

SSN オントロジに基づくエッジ側での効率的な IoT データフィルタリング手法

徐運康^{†1} 岸知二^{†1}

^{†1} 早稲田大学 創造理工学研究科 経営システム工学専攻

1. 研究背景・目的

近年の IT 技術の進展の中、センサが小型で安価に提供されるようになり、それらが IoT (Internet of things) システムの主要な構成要素となっている。一般的には、IoT のアーキテクチャはエッジ側でセンサが環境を測定し、ゲートウェイがセンサデータを収集したり、加工したりする。その後、ゲートウェイから送られたデータ群をクラウド側で蓄積あるいはデータ分析する。エンドユーザはこのようなデータをスマホアプリケーションなどから利用する。このように複数のセンサで随時生成されるセンサデータをクラウドに集約するために、通信データ量が膨大となる。さらに、収集されたデータにはセンサの種類などに応じて異種なデータ群が混在している。従って、クラウド側での処理を考えると、それらの表現を標準化することが望ましい。

2. 従来研究

センサデータを標準化するために、データの記述に W3C の提案する SNS (Semantic Sensor Network) オントロジを用いる方法がある。SSN オントロジでは、Web 上の「リソース」を記述する RDF (Resource Description Framework) [1]などのトリプルという構造で、デバイス情報を表現している。図 1[2]の上部の点線枠は、「センサはデータを持つ」を表す RDF トリプルの例であり、メタデータと呼ばれる。一方、図の下部は具体的なデータを表している。メタデータとデータとの対応付けをデータマッピングと呼ぶ。

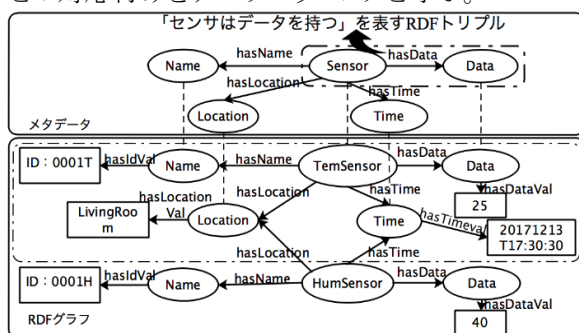


図 1. 温度と湿度データをメタデータにマッピング [2]

クラウド側とゲートウェイで SSN オントロジを利用することで、異種データを統一なフォーマ

ットで扱うことができる。さらに相互運用性が向上し、IoT アプリケーションの開発においてデータの取り回しも容易になる。

こうしたオントロジを用いてデータを処理する手法の研究として、例えば Mario ら [2] は、オントロジを利用して異種センサデータを標準なモデルにマッピングする手法を提案している。しかしこの方式では、元 JSON データより RDF など記述する通信データ量が増える可能性がある。

それに対して、Ahmed ら [3] の研究では、異種データ群を標準化するために、ゲートウェイ層で SSN などのオントロジに基づき、データにセマンティック注釈を付けてモデリングする手法を提案している。また、クラウドにあるルールオントロジをインスタンス化したルールでフィルタリングし、ゲートウェイ層にクラウドに送るデータ数を減らしている。

3. 提案手法

本研究では、[3]を踏まえ、さらに通信データを軽量化するための、ゲートウェイでのデータフィルタリング手法を提案する。RDF は、新規なトリプルを追加するなど、トリプルの変更によって構造の変化に柔軟に対応可能である。この特徴を利用して、エッジ側で、RDF で表現されたデータ中のトリプルから差分のみを取り出しクラウドに送信し、クラウド側でそれを補完するという方法で軽量化を行う。

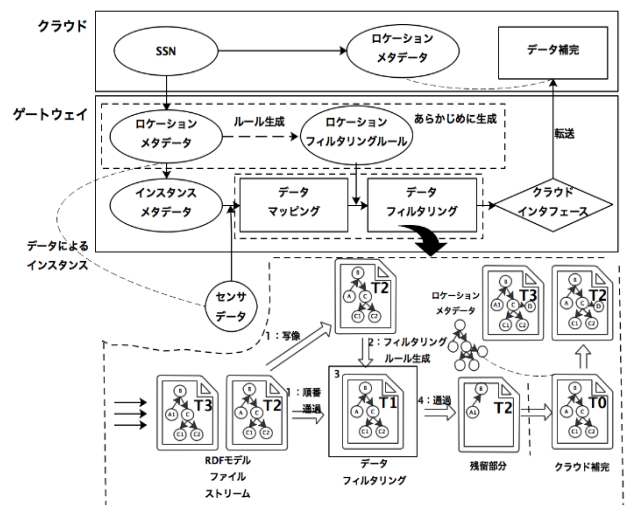


図 2. データ処理アプローチ

図 2 は、1 章で説明した IoT アーキテクチャを踏まえて提案手法を示したものである。以下、説明を行う。

1. ゲートウェイに接続されたセンサに関わるオント

An Efficient IoT Edge-side Data Filtering Method based on SSN Ontology

^{†1} Department of Industrial and Management Systems Engineering CSE Graduate School, Waseda University

ロジ部分をクラウドから取り出す。ここに定義されるメタデータをロケーションメタデータと呼ぶ。これに基づきインスタンスメタデータとロケーションフィルタリングルールを得る。

2. センサから得られたデータをインスタンスメタデータにマッピングする。ゲートウェイがクラウドに転送する際に、ロケーションフィルタリングルールによってフィルタリングする。
3. クラウド側は、前データやロケーションメタデータに基づいて不完全なデータを補完する。

図2の下部はフィルタリング過程のデータストリームを示している。ここでは、前のデータからの差分のみを取り出す。図では、前のデータ T1 と比較して、T2 は変化が無く、T3 は変化していた状況を示している。この場合、T1 のデータスキーマ A1 と B に対応する部分のみが差分となる。

差分を受け取ったクラウド側では、欠落しているスキーマ部分は前のデータやメタデータを用いて補完することで、データ T2 や T3 を再現する。

4. 例題

提案手法を実証するために、Linux OS(Operation System)を標準搭載した Edison というデバイスを用いて室内環境測定 IoT システムを構築した。このシステムは温度や湿度センサデバイスは定期的に(1秒ごと)データを取得して、Edison デバイスのマイコンをゲートウェイとしてアマゾンウェブサービス(AWS)のクラウドに転送する。

図1のような IoT 分野での利用を目的とした SSN オントロジを用いて、メタデータを定義し、インスタンスの温度や湿度センサデータを RDF および OWL で表現した。(図3)

```
<?xml version="1.0" encoding="UTF-8"?>
<rdf:RDF
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
  xmlns:owl="http://www.w3.org/2002/07/owl#"
  xmlns:Tem="http://localhost/TemperatureSensorOntology#"
  xmlns:Hum="http://localhost/HumiditySensorOntology#"
  >
  <owl:Ontology rdf:about="TemSensor">
    <rdfs:hasName>Name</rdfs:hasName>
    ...
    <rdfs:hasData>Data</rdfs:hasData>
    <rdfs:hasTime>Time</rdfs:hasTime>
  </owl:Ontology>
  ...
  <Hum:hasData>Data</Hum:hasData>
</rdf:RDF>
```

図3. 温度ロケーションメタデータの RDF ファイル一部

図4に一部の温度や湿度 JSON ファイルを RDF ファイルへの変換(データマッピング)を示す。ここでは、図1で示したマッピング方法で、データを構造化(セマンティック注釈)し、図2で示したフィルタリング方法で不要なトリプルを削除している 図4の下部に本実験のフィルタリングルール一部を示す。

図4の線枠内に、フィルタリングの結果から得られる残留部分を示す。このように軽量化されたデータをクラウドに転送する。クラウド側では、図3のようなロケー

ションメタデータや前のデータに基づき、データを補完し、RDF ファイルを再現する。それらのデータに基づき、ユーザへの情報提供、冷暖房機器との連携などのアプリケーションを実現する。

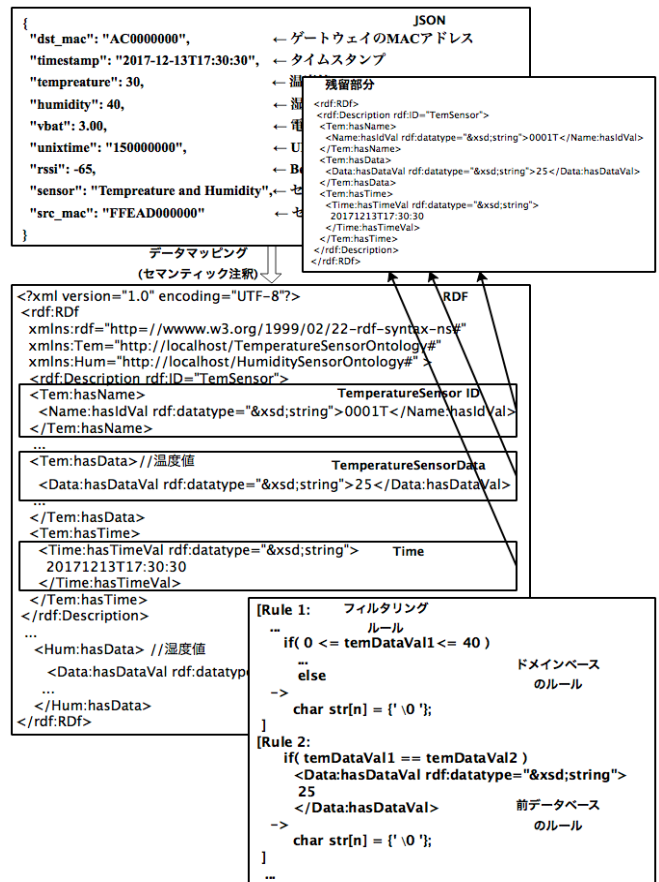


図4. データマッピングとフィルタリングしたファイル

5. まとめ

本稿では SSN オントロジに基づくエッジ側で異種データ群を RDF など標準化しフィルタリングする手法を提案した。本手法により、クラウドへの送信データの軽量化や、異種データの扱いの容易化が期待される。定量的な評価は今後の課題である。

参考文献

[1] O. Lassila and R. R. Swick: "Resource description framework (rdf) model and syntax specification" (1999). available at <http://www.w3.org/TR/1999/REC-rdf-syntax-19990222>.

[2] Buchmayr, M., Kurschl, W., Küng, J. "A rule based approach for mapping sensor data to ontological models in AAL environments", In: Castano, S., Vassiliadis, P., Lakshmanan, L.V., Lee, M.L. (eds.) ER 2012. LNCS, vol. 7518, pp. 3–12. Springer, Heidelberg, 2012.

[3] Ahmed Bali Email, Mahmud Al-Osta, Gherbi Abdelouahed. "An Ontology-Based Approach for IoT Data Processing Using Semantic Rules", SDL 2017: SDL 2017: Model-Driven Engineering for Future Internet pp 61-79, 2017.