特　集「Linked Dataとセマンティック技術」

ストリーム推論

Stream Reasoning

市瀬　龍太郎　国立情報学研究所情報学プリンシプル研究系
Ryutaro Ichise　National Institute of Informatics.
ichise@nii.ac.jp, http://ri-www.nii.ac.jp/

Keywords: stream reasoning, RDF stream, complex event processing, advanced driver assistance system.

1. はじめに

近年，さまざまな場所にセンサが取り付けられ，収集されたデータに基づき意思決定が行われるようになってきている．例えば，スマートシティでは，都市に張り巡らされたセンサの情報をもとに，省エネルギーで，効率的な都市にするための意思決定が行われているし，先進運転支援システムでは，自動車に取り付けられたセンサの情報をもとに，衝突回避などの安全性を増すための意思決定が行われている．そのようなシステムにおいては，センサから入力されてくるストリームデータを機械が処理し，そのデータから得られる情報に基づき意思決定が行われている．ストリームデータから複雑で知的な意思決定を行うためには，推論が大きな役割を果たすため，ストリーム推論（Stream Reasoning）と呼ばれる研究領域に注目が集まっている．ストリーム推論とは，データストリーム，セマンティックWeb，推論システムを統合するために必要な手法やツールなどを提供する学際的な研究分野である[Valle 09]．本稿では，ストリーム推論についての解説を行う．

本稿の構成は以下のようになっている．まず，2章では，ストリーム推論の概念，従来技術との違い，実現方法について説明を行う．続く3章では，セマンティックWebで基本となるデータ形式RDFを用いてストリームデータを処理する方法について述べる．4章では，ストリーム推論を使った応用例として，先進運転支援システムを例に取り，どのようにしてシステムを実現できるかの説明を行う．最後の5章で，本稿をまとめる．

2. ストリーム推論

2·1　ストリームデータからの意思決定

ストリームデータを使って，知的な意思決定を行う際には，いくつかの課題があり，Valleらは，その課題を広く論じている[Valle 09]．本稿では，次の四つに整理して述べる．

（1）データ量
（2）ストリーム処理
（3）データの多様性
（4）推論

まず，第1の課題は，大量のデータをどのように処理するかという課題である．ストリームデータは，センサごとに随時データが生成されるため，データが大量になるという性質がある．そのため，大量のデータを処理するための機構が必要となる．第2の課題は，ストリームデータの継続処理である．第1の課題とも関連するが，ストリームデータは，随時生成されるため，データが生成されると同時に継続的に処理されないと処理が滞ってしまうという問題が起こる．第3の課題は，ストリームデータの多様性である．通常，ストリームデータは，1種類のセンサから発生するものを処理するだけではなく，さまざまな種類のセンサから集められるさまざまな種類のストリームデータを処理しなければならない．第4の課題は，得られたストリームデータを用いて，どのように推論を行うかである．通常，ストリームデータから得られる情報は，センサが置かれた環境下の状況を表している．意思決定を行うには，多数のセンサから得られたそれぞれの状況を組み合わせて判断を行う必要がある．そのため，高度な意思決定を行うには，推論機構が必要となる．

2·2　複合イベント処理

2·1節であげた課題を解決するために，データベース技術の観点からは，ストリームデータを処理する複合イベント処理（CEP：Complex Event Processing）という考え方が提起されている．従来のデータベース技術では，データをデータベースに格納し，改めて分析するというアプローチが取られていた．しかし，ストリームデータの場合には，データが随時生成されるため，そのようなアプローチを取るのが困難である．そこで，イベントに着目し処理するというのが，複合イベント処理の基本的な考え方である．複合イベント処理では，あらかじめデータに起こるであろうイベントの条件を複合イベント

処理エンジンに登録し，ストリームデータの監視を行う．そして，ストリームデータにそのイベントが起こった際に，あらかじめ定義した処理を行う．例えば，株式取引において，ある銘柄の株価が急に下落した場合に，その銘柄を売るなどの処理が，この枠組みによって容易に実現できる．2009 年前後から，IBM や Oracle，マイクロソフトといった大手の会社が，複合イベント処理のための基盤ソフトウェアに次々と参入している．最近では，オープンソースの Apache Storm [Storm] がつくられ，Yahoo! や Baidu でも利用されるなど，急速に環境が整ってきている．複合イベント処理に関する近年の状況は，Vincent がまとめている [Vincent 14].

2·3 ストリーム推論

複合イベント処理は，主に課題 1，2 を解決することに注力している．一方，セマンティック Web 技術は，課題 3，4 を解決することには適しているが，課題 1，2 に対応することが困難であった．しかし，近年の技術開発により，セマンティック Web 技術と親和性をもった形で，課題 1，2 が解決されつつある．そこで，推論手法をもち，多様なデータを統一的に扱えるセマンティック Web 技術を用いることで，ストリームデータの推論を実現するのが，ストリーム推論の基本的な考え方となる．

ストリーム推論を行うためには，随時生成される大量のストリームデータから得られる情報をどのようにして，推論の枠組みに入れるかが最大の課題となる．そのための主な枠組みとして，データ駆動型ストリーム推論と検索駆動型ストリーム推論の二つがある [Balduini 14].

§1 データ駆動型ストリーム推論

データ駆動型ストリーム推論を図示すると，図 1 のように表すことができる．データ駆動型ストリーム推論では，ストリームデータは，ストリーム処理をする機構により，イベントなどを示す RDF データに変換される．その RDF データは，推論器に投入される．推論器は，新しい RDF データが生成されるたびに，オントロジーを用いることで推論を行い，推論結果のデータを生成する．その推論結果のデータに対して，SPARQL で検索を行い，意思決定に利用する．この方式では，ストリーム処理に独自の機構を入れてストリームデータを RDF データに変換する．そのため，それ以降の処理は，セマンティック Web で研究されている推論器などを利用できるという利点がある．しかし，ストリームから生成される RDF データが随時推論に利用されることになるので，全体の処理が重くなるという欠点がある．この方式の実装としては，C-SPARQL [Barbieri 09] などがある．

§2 検索駆動型ストリーム推論

検索駆動型ストリーム推論を図示すると，図 2 のように表すことができる．検索駆動型ストリーム推論では，検索が行われると，オントロジーを用いて推論が行われ，ストリーム処理に適したような検索式に変換される．その検索式をストリーム処理に適用することで，意思決定に利用する．この方式では，推論処理に独自の機構を入れて，検索式から推論することで，ストリーム処理に適した検索式に変換を行う．ストリーム処理部分は，RDF に対応したストリーム処理機構を用いてもよいが，データベース分野などで研究されている機構も利用可能である．この方式を用いると，検索に関連するストリームデータだけを処理すればよいので効率的ではあるが，ストリームデータを操作するための時系列推論をどう行うかなどの問題点がある．このような研究としては，オントロジーに基づくデータアクセス [Calbimonte 10] がある．この研究では，S_2O と呼ばれるマッピングを使うことで，検索の書換えを行う．この方式の実装としては，Morph-Stream [Morph-Stre] がある．

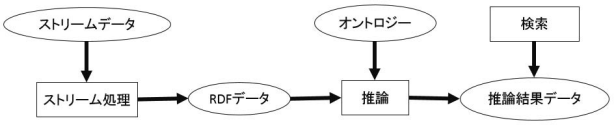

図 1 データ駆動型ストリーム推論

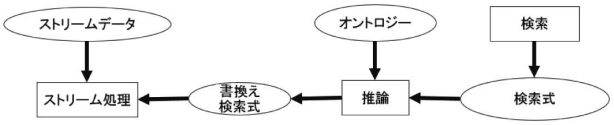

図 2 検索駆動型ストリーム推論

3. RDF を用いたストリーム処理

本章では，2 章で述べた方式のストリーム処理部分について説明する．検索駆動型ストリーム推論では，使う変換方法により複合イベント処理の機構をそのまま利用することもできるが，本稿では両方式で使える RDF を用いたストリーム処理に着目して説明を行う．

3·1 ストリームデータの RDF 表現

セマンティック Web においては，さまざまなデータの表現に Web 上でデータ交換をするための標準モデルである RDF（Resource Description Framework）[RDF Working Group 14] を用いる．RDF は，主語（subject），述語（predicate），目的語（object）の三つの要素で表され，これをトリプルと呼ぶ．本稿では，トリプルを以下のように記述する．

```
<subject, predicate, object>
```

RDF では，この三つの要素を使うことにより，データの記述を行う．しかし，トリプルを単体で使った場合には，ストリーム処理の際に必須となるイベントが発生した時刻を表現することが難しい．例えば，太郎君が移動している最中に，神保町駅に滞在した場合に，下記のトリプルが得られる．

< 太郎，滞在，神保町駅 > （1）

しかし，その後，移動して大手町駅に到着した場合には，以下の別のトリプルが得られる．

< 太郎，滞在，大手町駅 > （2）

この二つのトリプルに対して，時刻情報が欠けていると，現在，太郎君がどこにいるかを判別できず，推論に利用することができない．

このような時系列のデータをRDFで表現するためには，「時刻に基づく表現（point-based）」と「期間に基づく表現（interval-based）」の二つの手法がある[Gutierrez 07]．時刻に基づく表現は，何かのイベントが起こった際に，時刻とともにRDFを記録して利用する方法である．もう一方の期間に基づく表現は，何かのイベントが起こった際に，そのイベントが起こっている期間（開始時刻，終了時刻）とともにRDFを記録して利用する方法である．後者のほうが，イベントの重なりを検知したりする場合など，複雑な状況を処理する際に，より簡単に表現することができる．しかし，期間に基づいてストリームデータを表現する場合には，イベント終了時までデータを蓄積して，その後，RDFに変換する必要が出てくる．そのため，ストリーミング処理の際には，イベント（データ）が発生したときを単に記録する前者のデータモデルが多く用いられている．

時刻とともにRDFを記録する方法として，Time-Annotated-RDF（TA-RDF）[Rodríguez 09]のように，従来のRDFの枠組みの中で表現する手法も考案されている．TA-RDFでは，ブランクノードを用いて時刻と値の表現を行う．しかし，そのような手法を用いると，推論に複雑な処理が必要になるため，RDFトリプルにタイムスタンプを付ける，時間トリプル（temporal triple）というモデルが広く用いられている．

時間トリプルは下記のように定義される．

定義1：時間トリプル

RDFトリプル $< s, p, o >$ に対して，時間ラベル t（t は自然数）を付与したとき，$<< s, p, o >, t >$ を時間トリプルと呼ぶ．

時間トリプルは，RDFトリプルに時刻を付与しただけのものである．この記法は，ストリームデータが生成されたときに，ストリームの値をRDFとして与え，生成時刻を時間ラベルとして記述するだけであるので，データの構成が容易である．また，この表現自体は，時刻に基づく表現になるが，時間トリプルの集合

$$\{<< s, p, o >, t > \mid t_1 \leq t \leq t_2\}$$

を考慮することで，$t_1 \sim t_2$ までの期間を用いた期間に基づく表現に変換することが可能となる．ここで定義された時間トリプルは，一般的にRDFストリーム（RDF stream）と呼ばれる．

3·2 RDFストリームの処理

関係データベースを検索するための検索言語としてSQLがある．それに対応して，RDFデータベースを検索するための検索言語としてSPARQLが設計された．SQLに対して，ストリームデータを扱えるようにした検索言語として，CQL [Arasu 06]がある．SPARQLに対しても同じように拡張を行い，RDFストリームを扱えるようにする検索言語が考案されている．例えば，C-SPARQL [Barbieri 09]，Streaming SPARQL [Bolles 08]，SPARQLstream [Calbimonte 10]，EP-SPARQL [Anicic 11]，CQELS [Le-Phuoc 11]などがあげられる．これらは，同じようなモデルを用いているが，それぞれの検索式は，微妙に異なって定式化されている．さまざまなシステムがあるが，本稿では，RDFストリームを取り扱う機構の代表として，C-SPARQL [Barbieri 09][*1]について解説する．

C-SPARQLは，SPARQL 1.1に対して，ストリームデータを扱えるように拡張を施したものである．SPARQL 1.1では，ある時点におけるRDFデータを検索することができる．それに対して，C-SPARQLでは，時系列のデータを検索できるように拡張が施されている．時系列で表現されるRDFストリームを扱うために，RDFデータセットを指定する際に，SPARQLで利用可能なFROM句とFROM NAMED句に加えて，FROM STREAM句が追加されている．さらに，RDFトリプルで時刻を取り扱うための関数がC-SPARQLでは追加されている．WHERE句の中で，timestamp関数を利用することができる．この関数でRDFストリームに付与されている時間を呼び出すことができ，大小比較などを行うことで，イベントの順序の確認などが可能となる．

C-SPARQLでは，検索式をREGISTERステートメントで登録して利用することができる．すると，登録された検索式に従って，随時，その時点における変数値などを返す．そのため，FROM STREAM句を用いて，検索式でRDFストリームを検索対象に指定すると，データストリームの監視を行い，条件に応じた動作をさせることができる．条件としては，一定の時間幅を設定し，その時間幅を指定した時間ステップごとにスライドさせな

表1 C-SPARQLの検索式の例

```
REGISTER QUERY OverSpeedCheck AS
SELECT ?car
FROM STREAM
<http://www.toyota-ti.ac.jp/Lab/Denshi/COIN/
  stream>
[RANGE 500ms STEP 50ms]
WHERE {
  ?car car:velocity ?speed . }
GROUP BY ?speed HAVING(AVG(?speed) >=
  maxSpeed)
```

*1 http://streamreasoning.org/download から入手可

がら監視することなどができる．例えば，自動車の速度センサの情報を表現する RDF ストリームを用いて，速度超過になっていないかを調べるためには，表 1 のような検索式を登録する．この検索式では，直近 500 ミリ秒の時間幅における速度の平均値を利用して速度超過を監視しており，50 ミリ秒ごとに更新している．

4. 先進運転支援システムへの応用

先進運転支援システム（Advanced Driver Assistance System）は，自動車の運転を支援するためのシステムであり,さまざまな自動車への導入が始まっている．例えば，先進運転支援システムの一つである前方衝突警報システムは，すでに市販車への設置が始まっている．このシステムでは，前方を監視するセンサの情報をもとに，衝突の可能性を検知し，自動的に自動車を停止させる動作を行う．前方衝突警報システムのような単純な動作を決定するだけの場合には，単にセンサ値が一定の値になったことを利用して，自動車の動作を決定することができる．しかし，複雑な動作を行う場合には，ストリーミングデータに基づいて推論を行う必要がある．

本章では，セマンティック Web 技術を用いてストリーミングデータを利用する例として，著者らが研究している先進運転支援システム [Zhao 14, Zhao 15] について説明する．先進運転支援システムを構成するために，知識を記述するオントロジーの設計を行い，関連する情報をインスタンスとして記述した．さらに，推論を行うための規則を記述した．それらの知識は，ストリームデータを処理するための C-SPARQL 検索，トリプルを処理するための SPARQL 検索，推論器を通して利用され，意思決定が行われる．以下，順に説明を行う．

4·1 オントロジーによる知識記述

§1 オントロジー

先進運転支援システムを実現するために，以下の三つのオントロジーの設計を行った．

- ●地図オントロジー
- ●制御オントロジー
- ●自動車オントロジー

地図オントロジーは，道路，交差点，車線など自動車が走れる環境の物理的な要素や，制限速度といった環境に付随する情報を記述するためのオントロジーである．地図オントロジーの主要なクラスを図 3 に示す．地図オントロジーは，全部で 78 個のクラスをもち，18 個のオブジェクトプロパティと 18 個のデータ型プロパティで構成されている．オブジェクトプロパティは，クラスのインスタンス間の関係を示すのに使われる．例えば，

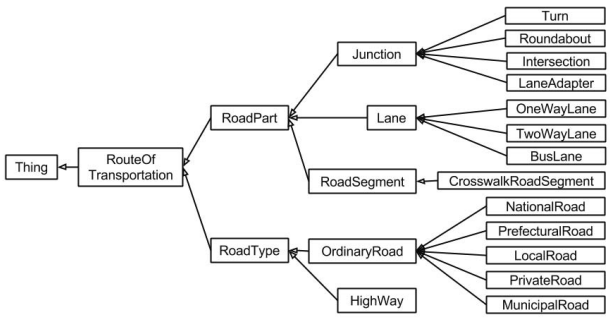

図 3 地図オントロジー

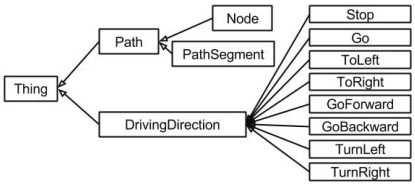

図 4 制御オントロジー

map:turnLeftTo[*2] というプロパティは，異なる道路部品の間の接続関係を記述するのに使われる．データ型プロパティは，インスタンスとデータの関係を示すのに使われる．例えば，map:speedMax というプロパティは，道路の最高速度を記述するのに使われる．

制御オントロジーは，自動車の制御に関する情報を記述するためのオントロジーである．制御オントロジーの主要なクラスを図 4 に示す．制御オントロジーは，全部で 34 個のクラスをもち, 15 個のオブジェクトプロパティと 2 個のデータ型プロパティで構成されている．制御オントロジーにおける，オブジェクトプロパティとしては，運転経路を表す control:nextPathSegment[*3] や，他の自動車との衝突警告を表す control:collisionWarningWith などがある．データ型プロパティとしては，経路セグメントの ID を示す control:pathSegmentID などがある．

自動車オントロジーは，自動車や，その自動車に搭載されているセンサなどのデバイスに関する情報を記述するためのオントロジーである．自動車オントロジーの主要なクラスを図 5 に示す．自動車オントロジーは，全部で 33 個のクラスをもち，3 個のオブジェクトプロパティと 15 個のデータ型プロパティで構成されている．自動車オントロジーにおける，オブジェクトプロパティとしては，自動車がいる位置を表す car:isRunningOn[*4] などがある．データ型プロパティとしては，自動車の速度を表す car:velocity などがある．

§2 インスタンス

先進運転支援システムを構築するために，前項で示した三つのオントロジーを利用して，以下の三つのインスタンスを構築した．

---

*2 PREFIX map：<http://www.toyotati.ac.jp/Lab/Denshi/COIN/Map#>

*3 PREFIX control：<http://www.toyotati.ac.jp/Lab/Denshi/COIN/Control#>

*4 PREFIX car：<http://www.toyotati.ac.jp/Lab/Denshi/COIN/Car#>

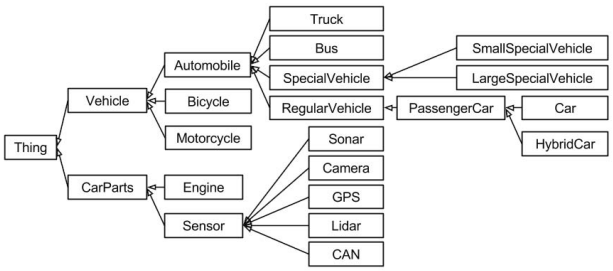

図5 自動車オントロジー

●天白区地図インスタンス：名古屋市天白区の地図の一部を記述したもの．
●経路インスタンス：自動車が移動する経路を記述したもの．
●自動車インスタンス：自動車とそれに搭載されているデバイスを記述したもの．

4·2 推 論 規 則

意思決定に用いる推論規則を記述するために，セマンティック Web 規則記述言語（SWRL：Semantic Web Rule Language）[Horrocks 04] を用いた．例えば，信号のない交差点における，優先通行権の規則を記述するために，以下の二つの規則を用いた．

●collisionWarningWith（?carX, ?carY）
⇒CollisionWarning（?carX）∧CollisionWarning（?carY）
●CollisionWarning（?carX）∧CollisionWarning（?carY）
∧GoForward（?carY）∧TurnRight（?carX）
⇒Stop（?carX）∧giveWay（?carX, ?carY）

この規則は，衝突警告が出た場合には，右折する自動車が止まり，直進する自動車に道を譲るということを示している．

4·3 C-SPARQL 検索

センサデータから入力される情報を監視するために，C-SPARQL を用いて RDF ストリームの検索を行っている．例として，速度超過になっていないかを調べるために利用した C-SPARQL 検索式を表 1 に示す．この検索式は，500 ミリ秒間の平均速度が maxSpeed を超えていないかを調べている．

4·4 SPARQL 検索

自動車が走行する際に，現在走行している場所から次に走行する場所を検索したり，走行している道路の制限速度を検索したりするために，SPARQL を用いて検索を行っている．例として，制限速度を検索するために利用した SPARQL 検索式を表 2 に示す．この検索式では，現在の経路上の位置として，道路と交差点のどちらでも利用可能なように記述している．

4·5 知的意思決定システム

上記で説明した，オントロジー，インスタンス，推論規則などを利用し，知的意思決定システムを作成した．規則に基づいて推論を実行するための推論器として，Pellet[Sirin 07] を用いた．このシステムは，信号のない交差点において，自動車が意思決定を行うためのシステムである．信号がない交差点においては，周囲の自動車などの状況に応じて，行動を決めなければならない．例えば，右折しようとした際に，前方から直進する自動車が来た場合には，その車は前方から来た車が通過するまで待たなければならない．そのような意思決定を行うシステムを構築した．

表2 SPARQL 検索式の例

```
SELECT ?max
WHERE { {
  currentPathSegment map:isLaneOf
  ?roadsegment.
  ?roadsegment map:isRoadSegmentof ?road.
  ?road map:speedMax ?max.
} UNION {
  ?road map:hasInstersection
  currentPathSegment.
  ?road map:speedMax ?max. } }
```

システムの処理の流れを図 6 に示す．図に従って順に処理を説明する．

(1) 自動車に搭載されたセンサから得られたデータは，センサデータ受信機を通して，SPARQL 検索エンジンと SWRL 規則推論器に送られる．
(2) オントロジーで記述された 3 種類のインスタンスと推論規則から構成される知識ベースを使って，現在の車線，隣の車線，運転方向などの自分の車の状況を検索する．
(3) SWRL 規則推論器が，他の車の速度や進行方向，衝突警告などの周囲の状況に関する情報を知識ベースに付加する．例えば，車 X が車 Y に衝突することを検出した場合には，<carX, control：collisionWarningWith, carY> というトリプルを知識ベースに追加する．
(4) SWRL 規則推論器が，知識ベースを使って推論を行い，推論結果を知識ベースに付加する．例えば，車Xが停止し,車Yに道を譲るという推論がなされる．
(5) SPARQL 検索エンジンが，知識ベースから実際に使われる運転コマンドを検索する．
(6) 運転コマンドが経路プランニングシステムに送ら

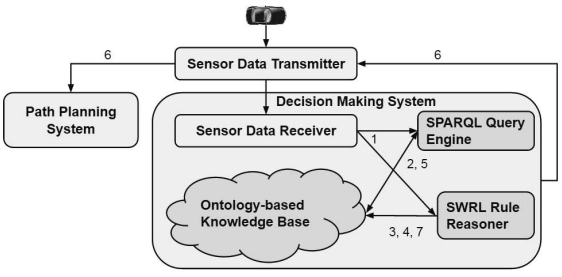

図6 知的意思決定システムの処理

れ，運転経路や運転動作が更新される．

(7) ステップ 3，ステップ 4 で新しく付加された情報が知識ベースから取り除かれる．

センサデータは，UDP を使ってリアルタイムでシステムに送られる．ここで使われるセンサデータは，緯度や速度などの情報が含まれている．トヨタの自動車，エスティマを使って，実装したシステムの実験を行った．実時間で推論が可能であるかを検証したところ，実行に要した平均時間は 99 ミリ秒であり，77 ミリ秒から 312 ミリ秒までの幅があったものの，交差点において，衝突を回避するような推論を行うことが可能であった．

5. おわりに

本稿では，セマンティック Web 技術を用いて，ストリームデータを処理するための枠組み，ストリーム推論について，例とともに解説を行った．多様なセンサデータを使って，複雑な意思決定を行う場合には，推論が必要となるため，本稿で述べたストリーム推論の枠組みは，大きな力を発揮するであろう．しかし，本稿では述べなかったが，ストリームデータに対して推論する枠組みは，セマンティック Web 技術だけに限られるわけではなく，解集合プログラミング [Gebser 12] を使うアプローチなど，別のアプローチもある．

ストリーム推論は，セマンティック Web において非常に注目を浴びているトピックの一つとなっており，国際セマンティック Web 会議などで頻繁にチュートリアルが開かれている．チュートリアルの資料は Web 上で公開 [Balduini 14] されており，さらにこの分野を深く知るための手掛かりになるであろう．また，2015 年開催の国際セマンティック Web 会議でも，ストリーム推論に関するチュートリアルが開かれる予定となっている．興味のある読者はぜひ，参加していただきたい．

謝辞

本稿の執筆にあたり，共同研究者で豊田工業大学の Lihua Zhao 博士，および，東芝の長野伸一博士，大阪大学の古崎晃司先生より助言をいただいた．記して感謝の意を表します．

◇ 参考文献 ◇

[Anicic 11] Anicic, D., Fodor, P., Rudolph, S. and Stojanovic, N.: EP-SPARQL: A unified language for event processing and stream reasoning, *Proc. 20th Int. Conf. on World Wide Web,* pp. 635-644, ACM (2011)

[Arasu 06] Arasu, A., Babu, S. and Widom, J.: The CQL continuous query language: Semantic foundations and query execution, *VLDB J.,* Vol. 15, No. 2, pp. 121-142 (2006)

[Balduini 14] Balduini, M., Calbimonte, J.-P., Corcho, O., Dell'Aglio, D. and Valle, E. D.: Tutorial on stream reasoning for linked data at ISWC 2014, http://streamreasoning.org/events/sr4ld2014 (2014) (参照 2015-06-29)

[Barbieri 09] Barbieri, D. F., Braga, D., Ceri, S., Valle, E. D. and Grossniklaus, M.: C-SPARQL: SPARQL for continuous querying, *Proc. 18th Int. Conf. on World Wide Web,* pp. 1061-1062, ACM (2009)

[Bolles 08] Bolles, A., Grawunder, M. and Jacobi, J.: Streaming SPARQL-extending SPARQL to process data streams, *Proc. 5th European Semantic Web Conf.,* Vol. 5021 of LNCS, pp. 448-462, Springer (2008)

[Calbimonte 10] Calbimonte, J.-P., Corcho, Ó. and Gray, A. J. G.: Enabling ontology-based access to streaming data sources, *Proc. 9th Int. Semantic Web Conf.* (1), Vol. 6496 of LNCS, pp. 96-111, Springer (2010)

[Gebser 12] Gebser, M., Grote, T., Kaminski, R., Obermeier, P., Sabuncu, O. and Schaub, T.: Stream reasoning with answer set programming: Preliminary report, *Proc. 13th Int. Conf. on Principles of Knowledge Representation and Reasoning,* pp. 613-617, AAAI (2012)

[Gutierrez 07] Gutierrez, C., Hurtado, C. A. and Vaisman, A.: Introducing time into RDF, *IEEE Trans. on Knowledge and Data Engineering,* Vol. 19, No. 2, pp. 207-218 (2007)

[Horrocks 04] Horrocks, I., Patel-Schneider, P. F., Boley, H., Tabet, S., Grosof, B. and Dean, M.: SWRL: A semantic web rule language combining OWL and RuleML, http://www.w3.org/Submission/ (2004) (参照 2015-06-29)

[Le-Phuoc 11] Le-Phuoc, D., Dao-Tran, M., Parreira, J. X. and Hauswirth, M.: A native and adaptive approach for unified processing of linked streams and linked data, *Proc. 10th Int. Semantic Web Conf.* (1), Vol. 7031 of LNCS, pp. 370-388, Springer (2011)

[Morph-Stre] Morph-Streams, https://github.com/jpcik/morphstreams (参照 2015-06-29)

[RDF Working Group14] RDF Working Group : Resource Description Framework (RDF), http://www.w3.org/RDF/ (2014) (参照 2015-06-29)

[Rodríguez 09] Rodríguez, A., McGrath, R. E., Liu, Y. and Myers, J. D.: Semantic management of streaming data, *Proc. 2nd Int. Workshop on Semantic Sensor Networks,* Vol. 522 of CEUR Workshop Proceedings, pp. 80-95 (2009)

[Sirin 07] Sirin, E., Parsia, B., Grau, B. C., Kalyanpur, A. and Katz,Y.: Pellet: Apractical OWL-DL reasoner, *J. Web Semantics,* Vol. 5, No. 2, pp. 51-53 (2007)

[Storm] Apache Storm, http://storm.apache.org/ (参照 2015-06-29)

[Valle09] Valle, E. D., Ceri, S., Harmelen, van F. and Fensel, D.: It's a streaming world! reasoning upon rapidly changing information, *IEEE Intelligent Systems,* Vol. 24, No. 6, pp. 83-89 (2009)

[Vincent 14] Vincent, P.: CEP Tooling Market Survey 2014, http://www.complexevents.com/2014/12/03/cep-tooling-marketsurvey-2014/ (2014) (参照 2015-06-29)

[Zhao14] Zhao, L., Ichise, R., Mita, S. and Sasaki, Y.: An ontology-based intelligent speed adaptation system for autonomous cars, *Proc. 4th Joint Int. Semantic Technology Conf.,* Vol. 8943 of LNCS, pp. 397-413, Springer (2014)

[Zhao 15] Zhao, L., Ichise, R., Mita, S. and Sasaki, Y.: Ontologies for advanced driver assistandce system, 人工知能学会セマンティックウェブとオントロジー研究会資料, Vol. SIG-SWO-035-03 (2015)

2015 年 8 月 3 日　受理

著者紹介

市瀬　龍太郎（正会員）

1995 年東京工業大学工学部情報工学科卒業．2000 年同大学院情報理工学研究科計算工学専攻博士課程修了．博士(工学)．同年から，国立情報学研究所助手，助教授を経て，2007 年より，同研究所情報学プリンシプル研究系准教授．総合研究大学院大学准教授を併任．機械学習，知識発見，知識共有などの研究に従事．電子情報通信学会，情報処理学会，各シニア会員．AAAI，日本認知科学会，各会員．