令和5年度　秋学期　卒業研究3 (BI)
学士論文

題目 IoTシステムの水平同期性異常検知のためのRDFグラフ生成

指導教員 上山憲昭 教授

立命館大学 情報理工学部 セキュリティ・ネットワークコース

学籍番号 26002005254
Park Jooinh

令和6年1月31日

概要

今の IoT システムは実世界を観測し変容を促す一連の過程が正しく実施されたことを証明する手段として，ブロックチェーン等を活用しデータの改ざんを防ぐ方法が注目されている．しかし，IoT システムから実世界に配備されたデバイス/ネットワークへの攻撃や異常の検知は困難であり，実世界の状態の誤解により深刻な被害が発生する危険性が指摘されている．本研究では複数の IoT システムと実世界との同期性を解析し，高度な IoT セキュリティを実現する IoT システム連携制御基盤の実現を目的とする．具体的には，複数の IoT システムが実世界の環境/状態を観測した結果の共通性（水平同期性）を解析することで異常要素を特定できる水平同期性異常検知アーキテクチャを提案し，それに必要な Semantic-Rich な RDF グラフの生成を行う．

目 次

第1章　序論

1.1　研究背景

デジタルトランスフォーメーションの拡大に伴い，センサーデバイスを使いあらゆるものを幅広く観測し，フィードバックにより実世界に影響を与えるIoT（Internet of Things）システムはより広く使われるようになっている．よって，未来では今より遥かに多くのIoTシステムが存在し，センシング能力は違っていても同じ実世界の変容を同時に観測できる状況が想定できる．例えば，GPSによって車両の位置を特定する位置データとその車両を認識した防犯カメラの位置データがそれにあたる．一つのシステムではわからない情報や見逃していたエラーが複数システムの共通した実世界変容の観測により水平同期性から確認できるようになればIoTシステムの更なる拡大と大規模IoTシステムの構築への一助になると考える．

IoTシステムのデータの信頼性はデータの完全性から来ており，データを分散管理し改ざんを防ぐことができるブロックチェーン技術が注目を集めている．

1.2　研究目的

本稿ではIoTシステム同士を連携させることにより，データ改ざん防止等では対処できないデータのエラーに対処できる高信頼IoTシステム連携制御基盤を提案している．

この高信頼IoTシステム連携制御基盤大きく分けて二つの機能を有している．一つは図1.1に示す異常検知モデルである．異常検知モデルは異常検知はIoTシステムから収集したデータとそのメタデータとなるセマンティックデータを解析し，共通した事象に対する観測において不均一な値を示している箇所（水平同期性の乱れ）を特定する．

もう一つは図1.2で示す正常化モデルである．正常化モデルは異常検知モデルで特定した構成要素を除外した全体構成をブロックチェーンに登録されている正常モデルと比較しその差を埋める役割を担っている．異常であると判断されたデータやデバイスはそのまま放置されず，修正後のデータとして最も相応しいデータに置き換えることができるようにすることで自動化された高信頼アーキテクチャが成り立つ．

本稿では高信頼IoTシステム連携制御基盤の全体ではなく前半部分の異常検知モデルに集中し，再構成モデルの詳細については触れないものとする．

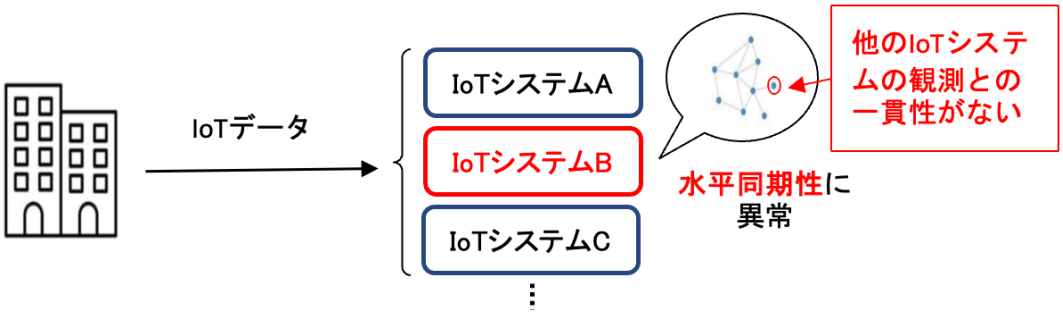

図 1.1: 異常検知モデル

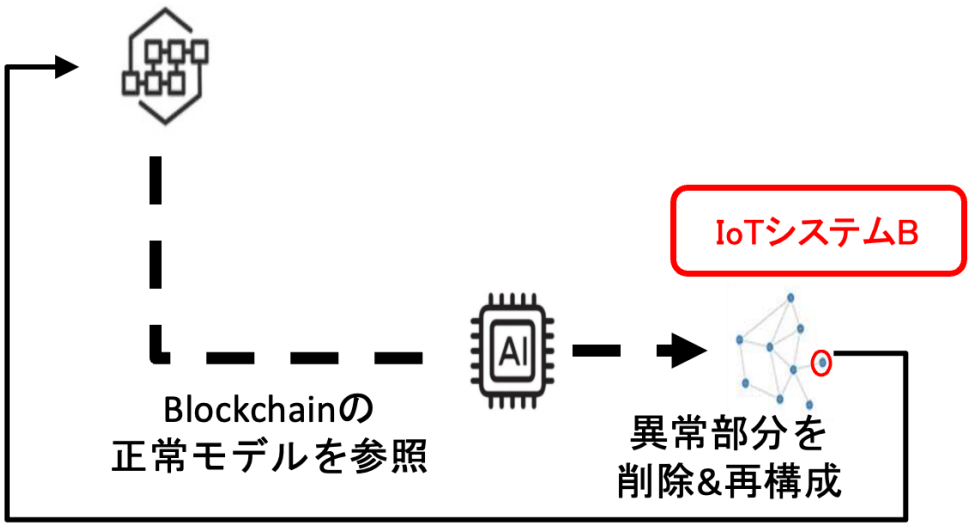

図 1.2: 正常化モデル

第2章　関連研究

2.1　Semantic Interoperability

IoT プラットフォームは実世界の観測と変容において非常に重要な役割を持っている．しかし，IoT プラットフォームはそれぞれ違うプロトコル，デバイスデータフォーマット，通信環境，使用されている技術，ハードウェア等の問題によりプラットフォーム内での垂直なコミュニケーションしか取ることができないことが多い．これにより，大規模な IoT システムの展開や小規模な IoT システムの連携が非常に難しい状態である．

よって，IoT システム同士の連携のために Semantic Interoperability(セマンティック相互互換性) が様々な方面で研究されている．これらの研究は大まかに 2 つに分かれている．一つのは特定の単一 IoT デバイスに対する Semantic Interoperability で，もう一つは不特定の IoT デバイス同士が透明性のあるメカニズムを用いて作用しあえるエコーシステムを提案したものである．

特定の単一デバイスに対するアプローチの場合の多くは共通の API，データフォーマットとデータモデルを要求する．デバイスはそれらの仕様に合わせたアダプターレイヤーを使う等のやり方を取ることになる．しかし，それぞれの IoT デバイスの仕様を微調整し合わせるやり方では様々な IoT システムを組み込んだ大規模な連携制御基盤のアーキテクチャに組み込むことはできない．

よって，本稿では複数の IoT システムに対する Semantic Interoperability を確保するためのアーキテクチャを提案する．

2.2　水平同期性

水平同期性とは実世界の変容を観測したそれぞれ違う IoT システム同士が矛盾のないデータを収集する状態を指す．現在の IoT の用途から見てこのような状況は少ない．何故なら同じ対象や場所に対して同じ時間に同じようなデータを収集する IoT システムが個別的に点在する理由がないからである．現在の IoT システムは冗長を消し，特定目的に適したシステムのみを構築することに集中しているが，IoT の用途が更に拡大され，ユビキタス（Ubiquitous）な状態になれば計画都市のように必要最低限の IoT デバイスだけを設置した環境ではなく様々な製造元・管理者を持ったお互いに自立した IoT システムが溢れかえる環境になると考える．そのような環境において大規模な IoT システムの高信頼連携基盤を作るために大規模なデータから異常検知の精度を確保できる水平同期性に注目した．

水平同期性を基準にした異常検知は外部からの攻撃ではなくデータの異常そのものを検知する．これは既存の改ざん防止を元にした異常検知では対処できないエラーに対応でき

るようにIoTシステムが収集したデータの特徴やデバイス・システム全体の情報を元にした異常検知モデルの実現を可能にする

第3章　提案方式

3.1　GCNN 異常検知

3.1.1　概要

本稿で提案する連携制御基盤は IoT データと Semantic Knowledge の２つを入力とする，Transformer を使った GCNN を使い IoT システムのモニタリングを行う．

AI を使った IoT のモニタリングと異常検知では IoT データかドメイン知識に偏って解析を行う．何故なら Edge analytics のアプローチ [4] では二つ同時に扱い全体的な解析するのは難しいためである．そこで既存研究 [3] ではどんな IoT システムトポロジーにも対応でき，ドメイン知識・Semantic Knowledge を考慮しつつ IoT データを解析できるように GCNN（Graph Convolutional Neural Network）を使っている．また，GCNN でのシーケンス処理速度を大幅に向上させるためにシーケンスデータを一括で処理を可能にするため Attention メカニズムを使った Transformer[5] を組み込んだ上で特定の単一 IoT システムのモニタリングを行っている．本稿で提案するアーキテクチャでは Semantic Knowledge を複数の IoT システムのものに置き換えることで複数の IoT システムのモニタリングを可能にする．

3.1.2　異常検知の原理

GCNN と Transformer を使った異常検知モデルは IoT データと Semantic Knowledge の２つの入力の現在時刻 $t-1$ までのデータを使い現在時刻 t の IoT データの Transformer 出力を予測する．そこで予測演算された現在時刻 t と実際に取得した現在時刻 t の Transformer の値が異なっている場合水平同期性が乱れていると判断し，異常として検知する．GCNN と Semantic Knowledge を使うことで Transformer への入力と出力の間の複雑なマッピングが可能になっているため，異常検知の原因となる箇所をグラフから求めることができるため，高信頼 IoT 連携制御基盤の異常箇所再構成機能に繋げることができると考えている．

GCNN の式 [9] は以下の通りである．

$$f(X, A) = \sigma(\hat{A}ReLU(\hat{A}X, W^{(0)})W^{(1)}) \tag{3.1}$$

X はノードレベルでの Feature マトリックス，A は隣接マトリックス，$\hat{A}$ は前処理段階で，$\hat{D}, W^{(0)}, W^{(1)}$ は隣接ノードのマトリックス強度を示す．

3.2 Semantic Extractor

3.2.1 概要

既存研究では GCNN の入力として必要な Semantic Knowledge はロボットアームの構造グラフが使われている．本稿ではロボットアームの代わりに複数の IoT システムとその中に含まれるデバイス間の関係を表す Semantic Knowledge グラフである．特定 IoT システムだけを対象にした場合，Semantic Knowledge グラフはシステムの外で把握した情報を基に手動で入力されることになる．しかし，多種多様な IoT システムを対象としたモニタリングシステムにおいてそれぞれの IoT システムを調べ手動で関係図を導くことはできない．よって，Semtantic Extractor は IoT データと Semantic データを基にデバイス同士の関係とシステム間のリンクを作り，Semantic-rich な RDF(Resource Description Framework) グラフを生成する．

3.2.2 RDF グラフ

RDF グラフは Semantic Interoperability で最も重要な Semantic Knowledge を取得し入力するために使われる．本稿で提案するアーキテクチャでは複数の IoT システムのデータが一つの大規模なシステムから来ているように扱うことでおとがいを比較し，異常な要素を見つことにある．Semantic Interoperability は IoT システムがもつデバイスの観測データだけでは達成できない．RDF から観測データ以外のメタデータ等を使いシステム内部でのデバイス同士の関係，他 IoT システムとの関係等の情報把握することができる．GCNN で使われた単一 IoT システムの構造図の代替できる Semantic Knowledge として使用するためにグラフ構造の RDF を選択する．

3.2.3 グラフ生成の原理

Semantic Extractor は大きく 3 つの部分に分かれている．Syntax Processor，Semantics Annotator と Correlation Analyzer である．まず，Syntax Processor は IoT データと Semantic データからスタブグラフを生成し，データの種類を判別した後，初期の RDF グラフを生成する．既存研究では REST API にのみ対応した REST Crawler を使っていたが，本稿では複数の大規模 IoT システムに対応できるようにするため SPARQ Endpoint を使い情報を取得している．次に，Semantics Annotator はキーワードマッチング，クラスとドメイン範囲，DogOnt[6] を使いデバイス間・システム間のリンクを形成する．最後に，Correlation Analyzer はデータの変動パターンからリソース同士の関係を形成する．

3.3 eWoT

3.3.1 概要

Semantic Extractor から Semantic-rich な RDF グラフを生成するためには IoT システムから Semantic データを取得する必要がある．センサーデバイスのデータだけではデバイ

ス同士の関係を構築するための材料がないからである．しかし，お互いに互換性を持たない IoT システム同士では数値データは得られても意味のあるドメイン知識を得ることはできない．また，多くの IoT システムを対象にした連携制御基盤を作る上では互換性の問題を克服できる自動化された手段が必要になる．よって，それぞれの IoT システムから手動で Semantic データを貰うわけにはいかない．そこで，本稿では eWoT を使い TD(Thing Description)[7] を拡張した WoT-Mapping[1] を通してあらゆる IoT デバイスのデータを RDF に変換する作業を行う．

3.3.2 原理

IoT システムが SPARQL Endpoint を使っている場合 eWoT は直接 SPARQL クエリーを実行し，エコーシステム内のすべてのデバイスと連合する．しかし，SPARQL Endpoint が活性化されてない場合は IoT センサーのすべてのメタデータを Triple Store[10] に保存し，順次新しいデータを Triple Store にプッシュすることで本アーキテクチャ内で SPARQL クエリーを実行しお互い通信できるようにすることで IoT システム同士の互換性問題を解決する．

よって，eWoT に新しいデバイスを追加するとき，WoT マッピングを含めた TD を登録するだけで IoT Interoperability を確保することができる．

第4章　実験

4.1　前提条件

本稿の提案する高信頼 IoT 連携制御基盤は大規模な IoT システム同士の連携を想定しているため，膨大なデータを扱うことになる．しかし，現段階では実際の IoT システムの規模のデータを扱った実験は難しいため，このアーキテクチャが想定している IoT システムが今より遥かに普及している環境を模倣するためにデータを手動で入力し，重複部分を作る．既存研究 [2] で使われた openHAB, EclipseIoT と Datavenue のデータを模倣し，eWoT と連携して再現することで実験を行う．

また，このアーキテクチャは多くの重複データから学習し，蓄積されたデータと異なる動きを察知する仕組みである．この仕組みでは小規模の実験では些細な誤差もエラーとして検出するため,「共通する地域を対象とした複数の IoT システムは，同じ実世界の変容を同時に観測する」ということを前提にエラーを検出している．

4.2　グラフ生成

まず，簡単な IoT システム 3 つのデータの SPARQL Enpoint を用意する．eWoT を使ったデータ解析に SPARQL Endpoint は必須ではないが，REST Entrypoint の代わりとして用意する．用意したデータは Room1 と Room2，そしてその間を繋ぐ Corridor の 3 つの異なる IoT システムを模倣している．

まずは SPARQL のクエリーを使い図 4.1 のようなスタブ RDF グラフを生成する．スタブ RDF グラフは IoT システム間でのつながりはない状態でそれぞれの IoT システムの構成を簡単に示すグラフで，ここに Semantic 情報を追加していくことで最終的な Semantic-Rich な RDF グラフを求める．

次に eWoT から取得する Semantic データを使いデバイスとデータの種類を判別し記入する．WoT オントロジーを基に WoT マッピングを行う上で Things Description の他に相互作用パターン，エンドポイントから分かるウェブ上の位置，そしてデータスキーマからドメイン知識を抽出することでデバイスがリソースタイプかを判別し，デバイス同士にリンクが存在する場合結びつける．ここで使われるモジュールは Ontology Matcher で，WoT オントロジーのデータからキーワードマッチングを行いデータの種類を判別する．Semantic Matching アルゴリズム [8] を使い RDF Predicate とオントロジークラスをマッチングする．図 4.2 では Room1 と Presence Sensor，Door Actuator の間にリンクが存在することを示し，それぞれがリソースタイプであることを記入している．

Semantic Reasoner は最初にオントロジーのクラスと Property を参照しリンクの種類に関する情報を追記し，追記した情報を基に数回論理的推定アルゴリズムを使い関係図を

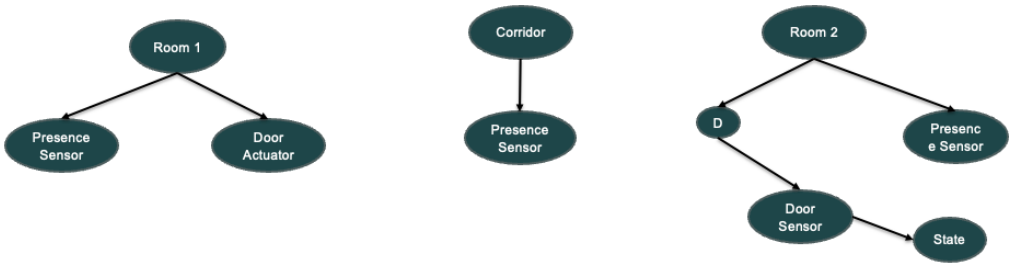

図 4.1: Syntax Extractor から生成した初期 RDF グラフ

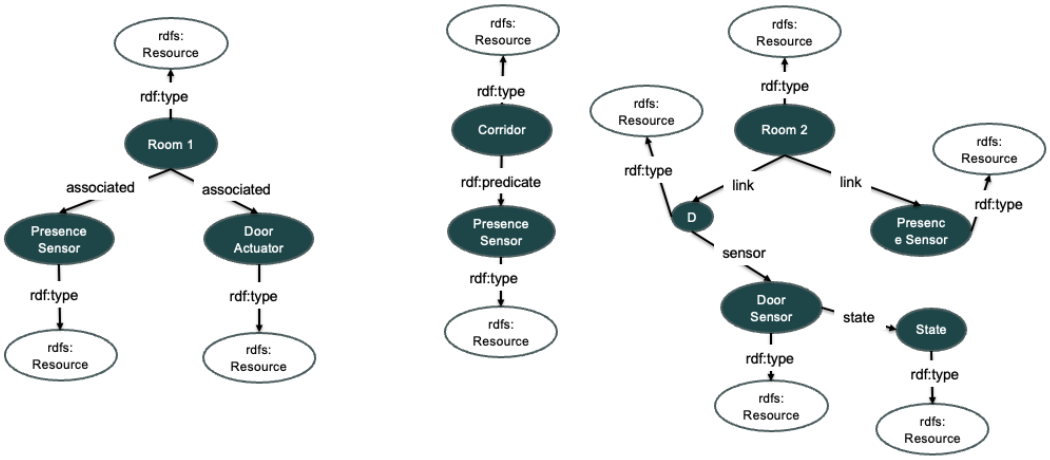

図 4.2: Ontology Matcher で生成した RDF グラフ

広げていく．最初に Presence Sensor と Door Actuator が Room1 に所属していることを判明し，図 4.3 では Room2 の状態についても導いている．最後に，デバイスだけでなく Room2 とそこに所属する扉 D との関係も求まる．

1) Transitive Property: $p(x,y) \wedge p(y,z) => p(x,z)$;
2) Symmetric Property: $p(x,y) => p(y,x)$;
3) Functional Property: $p(x,y) \wedge p(y,z) => y == z$;
4) InverseOf: $p1(x,y) => p2(y,x)$;
5) Inverse Functional Property: $p(y,x) \wedge p(z,x) => y == z$;

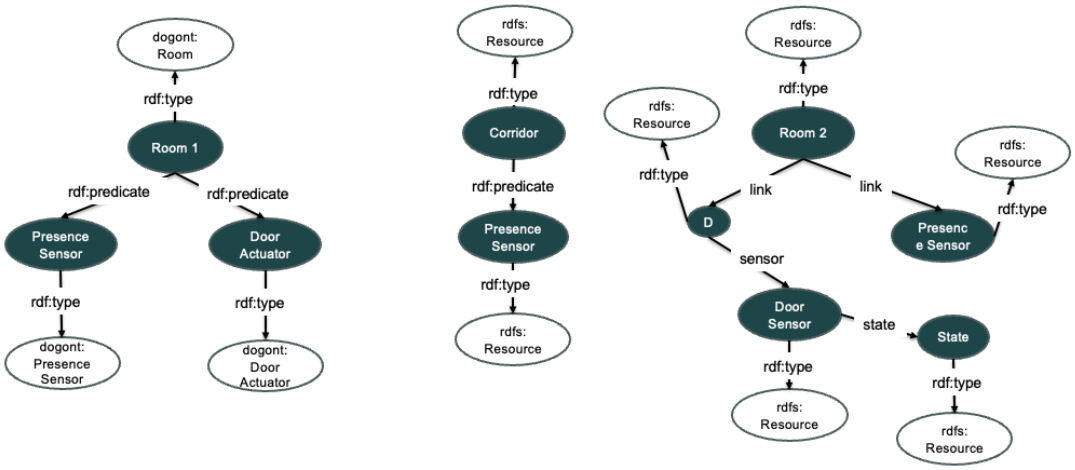

図 4.3: eWoT の Semantic データと Semantic Reasoner から生成した RDF グラフ

最後に，Classifier と Correlation Analyzer ではシステム同士のリンクを結び定義する．図 4.4 のグラフにおいて Room1 と Corridor，そして Corridor と Room2 の間に何らかのリンクが存在することは Correlation Analyzer がそれぞれのデータの流れと変動値を解析することで導くことができる．更に，図 4.5 では Classifier の事前学習されたオントロジー分類アルゴリズム [11] を使い関連クラスを更に詳細な関係に定義する．結果的に Room1 と Corridor，Corridor と Room2 が物理的につながっていることを導く．

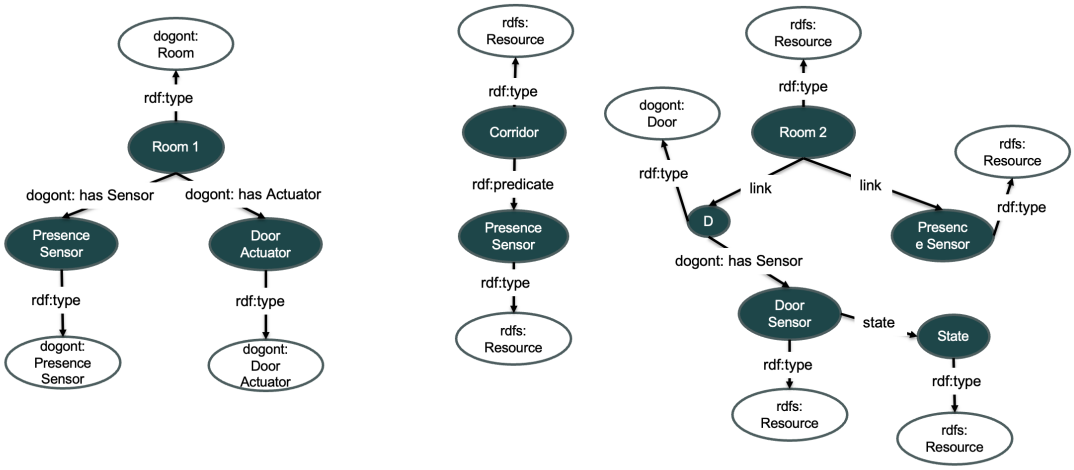

図 4.4: Ontology Matcher の結果

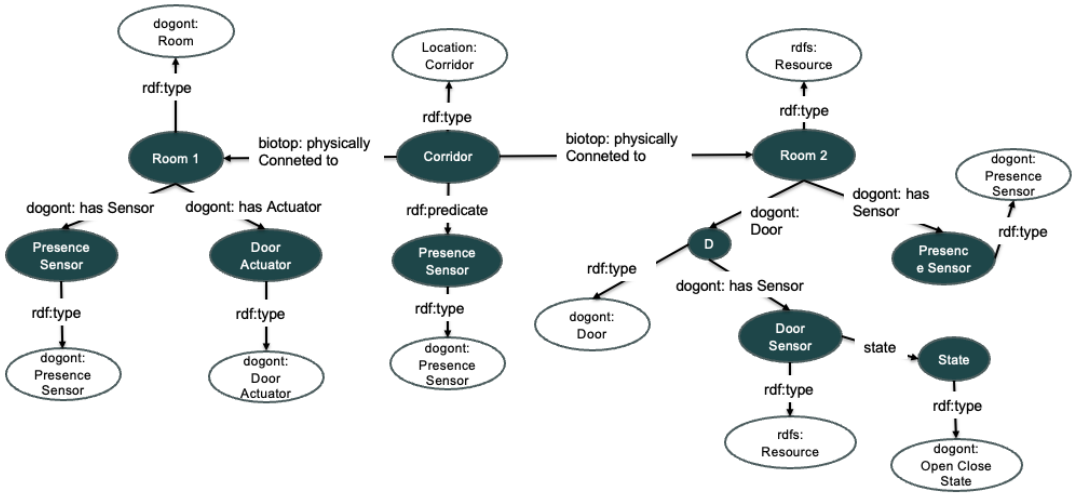

図 4.5: Semantic Extractor の最終結果

第5章　まとめ

本稿ではIoTシステムが更に普及しそれぞれ異なるIoTシステム間でデータの重複が発生する社会でIoTシステム間のデータの連携を可能にすることでモニタリング・異常検知を行える高信頼IoT連携制御基盤を提案し，その実現に必要なSemantic-richなRDFグラフの生成を行った．大規模な実験を行うことはできなかったため既存研究のデータを模倣する形で本稿で提案するアーキテクチャが同じ形のRDFグラフをより多くのIoTシステムから抽出して生成することができることを示し，GCNNを用いた複数のIoTシステムの同時モニタリングと異常検知の可能性を示唆した．

IoTシステムの未来として理想的なのはどのIoTシステムも制限なくお互いに通信・連携し，リアルタイムで情報を共有できるようになり，社会全体を支える大規模なシステムとして機能できるようになることである．しかし，現在のIoTデバイスは特定目的に限定的に使われており，システム連携を考慮できる環境には至っていない．よって，よりIoTシステムを普及させる第一歩としてデータの連携を可能にすることに大きな意味があると考える．IoTシステムの稼働が汎用的な価値を持つようになることにより，IoTシステムを設置し維持することに価値が生じることになる．一般的IoTシステムが日常においてより普及されれば今後の完全なIoT相互互換性研究にも大きく役立つと確信する．

謝辞

本研究を行うに当たり，ご指導を頂いた上山教授に感謝します. また日常，有益な議論をして頂いた研究室の皆様に感謝します.

参考文献

[1] A.Cimmino, M. Poveda-Villalón, R. García-Castro, eWoT: A Semantic Interoperability Approach for Heterogeneous IoT Ecosystems Based on the Web of Things.

[2] W. Li, G. Privat, F. Le Gall, (2017), Towards a Semantics Extractor for Interoperability of IoT Platforms

[3] A. Ba, F. Lorenzi and J. Ploennigs, Monitoring of IoT Systems at the Edges with Transformer-based Graph Convolutional Neural Networks

[4] M. Chiang and T. Zhang, "Fog and IoT: An overview of research opportunities," IEEE Internet of Things J., vol. 3, no. 6, pp. 854–864, 2016.

[5] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. u. Kaiser, and I. Polosukhin, "Attention is all you need," in Advances in Neural Information Processing Systems, vol. 30, 2017.

[6] http://smartcity.linkeddata.es/ontologies/elite.polito.itontologiesdogont.owl.html

[7] Kaebisch, S.; Kamiya, T. Web of Things (WoT) Thing Description; First Public Working Draft, W3C: Cambridge, MA, USA, 2017.

[8] S. Khan and M. Safyan, "Semantic Matching in Hierarchical Ontologies," Journal of King Saud University - Computer and Information Sciences, vol. 26, no. 3, pp. 247–257, Sep. 2014.

[9] T. N. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," in Int. Conf. on Learning Representations, 2017.

[10] https://www.ontotext.com

[11] J. Gao and S. Mazumdar, "Exploiting Linked Open Data to Uncover Entity Types," presented at the Semantic Web Evaluation Challenge, 2015, pp. 51–62.

学会発表リスト

- 朴柱仁, 上山憲昭, “ IoT システムの水平同期性異常検知のための RDF グラフ生成 “ , 電子情報通信学会 2024 年総合大会, 広島, 2024 年 3 月