

IoT 相互運用に向けた ATL によるデータ変換技術の提案

藤原 浩司[†] 山下 蘭[†] 岩政 幹人[†] 長野 伸一[†]株式会社 東芝 研究開発センター[†]

1. はじめに

産業用 IoT では、独立した複数のシステムを連携することで新たなサービスや価値を創造することが期待されている。しかし、各システムが取り扱うデータは形式や構造が異なるため、それらを相互に理解可能にする相互運用技術が不可欠である[1]。各システムが持つデータは、データの構造や関係等を定義するデータモデル、およびデータモデルのモデルとなるメタモデルという階層構造を持った情報モデルで表現される。メタモデル間の対応関係から作成した変換ルールを使うことで、個別のデータに依存しないデータ交換が可能となる。本稿では、モデルベースの手法による IoT 相互運用の実現性を検証するために、モデル変換言語 ATL(Atlas Transformation Language)[2]を用いたデータ変換について確認し、得られた課題と解決のための手法について考察する。

2. ユースケース

今回、自動車から取得されたデータをスマートシティのシステムに接続し、各道路の交通状況を管理するユースケースを考える。各自動車が持つ情報モデルとシステムが持つ情報モデルが異なる場合、相互に接続することができない。そこで、異なる情報モデルのデータ同士を変換することでデータ交換を実現する、図 1 に示すようなデータ変換ゲートウェイについて考える。

情報モデルの例として、スマートシティの情報モデルである FIWARE や Connected Car の計測データを定義する ISO 22837/ISO 14817(以降 Connected Car)、産業 IoT 分野の OPC UA(OPC Unified Architecture)[3]などがある。FIWARE と Connected Car では、どちらの情報モデルにも自動車に関するデータ定義が含まれるため、これらのデータを相互にやり取りしたい。しかし、属性名やデータ構造、値の定義が異なるため、単純にデータをやり取りすることはできない。例えば、位置情報は Connected Car では緯度、経度、高度が個々の属性として定義されているが、FIWARE では 1 つの配列でまとめる geo:json 形式を採用している。速度情報では単位が Connected Car は m/s、FIWARE は km/h と異なっている。

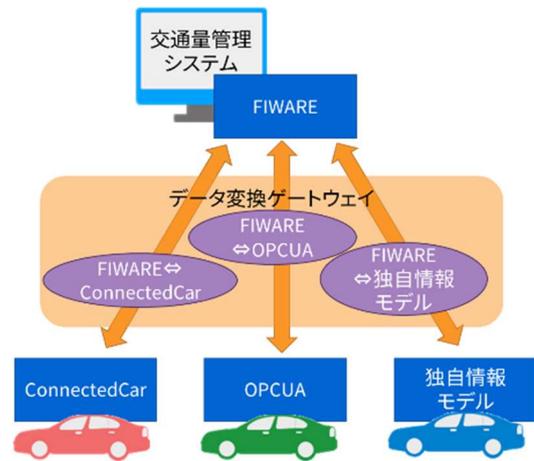


図 1 データ変換ゲートウェイ

3. 変換ルールによるデータ変換

メタモデルはインスタンス(個別のモデルとデータ)には依存しないため、メタモデルを用いて変換ルールを作成することで、異なるインスタンスに対しても変換ルールを再利用できる。

変換ルールを記述するための言語、および変換処理のためのフレームワークはいくつかあるが、今回は Eclipse プロジェクトの一部である ATL を選定した。ATL では、ecore 形式のメタモデルファイルと各種データフォーマットのインスタンスファイル、at1 形式の変換ルールを用いてデータ変換を実現できる。

Connected Car と FIWARE のデータ交換を行うため、2 章で示した位置情報と速度情報に対し ATL で変換ルールを作成する。ATL では、メタモデル間の属性の対応を記述する形式で変換ルールを作成する。その際に、属性の名前や構造の変更も指定できる。位置情報のように複数の属性を 1 つの配列にまとめる場合、Sequence という予約語を用いる(図 2)。速度情報のような単位変換は元の値に対する演算として記述する(図 3)。

ATL フレームワークを使用し、xml から json へのデータ変換処理やメタモデルファイル、変換ルールファイルを作成し、変換を実行した。変換の入力と出力のインスタンスのファイル内容を図 4 に示す。また、変換の処理時間を表 1 に示す。位置情報、速度情報の各変換ルール、両者を含む変換ルールの 3 パターンに対し、ファイル読み込みなどを含む全体の実行時間および変換処理自体の時間を各々 100 回計測し平均を取った。

A study on achieving Data Interoperability in IoT with ATL
[†]Koji Fujiwara, Lan Yamashita, Mikito Iwamasa,
 Shinichi Nagano, Corporate R&D Center, Toshiba Corporation.

```
point:Fiware!Point(
  type <- 'Point',
  coordinates <- Sequence[
    connectedCar.sensing.longitude.degree,
    connectedCar.sensing.latitude.degree,
    connectedCar.sensing.altitude.altitude
  ]
)
```

図 2 位置情報の構造変換ルール

```
helper def : transMsToKmh(ms : Integer) : Integer =
  ms * 3.6;

veh:Fiware!Vehicle(
  speed <- thisModule.transMsToKmh(
    connectedCar.vehicle.velocity.velocity),
  location <- geo
),
```

図 3 速度情報の単位変換ルール

```
<sensing>
  <latitude>
    <degree>35.67</degree> "vehicle": {
  </latitude>                "speed":54.0,
  <longitude>                "location": {
    <degree>139.65</degree>    "type": "geo:json",
  </longitude>                "value": {
    <altitude>                "type": "Point",
    </altitude>                "coordinates": [
    <altitude>40</altitude>    139.65, 35.67, 40
    </altitude>                ]
  }
  </sensing>
</vehicle>
  <velocity>
    <velocity>15</velocity> }
  </velocity>
</vehicle>
```

図 4 入力ファイル(左)と出力ファイル(右)

表 1 変換処理実行時間

単位:ms	位置情報	速度情報	両方
変換のみ	13.7	15.2	19.9
処理全体	218.9	220.5	408.6

4. 結果と考察

ATL を用いたデータ変換では、対象となるメタモデルを増やすためにはメタモデルファイルと変換ルールを作成する必要がある。相互運用性の実現のためには双方向の変換が必要であるが、ATL で実現するためには逆方向の変換ルールの作成も必要であるため、変換ルールの作成コストが高い。また、IoT 分野ではリアルタイム性が求められることが多いが、表 1 から ATL での変換は数百ミリ秒オーダーの時間がかかるため、性能改善が必要である。これらのうち、メタモデルファイルについては xml スキーマなどの定義ファイルから自動変換するプラグインが提供されており、性能については各データをメモリ上に展開したままデータ変換を行うことで問題点を改善できる。本稿では課題として変換ルールの作成コストを取り上げ、その解決のために変換ルールを自動生成する手法について考察する。

変換ルール作成の手順は(1)メタモデル同士で変換可能な属性を見つけ、(2)属性同士で構造や値の変換が必要な場合に変換ルールを記述するという 2 段階に分解できる。特に(1)では、メタ

モデルごとに名前が同じでも意味の異なる属性や、逆に意味が同じで名前が異なる属性が存在することが問題となる。この問題点を解決し、自動的に属性同士の関係性を判別するために、オントロジーを使った手法を提案する。オントロジーは、一意なリソース同士の関係性を定義したものである。そのため、属性同士の関係性を定義した属性オントロジーがあれば、各メタモデルの属性と属性オントロジー上のリソースと対応づけることで、属性同士の対応関係をオントロジーへの問い合わせで解決することが可能である。(2)についても、属性オントロジー内に ATL であらかじめ作成しておいた変換ルールを定義しておくことで解決可能である。特に、逆方向の変換ルールに関しても合わせて定義しておくことで、属性オントロジーへの問い合わせで双方向の変換ルールを得ることができる。

オントロジーと変換ルールを使ってシステム同士を接続する研究としては[4]がある。オントロジーとして表現されたインスタンス同士を変換する手法であり、json や xml などの一般的な形式のデータを直接扱うことができない。一方、本研究で提案した手法はルール生成の時だけ、あらかじめ作成された属性オントロジーを使い、変換処理は ATL によって実行されるため、任意のデータに対して適用可能である。

5. まとめ

本稿では、モデルベースの手法による IoT 相互運用の実現性を検証するために、モデル変換言語 ATL をデータ変換結果について確認した。特に変換ルールについて作成コストが高いことを明らかにし、その解決策としてオントロジーを用いた変換ルールの自動生成手法について考察した。今後は変換ルールを含む属性オントロジーを作成し、自動生成の具体的な手法を検討する。

参考文献

[1] Lin, Jie, et al., "A Survey on Internet of Things: Architecture, Enabling Technologies, Security and Privacy, and Applications." IEEE Internet of Things Journal vol. 4, no. 5, pp.1125-1142, 2017.

[2] "ATL," <https://www.eclipse.org/at1/>.

[3] "OPC Foundation," <https://opcfoundation.org/>.

[4] Santodomingo, Rafael, et al., "Rule-Based Data Transformations in Electricity Smart Grids." International Symposium on Rules and Rule Markup Languages for the Semantic Web. Springer, Cham, 2015.