分割管理されたグラフデータの境界データ統合技術

船矢　祐介† 　田中　剛† 　竹内　理† 　末永　晋也† 　多田　泰之††

† 日立製作所 研究開発グループ　〒 185–8601 東京都国分寺市東恋ヶ窪一丁目 280 番地
†† 日立製作所 社会ビジネスユニット　〒 319–1293 茨城県日立市大みか町五丁目 2 番 1 号
E-mail: †{yusuke.funaya.rd,tsuyoshi.tanaka.vz,tadashi.takeuchi.dt,shinya.suenaga.mt}@hitachi.com
††yasuyuki.tada.ae@hitachi.com

あらまし　本来一つのグラフデータが，分割された状態で作成・管理されることがある．電力網の設備データが一例であり，高度な分析を行うにはデータを統合する必要がある．しかし，それぞれの地域又は業務システムの特性に最適化して作成されたデータは，設備名称の命名規則が異なり，結果的に機械的に統合することが困難であった．そこで我々は，データに含まれる普遍的な値（物理量）に着目し，シンプルな操作で統合ルールを推測するシステムのコンセプトと，ルールを基にグラフデータを統合する手法を開発した．本報告では統合手法をプロトタイプにより評価し，推測したルールに基づき 2 つのグラフデータが機械的に統合できることを確認，コンセプト実現の見込みを得た．

キーワード　グラフデータ，設備データ，データ統合，ドメイン知識活用

1　はじめに

グラフ，及びグラフデータベースでは，RDB よりもデータの関係性を明確に表現することができ，クエリを実行することで，大量のデータからパターン，パス，コミュニティ，インフルエンサー，単一障害点，その他の関係を見つけることが高速かつ容易に可能となる．また，各ノードに自由に属性値を持たせることができるため，より柔軟なデータ表現が可能である [1]．そのため，カスタマどうしの関係性，データセンタネットワークなどの要素間の関係性，エンタテインメントにおける発信者と受信者の関係性など，様々な分野で，データ間のつながりをグラフで理解し解析することが，ビジネスにおいて今後も重要となってくる [2]．

しかし，本来一つのグラフデータであるべきものが，業務システムの構成や歴史的経緯などから，分割された状態で作成・管理されることがある．例えば，設備データであれば拠点ごとに，知識データであれば知識を獲得する場面（入力した人）ごとに，購買履歴データであれば店ごとに，それぞれデータが分割されて作成・管理される可能性が高い．ここで，先に述べたように関連する事象を検索したり，機械学習によって高度な分析を行うためには，これらバラバラに管理されているデータを統合して本来の形である一つの大きなグラフデータにする必要がある（図 1）．これらは現実世界では一連のデータなので，理論的には機械的に統合可能なはずである．しかし，データを作成・管理する業務システムが異なるために，名称や ID 等の命名規則が異なる，表記方法が異なる，粒度（ノードの省略の仕方など）が異なるなど，実際には機械的な統合を妨げる要因がいくつか存在する．

ここで，このようなグラフデータの一例として，電力ネットワークの設備データについて述べる．電力ネットワークは，発電所，変電所，そして需要家までを送電線で結んだネットワークになっており，それらの設備データは本来ひとつの大きなグラフデータとして表現可能なものである．しかし，計算機が現代のように高性能ではない時代から，各設備をきちんと管理し，安定した電力供給を実現するために，地域ごとの給電制御所においてデータを作成・管理し，それぞれの業務システムの特性に最適化していった [3] [4]．その結果，地域によって ID の命名規則が異なる，表記方法が異なるなど，結果的にデータを一つの大きなグラフデータに統合することが困難な状態となっており，将来の電力ネットワークに対する高度な分析・予測アプリケーションの発展の障壁の一つとなり得る．

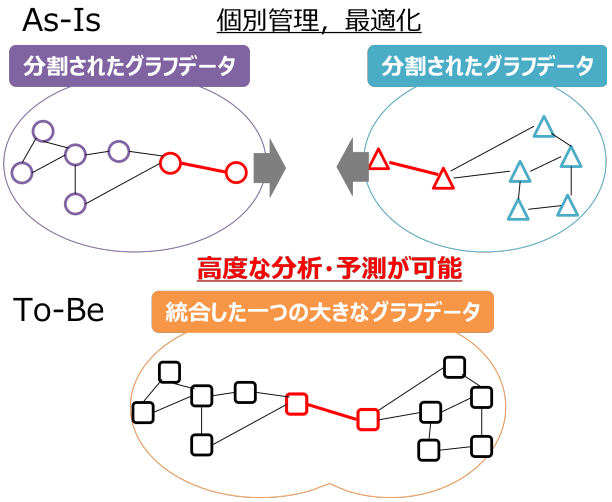

図 1　グラフデータの統合イメージ

もちろん，実際の設備を知っており，なおかつデータの作成・管理に直接携わっている人が手作業で統合していくことは不可能ではない．電力ネットワークの例では，各給電制御所に勤務する人などである．しかし，膨大な設備データを手作業で統合していくと，次のような課題が生じる．

- 統合にコスト（時間と人件費）がかかり本来の業務に支障が出る可能性がある．しかも，アプリケーション開発者の要望，または統合前の各データに追加・変更が生じるなどの理由で再統合が必要な場合，毎回統合作業が発生する．
- ヒューマンエラーにより統合結果にミスが生じる可能性がある．さらに，実際の設備に直接触れないアプリケーション開発者は，そのミスに気づくことができない．
- データ作成時の規則性（ドメイン知識と呼ぶこともできる）を知っている人が退職したら統合できなくなる可能性がある．

本研究の目的は，高度な分析・予測アプリケーション開発のために，バラバラに作成・管理されているグラフデータを統合し，新しい ID を振りなおすことで，実際の対象物に即した正しい一つの大きなグラフデータを生成する，データ統合手法を開発することである．そこで本研究では，データに含まれる普遍的な値（物理量）に着目し，シンプルな操作で統合ルールを推測するシステムのコンセプトと，ルールを基にグラフデータを統合する手法を提案する．さらに，統合手法をプロトタイプにより評価し，推測したルールに基づき 2 つのグラフデータが機械的に統合できることを確認，コンセプト実現の可能性を提示する．

以降，第 2 章ではデータ統合の課題を例を交えて述べる．第 3 章ではドメイン知識を用いたグラフデータ統合技術について述べる．第 4 章ではプロトタイプによる評価結果を述べる．

2　グラフデータ統合における課題

本章では，グラフデータ統合における課題を述べる．

第 1 章で述べた通り，本来一つのグラフデータであるべきものが，業務システムの構成や歴史的経緯などから，分割された状態で作成・管理されることがある．これらは現実世界では一連のデータなので，理論的には機械的に統合可能なはずである．しかし，データを作成・管理する業務システムが異なるために，名称や ID 等の命名規則が異なる，表記方法が異なる，粒度（ノードの省略の仕方など）が異なるなど，実際には機械的な統合を妨げる要因がいくつか存在する．

そこで本研究では，電力ネットワークの設備データを用いて，統合を妨げる要因を抽出した．電力ネットワークの設備データは基本的に各地域の給電制御所で作成・管理する．一方，実際の物理的な送電線は管理エリアをまたいでつながっており，つまり，各地域の送電線などの設備データは，給電制御所の管理エリアの境界で接続され，一つの大きな電力網として統合可能なはずである．よって，電力ネットワークの設備データは典型的な“ 分割されたグラフデータ ”の一つと言え，ここで示す統合の課題は他のグラフデータにも当てはまると考える．

まず，分割されたグラフデータの構造を図 2 を用いて定義する．グラフデータは，ノードと，ノード間を接続するエッジで構成される．ノードとエッジには，それぞれ属性値が含まれている．電力ネットワークの設備データでは，ノードは発変電所や送電線の区切りの設備（鉄塔などに相当する）であり，エッジは送電線である．属性値はそれぞれの設備のデータや管理用のタグである．

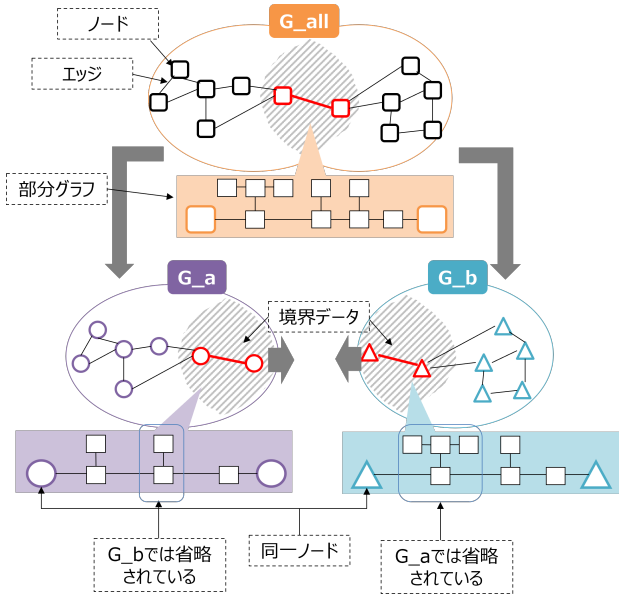

図 2　分割されたグラフデータの定義

ここで，一つのグラフデータ G_{all} が，領域 A のグラフデータ G_a と，領域 B のグラフデータ G_b に分割されて作成・管理されているとする．電力ネットワークの設備データでは，隣接する地域を管理する給電制御所 A と給電制御所 B のデータに相当する．G_a と G_b の境界部分では，本来繋がっているはずのノードとエッジが，本来のデータ構造とは無関係に分断された形になる．一方で，G_a 及び G_b は，領域 A 及び領域 B のそれぞれで既存の業務を遂行するために必要な計算や検索が可能な形になっている必要がある．さらに，G_{all} が持つすべてのノードとエッジは，少なくとも一か所以上の領域で管理されている必要がある．そのため，G_a 及び G_b の境界部分では，いくつかのノードとエッジが両方に含まれていることになる．ただし，重複しているノードとエッジは，それぞれの領域での名称や ID 等の命名規則，表記方法，粒度（ノードの省略の仕方など）が適用される．言い換えると，それぞれの領域の境界部分には，本来は同一のノードまたはエッジが，それぞれ異なる名前や表現方法で重複して登録されていることになる．以降，これらの重複したデータのことを境界データと呼ぶ．

また，グラフデータの一部を切り出したものを，本論文では部分グラフと呼ぶ．部分グラフは，あるノードから到達可能な別のノードまでの，両端ノードとその間のノード及びエッジの集合である．この考え方は，第 3 章のデータ統合の節で用いる．

これを踏まえて，G_a と G_b を統合する際の課題を考える．2 つのグラフデータを統合するには，G_a と G_b の境界データを特定し，それらを統合して一つのノードまたはエッジにすればよい．そこで問題となるのが，重複しているノードとエッジは，それぞれの領域での名称や ID 等の命名規則，表記方法，粒度（ノードの省略の仕方など）が適用されている，ということであ

る．具体的に言うと，図 2 に示すように，境界データには，同一ノードまたはエッジと，領域 B では省略されている（G_a にしかデータがない）ノードまたはエッジと，領域 A では省略されている（G_b にしかデータがない）ノードまたはエッジが含まれている．そのため，G_a と G_b を統合するには，名称や ID 等の命名規則，表記方法，粒度（ノードの省略の仕方など）が異なるデータ群の中から，どのように同一ノードとエッジを特定し統合するか，及びどうのように片方にしかないノードとエッジを適切な位置にマージするか，が課題となる．さらに，統合後には，ID に不整合が生じないように，ID 体系を再整理し，G_{all} としての ID を振りなおす必要がある．

グラフデータを統合する手法として，音声認識の精度向上を目的に，グラフデータで表現される音声データを統合する手法 [5]，営業の業務効率化を目的に，グラフデータで格納されている社員 DB と顧客 DB のデータを統合する手法 [6] が開示されている．これらは統合のルールを予め人間が複数個用意しておき，データからどのルールに当てはまるかを機械的に推測し，グラフデータどうしを統合している．しかし，先ほどから例に出している電力ネットワークの設備データなどでは，同一ノードまたはエッジを特定するためのルールが複雑かつ明示されていない．ここで，明示されていないというのは，給電制御所ごとにルールが異なり，かつ給電制御所どうしをまたいだルールは実際の設備をよく知る人にしかわからない（データからは推測できない）ということである．そのため，既存手法では，予め人間がルールを用意して置くことは困難であり，上記の手法ではグラフデータをマージすることができない．

また，電力ネットワークの設備データを統合する研究として，オブジェクト指向データベース（OODB: Object-Oriented Database）を用いて統一的に管理する試みがある [3] [4]．彼らは，同じ設備を対象としながらそれぞれ個別のデータを持つ複数のシステムが存在し，それらのデータの整合性を保ちつつ管理するのは困難であるという課題に対して，OODB を用いて，管理性，検索速度，データの取り出しやすさの両立を実現している．ただし，これは各システムが持つ個別のデータについて，既存のデータをマイグレーションすることについては触れられておらず，既に統合されているか，又は新規にデータを作成しなおすことが必要となる．また，OODB ではグラフ DB よりもデータ表現方法の拡張性が乏しいため，将来の高度な分析を活かした系統制御や，そのための新しい考え方に基づいた電力設備に対応するためには，グラフデータのまま統合・管理できる手法が必要である．よって，既存手法では，どのように同一ノードとエッジを特定し統合するか，及びどうのように片方にしかないノードとエッジを適切な位置にマージするか，の課題を解決することはできない．

ここで，データ品質を改善し，データ分析の信頼性を高めるための最も重要なタスクの一つと言われているのが，実世界の同じエンティティを参照する（データベース上の）複数の記述を識別する Entity Resolution（ER）である [7] [8]．本論文での課題を抽象化して考えると，本研究は，電力系統のグラフデータに対する Entity Resolution の取り組みの一つであると位置づけることができるだろう．

これらの関連研究も踏まえ，第 3 章では，設備をよく知る人が持つルール推測の知識を用いて，同一ノードまたはエッジを特定・マージする手法を提案する．

3 ドメイン知識を用いたグラフデータ統合技術

第 3 章では，ドメイン知識（設備をよく知る人が持つルール推測の知識）を用いて，ルールを機械的に推測するシステムのコンセプトと，推測したルールを用いて同一ノードまたはエッジを特定・マージするデータ統合手法を提案する．

ルールを機械的に推測するために，本研究では，関係者との議論を基に，まず，ID または名前から同一ノードまたはエッジを特定・マージするための情報を得た．以下にその一部を示す．

- 設備 ID の下 n 桁は領域 A と領域 B で共通である．
- 名称が一致する場合は同一設備であるが，同一設備の名称が一致するとは限らない．
- 名称は入力者によって全角・半角が統一されていないことがある．

これらにより，主要な設備（発変電所や高圧送電線）について，同一かどうかを特定することが可能である．しかし，これではまだ情報が足りず，主要設備間に存在するいくつかの種類の設備が特定できないことがわかった．そこで，本論文では，ID と名前に加えて，データに含まれる普遍的な値（物理量）に注目した．一例を以下に示す．

- 送電線の物理量 A は長さに比例する．つまり物理量 A が同じ区間は，名称や間のノード数が異なっていたとしても同一区間である可能性が高い．

これらの情報を抽象化すると，「データ内のどこの属性値をどうやって比較すれば，あるデータどうしに関連があることがわかるか？」ということである．それを機械的に推測するために，本研究では，同一のノードまたはエッジの一例をシステムに人間が教えることで，データからルールを推測し，グラフデータを統合するシステムを考案した．

3.1 グラフデータ統合処理の手順

提案するグラフデータ統合システムの処理の流れを図 3 に示す．処理の流れは以下の 6 ステップである．

STEP1 では，統合するデータ G_a と G_b を読み込み，データ処理のための基本的な前処理を行う．STEP2 では，G_a と G_b のそれぞれから，統合対象となるデータを抽出するためのルールを推測する．本システムで統合の対象となるデータは，2 章で定義した境界データのことを指す．またデータ抽出ルールを推測するために，ドメイン知識を持つ人に，GUI を通して簡単な入力を与えてもらう．GUI によるドメイン知識入力とルール推定方法は次節以降で述べる．STEP3 では，STEP2 で得られたデータ抽出ルールに従い，G_a 及び G_b から統合対象データを抽出する．STEP4 では，次に，統合対象データを統合するために必要なルールを推測する．データを統合するためには，G_a と

図 3　グラフデータ統合システムの処理概要

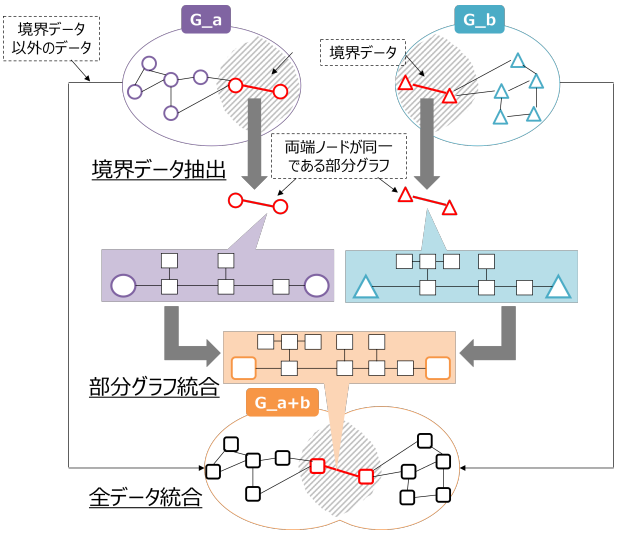

図 4　データ統合方法

G_b それぞれの境界データの中から，同一のノードとエッジを見つける必要があるため，ここでは同一ノード判定，及び同一エッジ判定ルールを推定する．ここでも STEP2 と同様に，ドメイン知識を持つ人に入力を与えてもらう．データ統合方法の詳細は次節以降で述べる．STEP5 では，STEP4 で得られたデータ統合ルールに従い，同一ノードと同一エッジを見つけて，そこから統合対象データの統合を実施する．STEP6 では，STEP5 で統合済みのデータと，STEP3 で抽出しなかったデータ（境界データではないデータで，統合処理の必要がない部分のデータ）を結合し，一つのグラフデータ G_{all} を作成する．

以上のステップにより，G_a と G_b を統合し G_{all} を作成する．

3.2　データ統合のルール推定の課題

本節では，データ統合方法について述べる．データ統合方法の説明を図 4 に示す．データ統合は，統合対象データを抽出し，その部分グラフを一つのデータに統合し，最後に全データを統合する．

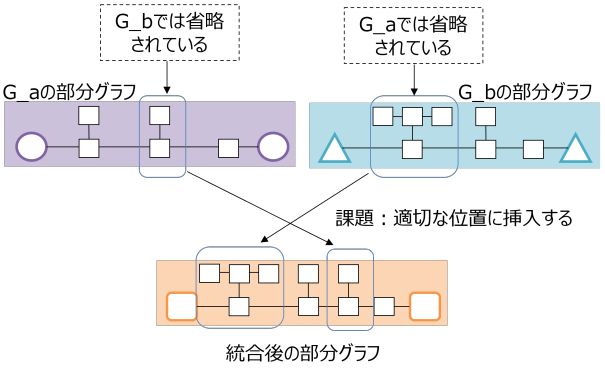

図 5　部分グラフ統合の課題

まず，統合対象データの抽出では，G_a と G_b のデータから，境界データ（重複して登録されているノードとエッジ）を見つける．これは，図 3 では STEP2「統合対象データ抽出のためのルール推定」にあたる．さらに境界データから部分グラフを抽出する．部分グラフとは，第 2 章で定義した通り，あるノードから到達可能な別のノードまでの，両端ノードとその間のノード及びエッジの集合であり，電力ネットワークの設備データの例では，ある変電所と別の変電所，及びそれを結ぶ送電線に相当する．部分グラフを抽出する時に必要なルールは，ID 名等の規則が異なっている G_a と G_b から，同一のノードとエッジを判定するためのルールである．

次に，抽出した部分グラフを一つの部分グラフに統合する．これは，図 3 では STEP4「部分グラフ統合のためのルール推定」にあたる．部分グラフは，両端ノードがそれぞれ同一であるが，両端ノードの間のノードやエッジは，G_a と G_b で，ID 名等の規則だけでなく，表記方法も異なっていることが想定される．つまり，G_a 側の部分グラフには存在するが，G_b 側の部分グラフでは省略されているなどの理由で存在しないノードとエッジがある可能性があるということである．部分グラフ統合の課題を図 5 に示す．電力ネットワークの設備データの例では，変電所間の送電線区間の区切り（鉄塔など）や，一部の送電先（需要家など）が各給電制御所のルールに従って省略されている場合に相当する．このような部分グラフを統合するには，G_a 及び G_b の部分グラフ内のノードが，統合後にどのような位置関係になるべきかを見つけ，G_a にしかないデータ，G_b にしかないデータ，両方にあるデータのそれぞれを適切な場所に挿入する必要がある．そのために必要なルールは，部分グラフの表記方法に関わらず，各ノードの挿入位置を判定するための基準となるノードを見つけるルールである．

以上をまとめると，必要なルールは，命名規則や表記方法が異なるデータの中から，同一のノードを見つけるためのルール，及び同一のノードどうしの位置関係を見つけるためのルールということになる．しかし，命名規則や表記方法が定式化されていない（もしくは自然言語によるもので定式化が不可能な）場合，例えば電力ネットワークの設備データなど，では，ルールを機械的に推測することは困難である．特に，STEP4 の部分グラフ統合のためのルール推定では，発変電所とそれらを結ぶ主

要な送電線を対象としている STEP2 とは違い，それらの間に存在するローカルな設備も対象とするため，関係者へのヒアリングで得た情報のみではルールの推定は非常に困難である．

3.3 ルールの推定方法

そこで本論文では，図 3 STEP4 のルール推定のために，命名規則や表記方法に依らず一定の規則に従う物理量に着目する．物理量は，人間が決める ID や名称とは違い，誰が測定しても同じ値となるはずである．そのため，物理量を用いることで，命名規則や表記方法が異なる設備でも，同一かどうか判定したり，ノードの位置関係を決めたりすることができるはずである．

その理由を図 5 を用いて説明する．今，G_a の部分グラフと G_b の部分グラフが図のような構成になっており，両端のノードは同一であると分かっているが，その間のノードとエッジは，G_a と G_b で命名規則や表記方法が異なるため対応関係が分からないと仮定する．ここで，送電線の物理量を用いて同一ノードまたはエッジを特定することを考える．送電線の物理量とは，例えば長さ（距離），材質，及びインピーダンス等の電気特性である．この中で，もし「長さ」が G_a と G_b 両データに含まれていたとすると，同一と分かっている端のノードから同一の長さ（距離）にあるノードは，ID や名称が異なっていたとしても同一であると判定することが可能である．また，インピーダンスは送電線の長さに比例するという物理法則を用いれば，インピーダンスの和を用いて,「長さ」と同様に判定することも可能である．

このように，各ノードおよびエッジの属性値の中にある物理量を用いてデータを加工することで，G_a と G_b それぞれのノードどうしの関係が分かり，先に述べたルールを作成できる．つまり，G_a の属性値と G_b の属性値のそれぞれに対して，何かしらの加工を行って 2 つの値が一致すれば，その属性値とその加工方法がルールであると推測でき，そのための加工方法を予め用意しておくことで，機械的にルールを推測することが可能である．

そこで，これをシステム化するために，本論文では，属性値リストと加工・判定メニューを用いた推測方法を考える．システム化では，図 3 STEP2 及び STEP4 の両方に共通して利用できる仕組とするため，物理量以外の属性値（例えば名称など）も扱うことを前提とする．属性値リストと加工・判定メニューを図 6 と図 7 に示す．属性値リストには，各グラフデータのノードまたはエッジが持つ属性値が記録されている．加工メニューには，属性値の加工方法が記録されている．加工方法とは，例えば，大文字・小文字や全角・半角の違い（文字種）を統一するなどの処理である．判定メニューには，属性値どうしが等しいか等しくないかを判定する方法が記録されている．判定メニューとは，例えば，文字列が等しいかどうかなどの処理である．これらを用いて，A の属性値 1 と B の属性値 1 を，加工方法 1 で加工し，判定方法 1 で判定し，判定結果が真（一致）であれば，その属性値どうしは関係性があり，加工方法と判定方法がそれらを関係づけるルールであると推測する．これをそれぞれ 1 から n まで総当たりで繰り返し計算することで機械的に

Aの属性値リスト

#	Key	Tyep	Value
1	属性値1	文字列	SA01
2	属性値2	文字列	SA02
3	属性値3	文字列	ABC
4	属性値4	数値	100
	…	…	…

Bの属性値リスト

#	Key	Tyep	Value
1	属性値1	文字列	SB02
2	属性値2	文字列	SB03
3	属性値3	文字列	Def
4	属性値4	数値	100
	…	…	…

図 6　属性値リスト

加工メニュー

#	加工名	処理内容
1	何もしない	Null()
2	文字種統一	Change_chars()
3	絶対値	Abs_nums()
4	数値の和	Sum_nums()
	…	…

判定メニュー

#	判定名	処理内容
1	文字列が等しい	Is_equal_chars()
2	数値が等しい	Is_equal_nums()
3	文字列が長い	Is_bigger_than_chars()
4	数値が大きい	Is_bigger_than_nums()
	…	…

図 7　加工・判定メニュー

Aの変電所データ

ID	属性値1 名称	属性値2 接続先	…
A100	**HITACHI-1変電所**	A101	…
A200	**HITACHI-2変電所**	A201	…

Bの変電所データ

ID	属性値1 名称	属性値2 接続先	…
0101	**Hitachi-1変電所**	0111	…
0102	**Hitachi-2変電所**	0112	…

図 8　メニューを用いた判定の例 1

ルールを推測する．

ここで，属性値リストと加工リストを使った判定の例を図 8, 図 9 に示す．一つ目の例は，変電所のデータから同一変電所を特定する場合である．図 8 では，データ A とデータ B で変電所の ID や名称がバラバラである．ここで，属性値リストに加工・判定メニューを用いてルールを推定すると，属性値リスト「属性値 1（名称）」の，加工メニュー「文字種統一（大文字と小文字を統一）」のあと，判定メニュー「文字列が等しい」を適用することで，HITACHI-1 変電所と Hitachi-1 変電所が同一変電所である，というルールを生成することができる．二つ目の例は，送電線のデータから同一区間を特定する場合である．図 9 では，ID や接続先の名前が似ているが，末尾の数字は A, B 独自に振られた番号であり数字が同じものが同じ設備とは限らないが，HITACHI-1 は同一の変電所であることが分かっているとする．この場合に考えられるルールは，インピーダンスは送電線の長さに比例するという物理法則を用いるものである．この例でも，属性値リストに加工・判定メニューを用いてルールを推定すると，属性値リスト「属性値 3（インピーダンス）」の，加工メニュー「数値の和（区間 1+ 区間 2+…）」のあと，判定メニュー「数値が等しい」を適用することで，A の区間 1（つまり HITACHI-1 変電所から電柱 1 まで）と，B の区間 1 + 2（つまり HITACHI-1 変電所から電柱 2 まで）が同一区間である，というルールを生成することができる．

一方で，これを人間から一切の情報受けずに，考え得るすべてのノードの組合せに対して，すべての属性値とすべての加工

Aの送電線データ

ID	属性値1 接続元	属性値2 接続先	属性値3 インピーダンス	…
区間1	HITACHI-1	電柱1	**100**	…
区間2	電柱1	電柱2	60	…

Bの送電線データ

ID	属性値1 接続元	属性値2 接続先	属性値3 インピーダンス	…
区間1	HITACHI-1	電柱1	**40**	…
区間2	電柱1	電柱2	**60**	…

図 9　メニューを用いた判定の例 2

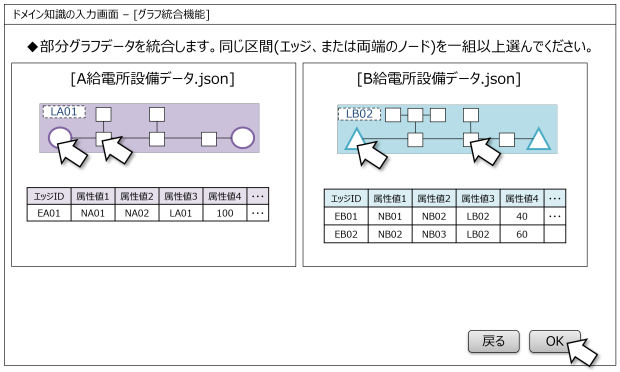

図 10　ドメイン知識獲得 GUI(1)

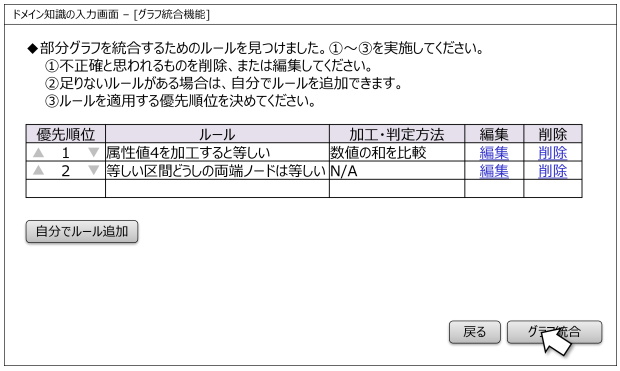

図 11　ドメイン知識獲得 GUI(2)

方法を用いて比較するのは，膨大な計算量になるだけでなく，たまたま値が一致するだけで間違ったルールを生み出す可能性も十分ある．それを解決するために，本論文では，同一のノードまたはエッジの一例をシステムに人間が教えることで，現実的な計算量で精度の高いルールを機械的に推測する方法を考案した．次節で述べる．

3.4　人が持つ知識の獲得方法

本節では，同一のノードまたはエッジの一例をシステムに人間が教える方法について述べる．人間がシステムに知識を入力するための GUI のコンセプトを図 10 から図 12 に示す．システムでは，統合対象データの抽出と部分グラフ統合の 2 回，ドメイン知識を入力する場面があるが，GUI の目的およびコンセプトはほぼ同じであるため，ここでは部分グラフ統合の GUI を例に示す．

まず，図 10 に示すように画面上に読み込んだグラフデータ

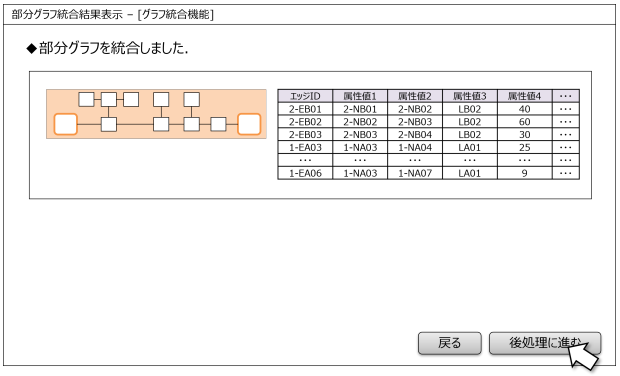

図 12　ドメイン知識獲得 GUI(3)

を表示し，知識のある人間が，そこから同一の区間（エッジまたは両端ノード）を，マウスなどを用いて指定する．次に，システムは入力された一組以上の同一ノードまたはエッジが持つ属性値に対して，前節で述べた加工・判定メニューを用いて一致するものを探し，ルールを推測する．最後に，システムは，すべての統合対象データに対してそのルールを適用することで，データを統合する（図 12）．ただし，ルールの候補が複数見つかる場合や，偶然指定された区間のみで一致する値がある場合を考慮し，一度人間がルールの候補確認する画面（図 11）を表示し，確認と修正を行えるようにする．

次章では，データ統合手法をプロトタイプによって評価する．

4　データ統合の評価

本章では，まず，提案するルールの推測方法により，ルールの候補を発見できるかどうか机上評価する．次に，開発したデータ統合機能のプロトタイプを用いて，2 つのグラフデータが，与えられたルールにより 1 つのグラフデータに統合できるかどうかを評価する．

4.1　ルール推測手法の机上評価

属性値リストに加工・判定メニューを用いてルールを推定できるかどうか机上評価する．評価に用いる部分グラフデータ，属性値リスト，及び加工・判定メニューを図 13 に，ルールの推定アルゴリズムの疑似コードを図 14 に示す．ここでは，ユーザが GUI により同一区間の例を一つ入力したと仮定し，選択された区間を使ってルールを推定する．ただし，今回のデータで発見されるべきルールは，この後の評価で用いるルールである「物理量 X の和が等しい区間は，同一区間である」というルールであることが分かっているため，机上評価では，そのルールが発見できるかどうかを評価する．

評価の結果を図 15 に示す．図は，疑似コードを実行した場合の，内部変数「加工済み A」「加工済み B」及び「判定結果」の値をループごとに出力したものである．机上評価の結果，24 ループ目で判定結果が真となり「属性値 4（物理量 X が格納されている）の和が等しい区間は，同一区間である」というルールをユーザに提案することができる見込みを得た．

ユーザが選択したAの区間

エッジID	属性値1	属性値2	属性値3	属性値4
EA01	NA01	NA02	LA01	100

ユーザが選択したBの区間

エッジID	属性値1	属性値2	属性値3	属性値4
EB01	NB01	NB02	LB02	40
EB02	NB02	NB03	LB02	60

Aの属性値リスト

#	Key	Tyep	Value
1	属性値1	文字列	NA01
2	属性値2	文字列	NA02
3	属性値3	文字列	LA01
4	属性値4	数値	100

Bの属性値リスト

#	Key	Tyep	Value
1	属性値1	文字列	NB01
			NB02
2	属性値2	文字列	NB02
			NB03
3	属性値3	文字列	LB02
			LB02
4	属性値4	数値	40
			60

加工メニュー

#	加工名	処理内容
1	何もしない	Null()
2	文字種統一	Change_chars()
4	数値の和	Sum_nums()

判定メニュー

#	判定名	処理内容
1	文字列が等しい	Is_equal_chars()
2	数値が等しい	Is_equal_nums()

図 13　机上評価に用いるデータ

```
1  def Func加工(x) = {加工メニューに従ってxを加工する}
2  def Func判定(x,y) = {判定メニューに従ってxとyを判定する}
3
4  for i_attr = 1 to num_of “属性値リスト”
5
6    for i_deform = 1 to numof “加工メニュー”
7      for i_judge = 1 to numof “判定メニュー”
8
9        加工済みA = Func加工[i_deform](属性値リストA[i_attr]);
10       加工済みB = Func加工[i_deform](属性値リストB[i_attr]);
11       判定結果  = Func判定[i_judge](加工済みA, 加工済みB);
12
13       if 判定結果==1
14         end;
15       else if 判定結果==0
16         continue;
17
18     end for
19   end for
20 end for
```

図 14　疑似コード：ルール推定アルゴリズム

#	加工済みA	加工済みB	判定結果
1	NA01	NB01, NB02	0
2	NA01	NB01, NB02	0
3	NA01	NB01, NB02	0
4	NA01	NB01, NB02	0
5	N/A(not数値)	N/A(not数値)	0
6	N/A(not数値)	N/A(not数値)	0
7	NA02	NB02, NB03	0
8	NA02	NB02, NB03	0
9	NA02	NB02, NB03	0
10	NA02	NB02, NB03	0
11	N/A(not数値)	N/A(not数値)	0
12	N/A(not数値)	N/A(not数値)	0
13	LA01	LB02	0
14	LA01	LB02	0
15	LA01	LB02	0
16	LA01	LB02	0
17	N/A(not数値)	N/A(not数値)	0
18	N/A(not数値)	N/A(not数値)	0
19	100	40, 60	0
20	100	40, 60	0
21	100	40, 60	0
22	100	40, 60	0
23	100	100	0
24	100	100	1

図 15　机上評価の結果

4.2　データ統合手法の評価方法

プロトタイプの処理内容を図 16 に示す．プロトタイプは，前処理，統合対象抽出機能，および部分グラフ統合機能から構成

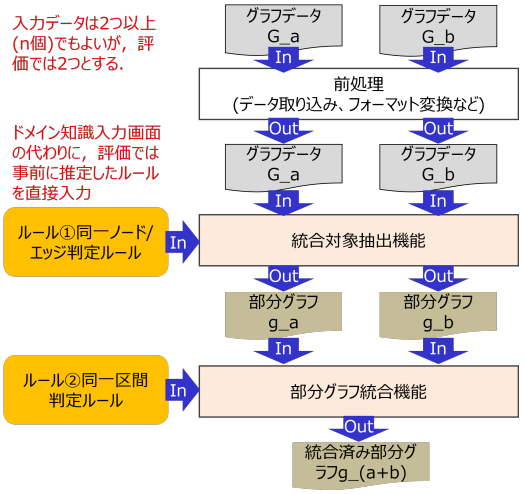

図 16　プロトタイプの処理内容

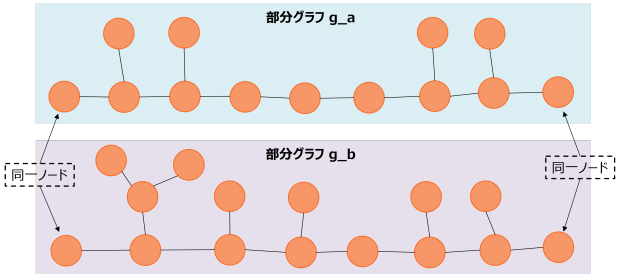

図 17　抽出された部分グラフ

されている．グラフデータ G_a 及び G_b の 2 つを入力とし，それぞれの部分グラフ g_a 及び g_b を統合した，統合済み部分グラフ $g_{(a+b)}$ を出力する．本評価では，ドメイン知識入力画面を用いたルール推定が完了したと仮定し，統合対象抽出のための同一ノード・エッジ判定ルールと，部分グラフ統合のための同一区間判定ルールを，直接プログラムに入力した．

入力データは，実際の電力ネットワークの設備データの一部を用いた．G_a と G_b はそれぞれ異なる給電制御所で作成・管理されていたデータであり，命名規則や表記方法が異なっている．ただし，本評価では，グラフ構造が正しく統合できるかどうかを評価できればよいので，以降の示すデータには実際の設備名称や属性値は記載していない．

4.3　評価結果及び考察

まず，統合対象抽出機能の出力結果である部分グラフ g_a および g_b を図 17 に示す．それぞれの部分グラフの両端ノードは，統合対象抽出機能によって同一と判定されたノードであり，両端ノードの間は同じ区間なので，実際のノード（設備）やエッジ（送電線）は同じ構成である．しかし，図 17 より，g_a と g_b はノード数やノード間のつながりなどに異なる点があることが分かる．

次に，プロトタイプの出力結果 $g_{(a+b)}$ を図 18 に示す．さらに，この区間の実際の設備構成，つまり正解データを図 19 に示す．図 19 の上図の正解データは公開されている系統図から筆者が書きだしたものである．上下にある太枠のノードは部分グラフ

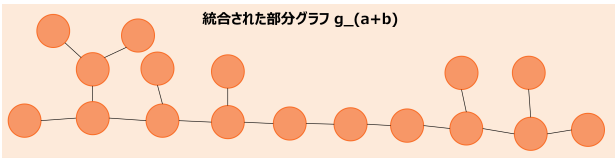

図 18　統合結果の部分グラフ

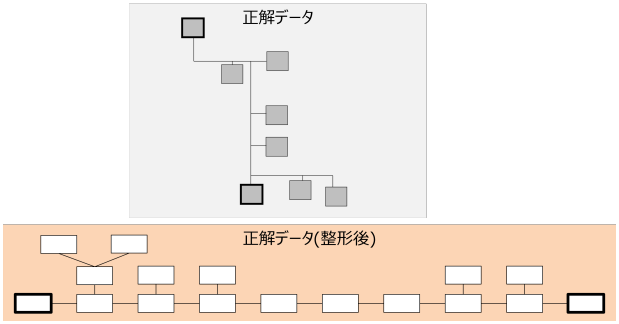

図 19　統合結果の正解データ

g_a または g_b の両端ノードにあたる設備である．図 19 の下図は，比較のしやすさのために，正解データの分岐点にノードを加え，トポロジーを維持したまま整形したものである．

プロトタイプの出力結果（図 18）と正解データ（図 19）を比較すると，部分グラフ内のノードとエッジの構成が同じであり，プロトタイプによって g_a と g_b が正しく統合できている．よって，同一ノード・エッジ判定ルールと，普遍的な値（物理量）に着目し同一区間判定ルールが与えられれば，データを正しく統合できることが分かった．これより，上記のルールを推測する機能および GUI のコンセプトが実現に至れば，分断され，異なる規則の下で管理されているグラフデータを，シンプルな入力操作で機械的に正しい一つのグラフデータに統合できる見込みを得た．

ここで，Entity Resolution の観点で本手法と本結果を見てみると，エンティティの類似度として，データベース上に存在する各データどうしの信頼度や関係性の近さのみでなく，本研究では，物理量に着目し，さらに値を加工した上で判定を行うことで，複雑な電気系統のデータで ER を実現できたと考えられる．さらに，ドメイン知識を入力する画面は，ER で言うブロッキング処理（総当たりによる膨大な判定数を削減する）をより効率的に実現できる形態を示したということができる．

5　結　　論

本来は一つのグラフデータが，分断されて異なる規則の下で作成・管理されている場合がある．高度な分析・予測アプリケーションを開発・発展させていくに，これらのグラフデータを統合して，本来の大きな一つのグラフデータにすることが課題である．しかし，命名規則や表記方法が異なるこれらのグラフデータを機械的に統合することは困難であった．そこで本論文では，データに含まれている普遍的な値（物理量など）に着目，シンプルな操作で統合ルールを推測するシステムのコンセプトと，ルールを基にグラフデータを統合する手法を提案した．統合手法をプロトタイプにより評価し，推測したルールに基づき 2 つのグラフデータが機械的に統合できることを確認，コンセプト実現の可能性を示した．今後の展望は，ルールを推測する機能および GUI のコンセプトの実装と検証である．

文　　献

[1] “What is a Graph Database?,” Oracle, https://www.oracle.com/autonomous-database/what-is-graph-database (2022.03.30 閲覧)
[2] Robinson, I. and Webber, J. and Eifrem, E., “Graph Databases: New Opportunities for Connected Data,” O’Reilly Media, 2015.
[3] 小笠原史久, 瀬川修, 若林光, 樋山良隆, 吉村吉彦, 石部直子, 原嶋秀次, “オブジェクト指向データベースを用いた電力設備統合データベースの構築,” 情報処理学会研究報告データベースシステム（DBS）, Vol. 1994, No. 62, pp. 29–36, 1994.
[4] 歌谷昌弘, “統合的電力系統解析支援システムの構築に関する研究，’’広島大学学位論文, 乙第 3214 号, 広島大学, 1999.
[5] 世木寛之, 都木徹, 田高礼子, 清山信正, “グラフ統合装置及びそのプログラム，’’公開特許公報, 特開 2010-32919, 日本放送協会, 2010.
[6] 有熊 威, “データ統合処理装置，システム，方法及びプログラム，’’再公表特許, WO2012/035754, 日本電気株式会社, 2014.
[7] Vassilis Christophides, Vasilis Efthymiou, Themis Palpanas, George Papadakis, and Kostas Stefanidis, “An Overview of End-to-End Entity Resolution for Big Data,” ACM Comput. Surv., Vol. 53, No. 6, Article 127, November 2021.
[8] Kopcke, Hanna, Andreas Thor, and Erhard Rahm, ”Evaluation of entity resolution approaches on real-world match problems,” Proceedings of the VLDB Endowment, 3.1-2, pp. 484-493, 2010.