this.dataList;
        }
        public Object analyzeData(ArrayList<Object> data){   ← 主要な解析プログラム
                解析処理；
                return analyzedData;
        }
}
```

図 2　SSFD のスケルトンコード

3.3.1　SSFD

SSFD への要求事項として，SNS に投稿されたデータを解析して社会センサデータを生成するためのプログラムを記述できることが挙げられる．図 2 に SSFD のスケルトンコード（プログラムの大枠）を示す．一例として，プログラミング言語に Java を用いて説明するが，使用するプログラミング言語は C++や Python など実装に依存する．

SSFD のクラス名は，任意に記述できる．dataList は，解析するデータを格納する領域である．Object はデータ型を指定する．クラスインスタンス生成時に呼び出されるコンストラクタは，クラス名と同一の名前である．コンストラクタが呼び出されると，SNSServiceInterface が実行され，dataList に解析するデータが読み込まれる．getDataList メソッドは，dataList に読み込まれた解析するデータを取得するための関数であり，社会センサデータ生成プログラム内で解析処理を行う際に，最初に呼び出されるメソッドである．analyzeData メソッドは，SSFD の中心となるメソッドであり，解析するデータを引数としてデータの解析を行う．返り値は解析するデータに対する解析結果であり，解析するデータと同様に，任意のデータ型を指定でき

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE SSTD[
  <!ELEMENT SSTD
(NAME,UPDATE,KEYWORD,ITERATOR,VALUES)>
  <!ELEMENT NAME (#PCDATA)>
  <!ELEMENT UPDATE EMPTY>
  <!ATTLIST UPDATE U_TYPE CDATA #REQUIRED
                   U_TIME CDATA #REQUIRED
                   F_TIME CDATA #REQUIRED>
  <!ELEMENT KEYWORD (#PCDATA)>
  <!ELEMENT ITERATOR (EVERY+)>
  <!ELEMENT EVERY EMPTY>
  <!ATTLIST EVERY I_NAME CDATA #REQUIRED
                  SIZE CDATA #REQUIRED
                  OVERLAP CDATA #REQUIRED
                  I_TYPE CDATA #REQUIRED
                  I_FUNCTION CDATA #REQUIRED>
  <!ELEMENT VALUES (VALUE+)>
  <!ELEMENT VALUE EMPTY>
  <!ATTLIST VALUE V_NAME CDATA #REQUIRED
                  REFERENCE CDATA #REQUIRED
                  V_TYPE CDATA #REQUIRED
                  V_FUNCTION CDATA #REQUIRED>
]>
```

図 3 SSTD の DTD

る．図中の矩形で囲まれたコードブロック内は，ユーザが登録する社会センサが行う解析内容である．

3.3.2 SSTD

SSTD への要求事項としてまず，新たな社会センサを登録するための，社会センサの名前などの識別子を記述できることが挙げられる．また，識別子で識別される社会センサデータを継続的に生成するため，社会センサデータの更新を記述できることおよび，検索に用いるためのキーワードなどのメタデータを記述できることが挙げられる．社会センサデータの生成では，ある期間の複数のツイートの単語出現頻度を算出するといったように，複数の入力データに対して解析を行うことがある．そこで提案システムでは，入力するデータを取得する部分（ITERATOR 要素）と，入力したデータ毎に行う処理（EVERY 要素）に分けて記述する．具体的な値の計算方法は SSFD に記述されているため，SSFD の関数名で計算方法を指定する．

これらの記述方法について，一例として XML を用いて説明する．図 3 に SSTD の XML の DTD（Document Type Definition，マークアップ言語の記述方法を定義したもの）を示す．SSTD には，NAME 要素，UPDATE 要素，KEYWORD 要素，ITERATOR 要素，VALUES 要素をこの順番で記述する．

NAME 要素　生成される社会センサデータ名を記述する．

UPDATE 要素　U_TYPE 属性，U_TIME 属性，F_TIME 属性にユーザの社会センサデータの更新要求について記述する．U_TYPE 属性には社会センサデータの更新の有無を記述し，U_TIME 属性には社会センサデータの更新間隔を記述する．また，F_TIME 属性には社会センサデータの更新終了時刻を記述する．

KEYWORD 要素　登録する社会センサに関する任意のキーワードを記述でき，記述内容は SSTD の検索に用いられる．

ITERATOR 要素　一つ以上の EVERY 要素を子要素としてもち，各 EVERY 要素には解析するデータに関する情報を記述する．EVERY 要素内の I_NAME 属性は，社会センサデータ生成プログラム内で用いられる変数名で，解析するデータを保持する．SIZE 属性にはデータの処理単位数を，OVERLAP 属性には処理対象データの重複数を記述する．I_TYPE 属性，I_FUNCTION 属性には，それぞれ解析するデータの型名，解析するデータを取得するための関数名を記述する．解析する SNS のデータの種類が複数ある場合，解析する SNS のデータの種類の数と同じ数の EVERY 要素を記述する．

VALUES 要素　一つ以上の VALUE 要素を子要素としてもち，各 VALUE 要素には生成される社会センサデータのプロパティ情報を記述する．VALUE 要素内の V_NAME 属性は，社会センサデータ生成プログラム内で用いられる解析結果を格納する変数の名前である．REFERENCE 属性には，先述した I_NAME 属性値のうち，解析するデータとして該当するものを記述する．V_TYPE 属性，V_FUNCTION 属性には，それぞれ解析結果として返されるデータの型名，解析を行うための解析関数名を記述する．生成される社会センサデータが複数のプロパティで定義される場合，プロパティの数と同じ数の VALUE 要素を記述する．

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE SSOC[
  <!ELEMENT SSOC (FILE_ACCESS,DATA_ACCESS,COLUMN,INDEXKEY)>
  <!ELEMENT DATA_ACCESS EMPTY>
  <!ATTLIST DATA_ACCESS CLASS CDATA #REQUIRED>
  <!ELEMENT FILE_ACCESS EMPTY>
  <!ATTLIST FILE_ACCESS CLASS CDATA #REQUIRED>
  <!ELEMENT COLUMN (COLUMN_DATA+)>
  <!ELEMENT COLUMN_DATA EMPTY>
  <!ATTLIST COLUMN_DATA C_NAME CDATA #REQUIRED
                        C_TYPE CDATA #REQUIRED>
  <!ELEMENT INDEXKEY (#PCDATA)>
]>
```

図 4 SSOC の DTD

3.3.3 SSOC

SSOC への要求事項としてまず，社会センサデータや SSFD，SSTD，SSOC へのアクセス制御方法を記述できることが挙げられる．また，生成した社会センサデータをデータベースに格納する際の，カラム型やカラム名および検索時に用いるインデックスキーを指定できることが挙げられる．

SSTD と同様に，一例として XML を用いて説明する．

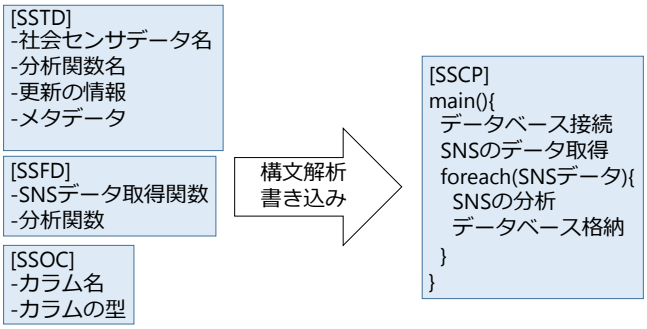

図 5　SSCP 作成のイメージ

SSOC には，DATA_ACCESS 要素，FILE_ACCESS 要素，COLUMN 要素，INDEXKEY 要素をこの順番で記述する．図 4 に SSOC の XML の DTD を示す．

DATA_ACCESS 要素　CLASS 属性をもち，社会センサデータのアクセス制御に関する情報を記述する．

FILE_ACCESS 要素　CLASS 属性をもち，社会センサデータ生成プログラムのアクセス制御に関する情報を記述する．

COLUMN 要素　一つ以上の COLUMN_DATA 要素を子要素としてもち，各 COLUMN_DATA 要素にはテーブルのカラム情報を記述する．

COLUMN_DATA 要素　生成された社会センサデータを社会センサデータベースに保存する際に用いるカラム名，カラムの型名を，C_NAME 属性および C_TYPE 属性に記述する．

INDEXKEY 要素　社会センサデータベースがインデックスを張るカラム名として，C_NAME 属性値の中から該当するものを記述する．

3.4　社会センサの登録

本研究では，SSFD，SSTD，SSOC から作成された社会センサデータ生成プログラムを，ジョブスケジューラに登録することを社会センサの登録と定義する．ユーザは 3.3 節で説明した三つのファイルを入力する．提案システムは入力されたファイルを構文解析し，社会センサデータ生成プログラムを作成し，ジョブスケジューラに登録する．

3.4.1　社会センサデータ生成プログラムの作成

ユーザは三つのファイルをシステムに入力する．この際，システムは SSTD と SSOC の書式が DTD に適合するか確認し，書式が正しければ，システムは入力を受理する．三つのファイルがシステムに入力されると，システムは，SSFD，SSTD，SSOC の三つのファイルを構文解析を行い，構文解析を行った結果から，社会センサデータを生成する関数および社会センサデータベースへの格納に関する SQL 等の記述を含んだソースファイルを作成する．本研究では，このファイルを SSCP（Social Sensor Creation Program）と呼ぶ．図 5 に SSCP の作成イメージを示す．

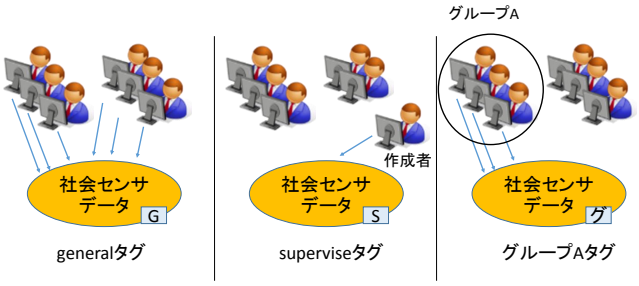

図 6　社会センサデータに対するアクセス制限の例

システムは，SSCP をコンパイルし，コンパイル時のログを Web インタフェース上に出力して SSCP の実行可能プログラムを作成する．コンパイルに失敗した場合，エラーログを出力する．SSCP の作成後，提案システムは，SSTD に従い，システムが作成したジョブスケジューラ用のファイルを編集し，SSCP をジョブスケジューラに登録する．

3.4.2　社会センサデータの生成および更新

作成された SSCP の実行可能プログラムを実行すると，SNS への投稿等の解析対象データが取得され，社会センサデータが生成された後，社会センサデータベースに格納される．また，SSTD に更新要求の情報が記述されている場合，SSCP の実行可能プログラムは社会センサデータ生成部で保持される．その後，ジョブスケジューラにより一定間隔で，SSCP の実行可能プログラムが実行され，社会センサデータが更新される．

3.5　グループ登録

ユーザはシステム内にグループを設定でき，グループ内で社会センサデータや，社会センサデータ生成に関する記述を共有できる．例えば，groupA というグループがシステムに登録されているとする．ユーザは社会センサデータに対して，general タグ，supervise タグ，groupA タグのいずれかを付与できる．図 6 に示すように，general タグが付与された社会センサデータはすべてのユーザが，社会センサデータや社会センサデータ生成に関する記述にアクセスできる．supervise タグが付与された社会センサデータは作成ユーザのみが，groupA タグが付与された社会センサデータは groupA に所属しているユーザのみがアクセスできる．

4.　社会センサデータ生成・共有基盤システムの実装

第 3 章で述べた設計に基づき，提案システムを実装した．

4.1　実装環境

CentOS6.7 搭載のワークステーションを用いて提案システムを構築した．構築したサーバのプログラムバージョン等の仕様を表 1 に示す．社会センサデータの更新のための

表 1　サーバの仕様

Web サーバソフトウェア	Apache/2.2.15 (Unix)
サーブレットエンジン	Apache Tomcat/8.0.30
データベースソフトウェア	mysql 5.7.10

表 2　社会センサ情報の具体例（名詞抽出）

社会センサデータ名	TweetNouns
更新間隔	1 時間ごと
終了予定時刻	2016 年 2 月 19 日 0 時
検索用キーワード	Twitter，トピック抽出，キーワード
分析内容	ツイートに含まれる名詞の抽出
データの公開範囲	全ユーザ
プログラムの公開範囲	グループメンバのみ
インデックス構築先カラム	ツイート時刻

ジョブスケジューラには，UNIX 系 OS に標準で搭載されている crontab を用いた．SSCP の作成，crontab ファイルの編集，データベースへの問合せなど，サーバ側の処理は Java1.8.0_66 で実装した．SSCP の作成および crontab ファイルの編集時の，ユーザが入力する XML ファイルの構文解析には，Java で XML を扱うための API である JAXP を用いた．また，サーバからのデータベースへのアクセスには JDBC5.1.37 を用いた．

4.2　具体例：ツイートに含まれる名詞の抽出

本節では，具体例を用いて，社会センサの登録，社会センサデータの生成について説明する．不特定多数の Twitter ユーザのつぶやきから抽出された，ユーザ間で話題となっているトピックは，社会情勢の変遷の把握，ユーザ推薦，マーケティングなど，様々な応用が考えられる．本項では二つ目の社会センサ例として，Twitter のツイート中に含まれる名詞のみを抽出する社会センサについて説明する．登録する社会センサの情報を表 2 に示す．ユーザは，これらの情報に従って SSFD，SSTD，SSOC の三つのファイルを作成し，社会センサ登録時にシステムに入力する．

4.2.1　名詞抽出の例で用いる SSFD

SSFD は JAR ファイルで作成されている．図 7 に，ツイートに含まれる名詞を抽出する例における SSFD として入力する JAR ファイルの構成を示す．TwitterApplication.jar ファイルでは，TwitterInterface メソッドを用いて tweetList にツイートを格納する．次に，getTweet メソッドを用いてツイートを取得し，取得したツイートを引数として extractNoun メソッドを実行して，ツイートを形態素解析する．ツイートの形態素解析には，Java の形態素解析器である Kuromoji[*7] を用いた．ユーザは，システムに SSFD を入力する際，自身の作成した JAR ファイルに加えて，用いた API に関する JAR ファイル等もシステムに

*7　https://github.com/atilika/kuromoji/downloads

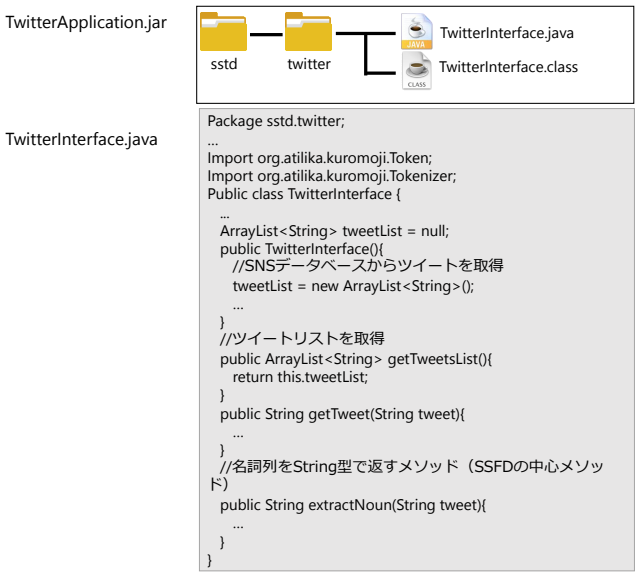

図 7　SSFD の具体例（名詞抽出）

```
<?xml version="1.0" encoding="UTF-8"?>
<SSTD>
<NAME>TweetNouns</NAME>
<UPDATE U_TYPE="EVERY "U_TIME="10 * * * *" F_TIME="201602190000"/>
<KEYWORD>Twitter,トピック抽出,キーワード</KEYWORD>
<ITERATOR>
      <EVERY I_NAME="tweet"
                SIZE="1"
                OVERLAP="0"
                I_TYPE=" ArrayList&lt;String&gt; "
                I_FUNCTION=" sstd.twitter.TwitterInterface.getTweetsList() "/>
</ITERATOR>
<VALUES>
  <VALUE  V_NAME="microblog"
            REFERENCE="tweet"
            V_TYPE="String"
            V_FUNCTION=" sstd.twitter.TwitterInterface.getTweet "/>
  <VALUE  V_NAME="nouns"
            REFERENCE="tweet"
            V_TYPE="String"
            V_FUNCTION=" sstd.twitter.TwitterInterface.extractNoun "/>
</VALUES>
</SSTD>
```

図 8　SSTD の具体例（名詞抽出）

アップロードする．

4.2.2　名詞抽出の例で用いる SSTD

図 8 にツイートに含まれる名詞を抽出する例の SSTD を示す．社会センサデータ名を，NAME 要素に記述する．毎時 10 分に社会センサデータを更新するように，UPDATE 要素の U_TYPE 属性に “Every” を，U_TIME 属性に crontab の記法である “10 * * * *” を記述する．更新終了予定時刻を 2016 年 2 月 19 日 0 時とし，F_TIME 属性に “201602190000” を記述する．処理対象となる SNS のデータは Twitter のツイートのみであり，ITERATOR 要素の子要素は 1 つのみとなる．SSCP 内で用いられる変数名である I_NAME 属性は，“tweet” とする．名詞の抽出は単一のツイートごとに行うため，SIZE 属性は “1”，OVERLAP 属性は “0” を記述する．I_TYPE 属性，I_FUNCTION 属性は，それぞれ SSCP 内で処理対象のデータ型名および処理対象のデータを返す関数名を指定している．生成する社会センサデータ

```
<?xml version="1.0" encoding="UTF-8"?>
<SSOC>
<FILE_ACCESS CLASS="Osaka-u"/>
<DATA_ACCESS CLASS="general"/>
<COLUMN>
<COLUMN_DATA C_NAME="microblog" C_TYPE="text"/>
<COLUMN_DATA C_NAME="nouns" C_TYPE="verchar(100)"/>
</COLUMN>
<INDEXKEY>nouns</INDEXKEY>
</SSOC>
```

図 9　SSOC の具体例（名詞抽出）

microblog	nouns
きつねかわいい！！！(2000138302回目)	きつね,！！！(,2000138302,回,目,),
@kamereo_royz もちろん	@,kamereo,_,royz,
きつねかわいい！！！(2000138303回目)	きつね,！！！(,2000138303,回,目,),
@syota449 どこがですか？	@,syota,449,どこ,
ＪＲ浜松町駅にて、小便小僧とBiBiとの奇跡のコラボ #小泉花陽生誕祭 #lovelive #ラブライブ #BiBi #小便小僧 https://t.co/RYaqdAuyeR	ＪＲ,浜松町,駅,小便,小僧,BiBi,奇跡,コラボ,#,小泉,花陽,生誕,祭,#,lovelive,#,ラブ,ライブ,#,BiBi,#,小便,小僧,https,://,t,.,co,/,RYaqdAuyeR,
高校時代も、髪型がエミネムだね とかドラゴンアッシュかっこいいよね。ってほのめかされた	高校,時代,髪型,エミネム,ドラゴンアッシュ,

図 10　社会センサデータの具体例（名詞抽出）

は，ツイート本文，ツイートから抽出された名詞であり，それぞれ VALUES 要素の子要素として記述する．SSCP 内で用いられる変数名である V_NAME 属性は“microblog”，“nouns” とした．社会センサデータの属性は，いずれも単一のツイートから算出されるため，REFERENCE 属性は全て I_NAME 属性の値を記述する．V_TYPE 属性は，それぞれのデータ型名を記述し，V_FUNCTION 属性は，それぞれの社会センサデータの属性値を算出する関数名を指定している．

4.2.3　名詞抽出の例で用いる SSOC

図 9 にツイートに含まれる名詞を抽出する例の SSOC を示す．登録する社会センサは，データを全ユーザに公開するため，DATA_ACCESS 要素の CLASS 属性を“general”で指定する．プログラムを“Osaka-u” グループに公開するため，FILE_ACCESS 要素の CLASS 属性を，“Osaka-u”で指定する．社会センサデータの属性値を格納するカラム情報は，COLUMN 要素の子要素にそれぞれ記述する．COLUMN_DATA 要素の C_NAME 属性は，SSCP 内で用いる変数名を記述し，C_TYPE 属性はデータベースに格納する際の型名を記述する．INDEXKEY 要素には名詞を格納する変数名“nouns”を記述する．

ユーザはこれら三つのファイルを入力する．提案システムは入力ファイルから SSCP を作成し，社会センサの登録を行う．提案システムが定期的に SSCP を実行することで，図 10 に示す通り，社会センサデータを生成する．この社会センサデータを共有することで，スポット名の書かれた名詞から，そのツイートに対して位置情報を付与するなどの応用に役立てられる．

表 3　機能比較

	データの解析	社会センサデータ生成に関する記述の共有	社会センサデータの共有	社会センサデータの可視化
提案システム	○	○	○	×
mTrend	△	×	×	△
Github	×	△	×	×
SeRAVi	×	×	△	△

5.　議論

本章では，提案システムと既存システムの機能比較などの議論を行う．

5.1　機能比較

本節では，提案システムと既存システムの機能比較を行う．比較対象は，社会センサデータの生成・共有に関連する機能を有する，mTrend[5]，Github，SeRAVi[9] とした．これらの概要については，2 章で説明している．機能比較の結果を表 3 に示す．

表 3 において，データの解析とは SNS への投稿等のデータを解析できることを示す．提案システムでは，SSFD に解析処理を記述することで，SNS のデータ等を解析できる．mTrend は，ツイートを解析して時空間的動向を可視化しており，解析可能といえる．ただし，ユーザが作成したプログラム等に従った任意の解析を行えない．一方で，Github や SeRAVi は主にプログラムやデータの共有を目的としており，SNS 等のデータを解析する機能を有していない．

社会センサデータ生成に関する記述の共有とは，Java 等のプログラミング言語で記述された社会センサデータ生成に関する記述の共有機能を指す．提案システムでは，SSTD，SSFD，SSOC の共有機能を提供しており，社会センサデータ生成に関する記述を共有できる．また，アクセス制御により，これらのファイルの共有を拒否することもできる．mTrend は，ツイートの解析を行っているものの，解析のプログラムを公開しておらず，社会センサデータ生成に関する記述を共有する機能を有していない．Github ではプログラムの共有は可能だが，提案システムのように三つのファイルに分けるなど，社会センサデータ生成に関する記述の共有には至っていない．また，SeRAVi はプログラムの共有機能を提供していない．

社会センサデータの共有とは，生成された社会センサデータを共有できる機能を指す．提案システムでは，Web インタフェースを介して社会センサデータを共有する機能を提供している．mTrend は，時空間的動向そのもののデータを共有する機能は提供していない．Github は，プログラムの共有に主眼を置いており，社会センサデータを共有する機能を提供していない．SeRAVi は，システムが収集したセンサデータを共有する機能を提供しているもの

の，生成された社会センサデータを共有する機能を提供していない．

社会センサデータの可視化とは，生成された社会センサデータの時間的，あるいは空間的な分布を可視化する機能を指す．提案システムは，生成された社会センサデータの分布を可視化する機能は提供していない．mTrend では，ツイートを解析して時空間的動向を可視化しており，ツイートを解析した結果は社会センサデータといえるため，社会センサデータの可視化を行える．ただし，ユーザが登録した社会センサデータを可視化できるわけではない．Github は，生成した社会センサデータの分布を可視化する機能は提供していない．SeRAVi は，モバイルセンサデータの分布を可視化する機能は提供しているものの，生成された社会センサデータを可視化する機能を有していない．

以上より，データの解析，社会センサデータ生成に関する記述の共有，社会センサデータの共有，社会センサデータの可視化を行うシステムはいくつか存在するが，社会センサデータの生成，共有に適したシステムは，提案システムのみといえる．

5.2 ストリームデータへの対応

提案システムでは，SSFD に SNS への投稿等を取得するプログラムを記述することで，それらのデータから社会センサデータを生成できる．SNS のデータ等は時々刻々と発生しているため，これらのデータを連続的に取得して社会センサデータをストリームデータ（連続的に発生するデータ）として生成することが考えられる．現状では，社会センサデータ生成プログラムを実行する度に社会センサデータを生成しているが，上記のようなストリームデータに対応することで，現在時刻に近いデータを用いた社会センサデータなどを生成できる．

6. まとめ

本研究では，社会センサデータを生成・共有するための基盤システムの設計および実装について述べた．提案したシステムでは，ユーザは社会センサデータを生成するために，SSFD，SSTD，SSOC の三つのファイルを作成する．提案システムは，これらのファイルを構文解析し，社会センサデータの生成プログラムを作成，コンパイルする．社会センサデータ生成に関する記述をこれらのファイルに分けることで，他のユーザはそれぞれのファイルの有用な部分を再利用でき，新たな社会センサを開発しやすくなる．

さらに，本研究では，SNS データの解析，社会センサデータ生成に関する記述の共有，社会センサデータの共有，社会センサデータの可視化という観点から提案システムと既存システムの機能を比較した．その結果，提案システムでは，SSFD に解析処理を記述することでデータの解析が可能であり，また，SSFD，SSTD，SSOC に分けて共有することで，社会センサデータおよび社会センサデータ生成に関する記述の共有に適していることを確認した．

今後の課題として，より現在時刻に近い社会センサデータを解析するために，社会センサデータのみをユーザの端末にストリーム配信することが考えられる．

謝辞　本研究の一部は，文部科学省科学研究費補助金・基盤研究（A）（26240013），挑戦的萌芽研究（26540045），JST 国際科学技術共同研究推進事業（戦略的国際共同研究プログラム）の研究助成によるものである．ここに記して謝意を表す．

参考文献

[1] Sakaki, T., Okazaki, M. and Matsuo, Y.: Earthquake shakes Twitter users: real-time event detection by social sensors, Proc. World Wide Web Conf., pp. 851–860 (2010).

[2] 杉谷卓哉，白川真澄，　原隆浩，西尾章治郎：教師あり機械学習を用いたツイート投稿時のユーザ位置推定手法，情報処理学会研究報告，Vol. 2013, No. 26, pp. 1–8 (2013).

[3] Aramaki, E., Maskawa, S. and Morita, M.: Twitter catches the flu: detecting influenza epidemics using Twitter, Proc. ACL Int'l Conf. on Empirical Methods in Natural Language Processing, pp. 1568–1576 (2011).

[4] Kokkinogenis, Z., Filguieras, J., Carvalho, S., Sarmento, L. and Rossetti, R. J.: Mobility Network Evaluation in the User Perspective: Real-Time Sensing of Traffic Information in Twitter Messages, Proc. Int'l Conf. on Web search and web data mining (WSDM 2008), pp. 183–194 (2015).

[5] Kim, K.-S., Lee, R. and Zettsu, K.: mTrend: Discovery of topic movements on geo-microblogging messages, Proc. of ACM Int'l Conf. on Advances in Geographic Information Systems, pp. 529–532 (2011).

[6] 橋本康弘，岡　瑞起：都市におけるジオタグ付きツイートの統計，人工知能学会誌，Vol. 27, No. 4, pp. 424–431 (2012).

[7] Turon, M. and Suh, J.: Mote-view: A sensor network monitoring and management tool, The Second IEEE Workshop on Embedded Networked Sensors, pp. 11–18 (2005).

[8] Michel, S., Salehi, A., Luo, L., Dawes, N., Aberer, K., Barrenetxea, G., Bavay, M., Kansal, A., Kumar, K. A., Nath, S. et al.: Environmental Monitoring 2.0, Data Engineering, 2009. ICDE'09. IEEE 25th International Conference on, pp. 1507–1510 (2009).

[9] Sazaki, H., Kanzaki, A., Hara, T. and Nishio, S.: SeRAVi: a Spatio-Temporal Data Distribution Visualization System for Mobile Sensor Data, Proc. Int'l Workshop on Future Technologies for Smart Information Systems (FTSIS 2014), pp. 88–93 (2014).

[10] Dabbish, L., Stuart, C., Tsay, J. and Herbsleb, J.: Social coding in GitHub: transparency and collaboration in an open software repository, Proc. ACM Int'l Conf. on Computer Supported Cooperative Work, pp. 1277–1286 (2012).