オープンデータ連携に向けたIDマッピングソリューション

Identity Mapping Solution for Open Data Federation

● Vivian Lee　● 後藤正智　● 伊豆哲也

あらまし

オープンデータは，とりわけDATA.GOVやDATA.GOV.UKなどのオープンデータガバメント・イニシアティブの立上げによって，近年更に一般的なものとなってきた。オープンデータは誰でも自由に入手可能であり，あらゆるデータの使用と再公開が可能である。しかし，公開のための標準規格が存在しないことから，非常に多くの場合，オープンデータはそれぞれ独自のフォーマットを用いて作成されている。このため，ある公開サイトのデータセットは，ほかの公開サイトのデータセットと連携して利用することが難しい(サイロ化された)ものとなっている。この問題の解決のため，欧州富士通研究所(FLE)ではオープンデータを正確に連携する手段として，IDマッピングをベースとするソリューションの開発を行っている。

本稿では，セマンティック拡張型の高信頼性エンティティID調整機能，およびマッピングベースの連携フレームワークの2種類の基本技術を紹介する。FLEのソリューションによれば，組織におけるオープンデータ活用の難しさを緩和し，より良い意思決定により競争優位性を強化することが可能となる。

Abstract

The use of open data is growing, especially since the launch of open-data government initiatives such as DATA.GOV and DATA.GOV.UK. Open data are freely available to everyone to use and republish. However, the lack of publishing standards means that open data are usually in a proprietary format. Moreover, the data sets are always siloed. To address these problems, Fujitsu Laboratories of Europe (FLE) has been developing an identity mapping based solution for federating siloed open data regardless of their format. In this paper, we introduce the two fundamental technologies used by this solution: semantically enhanced reliable entity identity reconciliation and mapping based federation. Our solution facilitates open data utilisation in organisations, leading to better decision making and improved competitive advantage.

ま　え　が　き

インターネットの隆盛により，オープンデータは近年一般的なものとなってきた。これはとりわけ，DATA.GOVやDATA.GOV.UKなどのオープンデータガバメント・イニシアティブの立上げによるものである。この概念は，ある種のデータは著作権，特許，そのほかの制御メカニズムによる制約を受けることなく，誰でも自由に入手可能であり，所望に応じて使用と再公開が可能であるべきというものである。(1)

オープンデータの主なソースとなるのは，科学分野と政府関係のデータである。その中でも，オープンガバメントデータの動きへの関心が特に高く，世界中で大きな注目を集めている。Socrata社がまとめた「2014 Open Data Benchmark Report」によると，可能な限りすぐにデータを公開することを目標としている政府が大きな割合（41％）を占めている。(2) ガバメントデータを一般に開放することで公共サービスが向上し，スムーズに意思決定ができるという思いが，主な動機となっている。

しかし，現時点ではオープンデータを公開するための標準的な方法は存在しない。このことから，各情報エンティティ（組織や企業，人間や建物などを表すもの）には同一のデータソース内でのみユニークとなるIDが割り当てられており，かつ多くの場合，データソースはそれぞれ独自のフォーマットで公開されている。多様なオープンデータのソース群から同一エンティティに関する情報を発見し統合して，より総合的な見解を得ようとする際には，こうした状況が大きな障害となる。この問題に対して欧州富士通研究所（FLE）では，オープンデータの連携における基盤として，IDマッピング技術を採用したEIDERプラットフォームを開発した。

本稿では，まずEIDERプラットフォームの概要について述べる。次に，IDマッピングの新たな手法を紹介し，IDマッピングフレームワークがどのようにその調整結果を用いてより正確な連携を行うのかについて述べる。最後に，EIDERソリューションの応用例を示す。

オープンデータにおける問題と技術的課題

社会に関するより多くの知識を提供するという面においては，オープンデータには明らかな可能性があるものの，その使用方法には疑問符も付く。これらは，標準化されていない未加工のフォーマットで公開されているため，データを使用することはもちろんのこと，コンピュータシステムによる自動処理も非常に難しい。

オープンデータの受信から適切なデータ解析ツールの適用に至る一連の処理における主な技術的課題は，複数のデータセットを扱おうとすると，データセットごとに辞書やマスターデータが異なり，連携して利用することが難しい（サイロ化されている）という点にある。オープンデータの提供団体は，情報エンティティに対してそれぞれ独自のフォーマットを定義しており，それ以外の情報ソースの存在を意識することすらない。このフォーマットというのは，Excel，CSV，RDBMS（Relational Database Management System） に蓄えられたデータなどの構造化されたデータ形式には限られておらず，XMLのような半構造化形式や，ウィキペディアのテキスト項目などの構造化されていない形式のものもある。更には，オープンデータの一つの理想型として，Linked DataというRDF（Resource Description Framework）を使ったフォーマットがW3C（World Wide Web Consortium）より提唱されている。それに加えて，たとえ異なるソースからのデータ統合が可能であるとしても，同じ情報エンティティを参照していることを示すためのデータが問題となることがある。FLEが顧客と話した経験によれば，全勤務時間の70％近くがデータの重複や誤記，表記ゆれを探し，修正・削除を行うデータクレンジングなどの前処理に費やされている。その理由としては，それぞれのデータソースが，別のソースでは異なる識別子を有する情報エンティティにローカルな識別子を発行する場合があり，現時点では，このローカル識別子に対する標準的な定義方法が存在しないためである。

この問題の典型的な事例を**図-1**に示す。この図が示すように，IBMのデータを収集するには，データソースごとに固有の識別子を正しく指定する必

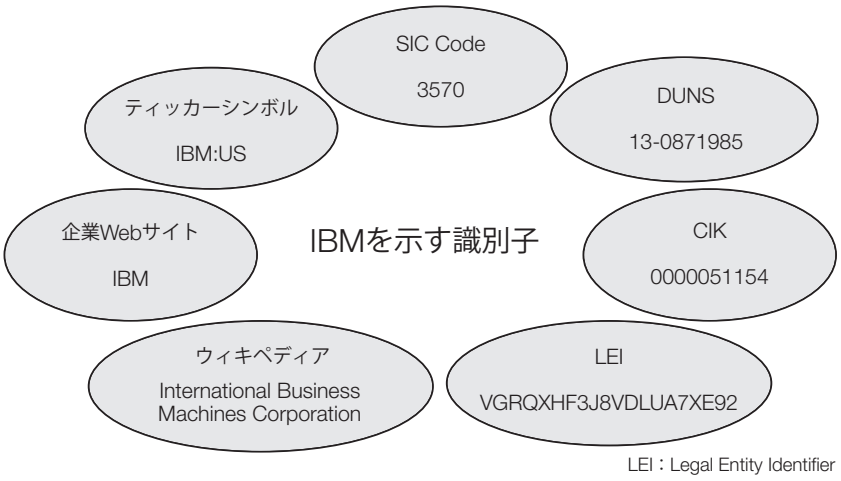

図-1　データソースごとの識別子の例

要がある。株価データを取得するには，ティッカーシンボルとしてIBM:USを，SEC（米国証券取引委員会）のサイトではCIK（Central Index Key）コード0000051154を使わないと，データの収集が非常に難しいものとなる。このようにオープンデータを扱う際には特に，データソースごとに異なる識別情報をリコンサイルするという処理が非常に重要となっている。

EIDERプラットフォームの概要

オープンデータには，その特性上，情報エンティティの統一が難しいことに加え，二つの懸念がある。一つはデータ自身が正しいかどうか判断が難しいこと，もう一つはデータが頻繁に更新されていくため，一度の統合処理では最新の状況を反映できないことである。

そこで，EIDERプラットフォームは，これらのオープンデータ内のエンティティ IDマッピング分野における問題に対処するために設計されている。

このプラットフォームは，業種に応じて変遷する情報エンティティをその特性に合わせて柔軟に識別し，同一判定できるようにロジックを選択する機能を提供する。また，公開先のデータの更新頻度に合わせて内部情報を更新する。更に，ユーザーの嗜好に合わせて検索を最適化するために，動的な情報マッピング機能を提供する。これらの機能によって，いかなるオープンデータ環境においても最適化されたデータの整理ができるようになる。

このプラットフォームの名称は「Entity Identification Reconciliation Platform」に由来している。オープンデータの自動連携を提供する目的で，EIDERプラットフォームでは以下の技術領域を対象とするソフトウェアパッケージが提供されている。

・データ抽出
・リコンシリエーションマネージャー/リコンシリエーター
・データ統合
・データ鮮度
・クエリ

アーキテクチャー

EIDERシステムのアーキテクチャーを**図-2**に示す。オープンデータは多様なオープンデータソースからクロール（情報取得）やエクストラクト（抽出）が行われる。その後，このデータは元のフォーマットのままキャッシュストレージに格納される。これをデルタデータソースと呼ぶ。リコンシリエーションマネージャーは，そこに登録されているリコンシリエーターを用いてリコンシリエーションタスクを起動し，この処理の結果はデータインテグレータで統合される。最終的な出力は，EIDERデータストレージに全て格納され，クライアントアプリケーションによるクエリが可能となる。EIDERシステム内部のデータが外部ソースと可能な限り同期が取れていることを保証する目的

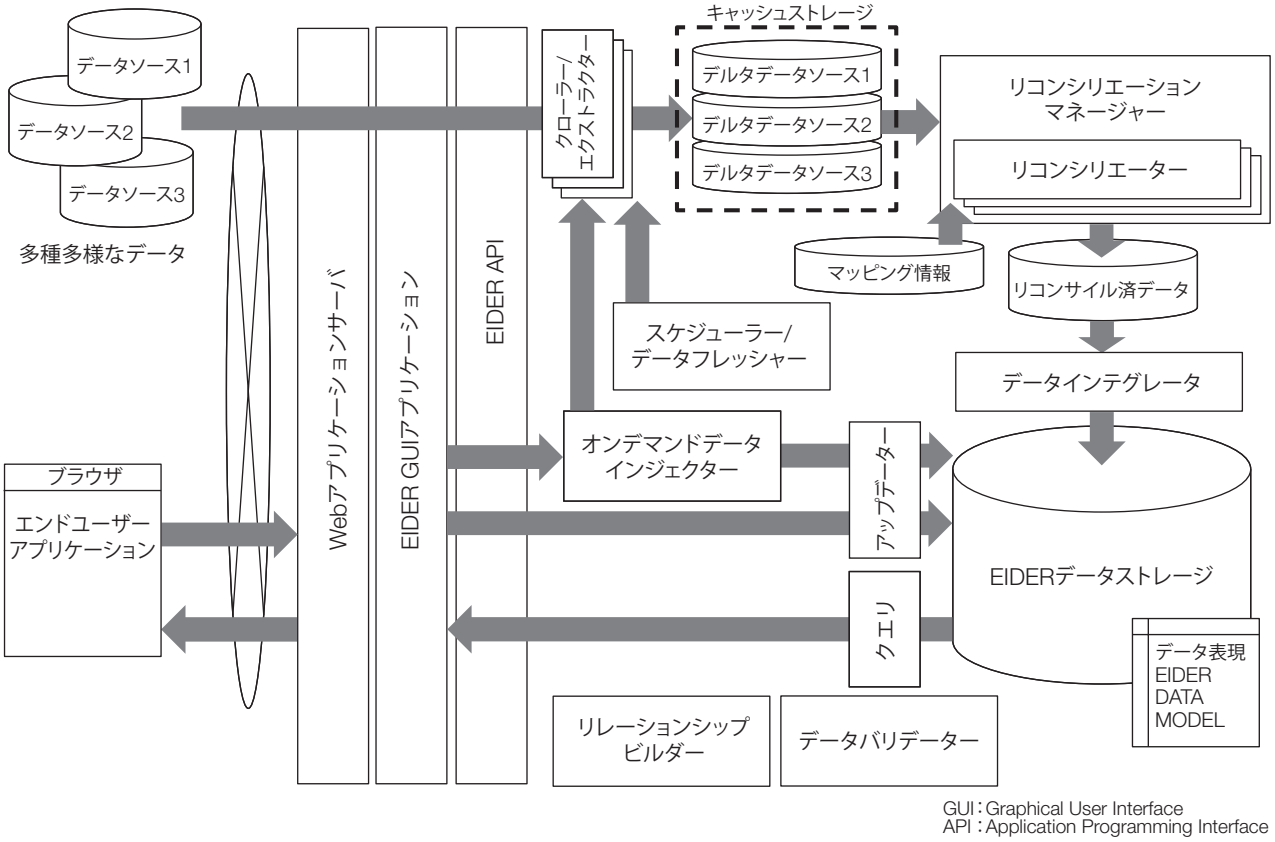

図-2　EIDERプラットフォームアーキテクチャー

で，データフレッシャーが定期的タスクとして更新ジョブを起動するか，あるいはオンデマンドで実行される。以下，IDマッピングとデータ統合に対するFLEの主要技術を中心に，リコンシリエーションマネージャーおよびデータフレッシャーの概念を簡単に説明する。

● **リコンシリエーションマネージャー**

リコンシリエーションの研究中に認識したのは，オープンデータの性質は多様であることから，情報エンティティの履歴が長い場合や複数言語で表現されている場合には特に，一つのリコンシリエーションロジックに頼っているだけでは必ずしも満足のいく結果は得られないという点である。

それぞれのロジックの強みを最大化してより望ましい結果を得るために，動作中にどのようなリコンシリエーションメカニズムでも適用可能なリコンシリエーションマネージャーの設計を行った。このリコンシリエーションマネージャーは，例えば，次節に述べているようにFLE社内のIDマッピング手法を使用してリコンシリエーションが実行可能である。そればかりではなく，OpenRefine[(3)]を通じて新たにリコンシリエーションリクエストを開始することも可能である。

システムの起動段階において，リコンシリエーションマネージャーはリコンシリエーターが格納されているディレクトリをスキャンして，各種リコンシリエーションメカニズムの登録を行う。その後，クライアントアプリケーションに利用可能なリコンシリエーターの情報が伝えられ，それによりユーザーは，より広範囲な結果の比較や導出のために，様々なリコンシリエーションメカニズムを選択することが可能となる。

● **データフレッシャー**

データ連携システムやデータ統合システムなどでも，システム自体が格納データの保有や生成を行っていないため，データ鮮度は常に重要な課題である。外部のデータソースがいつ変更されるかも知れないオープンデータ環境では，この要件は特に重要であるが，実現は困難である。

EIDERプラットフォームの重要な機能の一つ

に，データ鮮度の維持（リフレッシュ手法）がある。これは，キャッシュストレージ内のデルタデータソースは，外部オープンデータの変更点を可能な限り頻繁に複製する必要があることを意味する。リフレッシュ手法は，下記の2種類が提供されている。

・定期リフレッシュ
・オンデマンドリフレッシュ

定期リフレッシュタスクは定期的に起動され，オンデマンドリフレッシュはクライアントがバックエンドのサーバに対してクエリを発行したときにトリガがかかる。リフレッシュされるデータは，クエリに関係するものだけとなる。

IDマッピングの新たな手法

多様な形式のオープンデータソース群からエンティティ IDをリコンサイルする手法の一つは，エンティティ名が合致するかどうか比較対象として使用する方法である。しかし，エンティティ名が曖昧であると，リコンシリエーション結果の精度に重大な影響を与える。完全で正確なエンティティ名や法的に定められたエンティティ名であっても，複数の表記を持つことがあるためである。データに多言語が用いられている場合，この問題は更に複雑化する。

自然言語処理（NLP：Natural Language Processing）は十分に確立した技術であり，テキストベースの解析やプロセスの解決に適用可能である。しかし，NLPが提供する様々なタスクと技術は特定の問題の解決のみに焦点が絞られており，全て独立している。高効率でありながらも高精度なオープンデータのエンティティ IDリコンシリエーションを提供する包括的手法は存在しない。

NLP技術の弱点を克服し，より包括的で高精度なリコンシリエーション結果を得るために，FLEでは新たにリコンシリエーション手法を開発した。この手法は，並列に実行される三つのプロセッサから構成されている。このリコンシリエーション手法の単純化された構成を**図-3**に示す。

● データエクストラクターおよびストレージ

このデータエクストラクターとストレージは，図-2に示すEIDERシステムアーキテクチャーの主要部分におけるエクストラクター /クローラー，およびデルタデータソースに対応するものである。

データエクストラクターの役割は，オープンデータソースから適切な情報を取得することである。オープンデータの性質は変化に富んでいることから，データエクストラクターが提供する機能には，ファイルのダウンロード，Webクローリング，構造化/半構造化/非構造化データの構文解析，および抽出が含まれている。

オープンデータの大半からは，構造化/半構造化データのダウンロードが可能であるが，この場合ダウンロードされたデータは元のフォーマットの

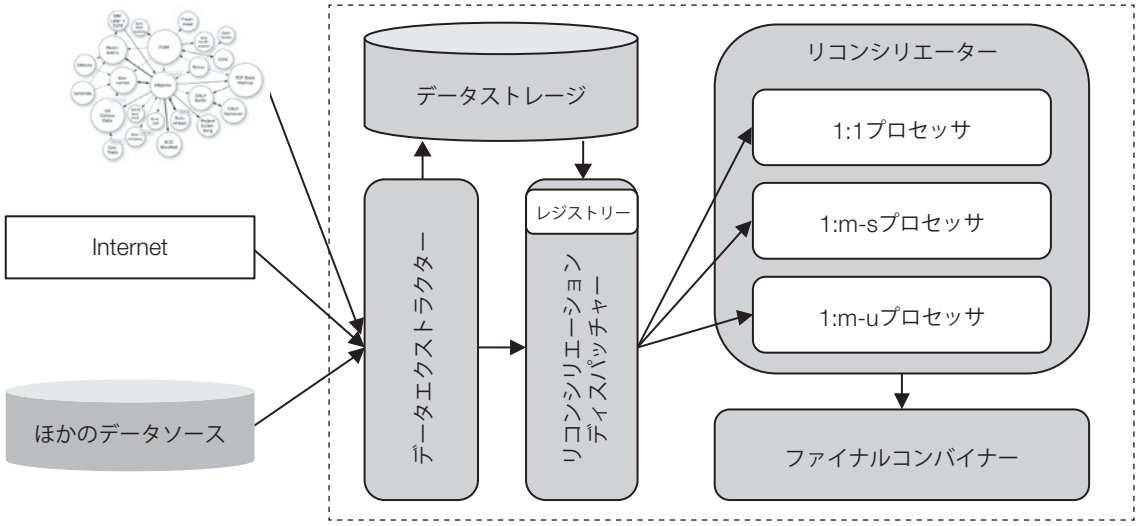

図-3　リコンシリエーション・コンポーネントアーキテクチャー

ままデータストレージに保存される。リコンシリエーションコンポーネントは実行時にデータの抽出が可能となっているが，少なくともデータの大部分をローカルネットワーク内に格納して，ネットワークアクセスのオーバーヘッドを回避することが性能面の理由からより適切である。

● **リコンシリエーションディスパッチャー**

データがデータストレージに格納されると，リコンシリエーションディスパッチャーは各ソースを検査して，それぞれのデータソースに対するキーと名前の間のマッピング値を生成する。この処理のフローは以下のとおりである。

1. キー値を確認して重複するエントリーが存在するか調べる。
2. 重複エントリーが存在しない場合には，マッピング値を1:1とする。
3. 重複エントリーが存在する場合には，名前の値を確認する。値が同じ場合にはマッピング値を1:1，異なる場合にはマッピング値を1:mとする。
4. 1:mの場合には，更に名前値の型を確認する。単純な文字列である場合には，通常であればその長さがある一定のしきい値（例えば100）未満であり，単なる企業名であるものと仮定して，このマッピング値を1:m-sと定める。長さが100を大幅に超える場合には，構文解析が必要なテキストのブロックが存在すると見なし，そのマッピング値を1:m-uとする。

マッピング値は状況に応じて拡張可能であるが，ここでの定義は3種類とした。現在のデータ型カテゴリの処理にはこれだけで十分であるためである。値に関する別の例を**図-4**に示す。

マッピング値の結果は，レジストリーに記録される。レジストリーの物理的位置は，環境に応じてメモリ上かファイルシステム上となりうる。レジストリー内部の値の例を以下に示す。

リコンシリエーターは，レジストリーに格納された情報に基づいて，データのグループ分けとリコンシリエーション処理に向けて後述するリコンシリエーターのサブモジュールを初期化できる。リコンシリエーションディスパッチャーのレジストリーの例を**表-1**に示す。

● **リコンシリエーター**

リコンシリエーターは，エンティティ名のリコンシリエーションタスクを処理するコアコンポーネントであり，1:1プロセッサ，1:m-sプロセッサ，1:m-uプロセッサのサブモジュール群から構成され

表-1　リコンシリエーションディスパッチャーのレジストリーの例

データソース	マッピング値	データタイプ
file://data/file1.csv	1:1	CSV
file://data/file2.ttl	1:m-s	turtle
file://data/file5.csv	1:m-u	CSV/text
file://data/file4.xml	1:m-s	XML
…	…	…

1:1マッピングの例

キー	名称	…
110	My Company Ltd.	…
120	Your Company Corp.	…
130	…	…

1:m-sマッピングの例

キー	名称	…
110	My Company Ltd.	…
110	My Company Limited	…
120	Your Company Corp.	…

1:m-uマッピングの例

Fujitsu Ltd. (富士通株式会社 Fujitsu Kabushiki-Kaisha) is a Japanese multinational information technology equipment and services company headquartered in Tokyo, Japan.[5] It is the world's third-largest IT services provider measured by revenues (after IBM and HP)

図-4　マッピング値の例

ている。各プロセッサは，マッピング値に対応したデータソースの処理を担当している。

・1:1プロセッサ

このプロセッサは，1:1マッピング値を持つデータソースを検査して一つのデータソースからの名前の値をインデックスとして使用し，残ったデータソースをペアでリコンサイルする。この処理は以下のように表される。

<データソース1> リコンサイル <データソース2> => 結果1

<データソース1> リコンサイル <データソース3> => 結果2

<データソース1> リコンサイル <データソース4> => 結果3

<データソース1> リコンサイル <データソース5> => 結果4

…

結果2，結果3および結果4がそれぞれ完了したら結果1に追加する。

このリコンシリエーション手法では，二つの名前の文字列の類似性の測定にJaro-Winkler距離（Winkler，1990）アルゴリズムなどの文字列類似性比較が用いられている。比較結果は0から1の間の値を持つスコアとなり，ここで0は類似性が存在しないこと，1は完全一致を示す。システムとしては，いかなる値であっても類似性比較に対する判定基準として採用可能である。ここでは比較実験の結果に基づき，差異のある二つの名前が同一のエンティティを指していることを確認する最低スコアとして0.98を定義した。例えば，fを二つの名前の類似性スコアを算出する関数とすると，以下の式のようになる。

$if\ f(name_1, name_2) >= 0.98,\ then\ name_1 \approx name_2$

この最低スコアの選択には自由度があるものの，より正確な結果とするために高めの基準を選択することとした。最低スコアの基準が高いがためにこの段階で除外された適格データは，以下の二つの手法を用いることで再度判定される。

・1:m-sプロセッサ

1:m-sプロセッサは，1:m-sのマッピング値を持つデータソースの名前をリコンサイルするために設計されている。二つの名前の間の関係構築には，セマンティック参照技術を採用している。この技術は，CSV，JSON（JavaScript Object Notation），Turtle（Terse RDF Triple Language）など様々なデータファイルの種類に適用可能であるが，ここではRDF Turtleを選択して説明する。この選択は，RDFが意味の表現が豊富な言語であることにもよる。

RDFデータは，ステートメント群から構成される。個々のステートメントは，「主語 述語 目的語」の形式で生成される。通常，主語，述語，および目的語のデータ型はanyURLである。例を以下に示す。

<http://dbpedia.org/page/IBM> dbpedia-owl:foundedBy dbpedia:Thomas_J._Watson

これは，「IBMはThomas J.Watsonによって設立された」というステートメントを主語（<>で囲まれたIBMを表すURL）－述語（設立されたことを意味するURL）－目的語（Thomas J. Watsonを意味するURL）によって表している。ただし，目的語型には文字列値などのリテラル値を有することもできる。

URL型の複数の値を持つ目的語では，異なる値の間の推論結果は，それらが同じ型のクラスに属するということのみとなる。例を以下に示す。

<http://dbpedia.org/page/IBM> dbpedia-owl:foundedBy dbpedia:Thomas_J._Watson

<http://dbpedia.org/page/IBM> dbpedia-owl:foundedBy dbpedia:Charles_Renlett_Flint

ここで，目的語のdbpedia:Thomas_J._Watsonとdbpedia:Charles_Renlett_Flintはいずれも人の型であり，これらの二つのURLは更に参照関係を構築することが可能であるため，異なる情報が含まれていなければならず，同一リソースではないということになる。

しかし，リテラル型の複数の値を持つ目的語では，推論結果は類似（同じ）であると解釈される場合がある。少なくともセマンティックのレベルでは，これらの二つの目的語はそれ以上参照関係を構築することが可能ではないため，その間に明らかな相違点は存在しない。更に，検査する目的語の値が，必ずしも一意である必要はないが，純粋な識別子となる名前である場合である。1:m-sプロセッサがこの型のデータを処理すると，例えば以下のようになる。

<http://dbpedia.org/page/IBM> dbpprop:name

“International Business Machines Corporation” <http://dbpedia.org/page/IBM> dbpprop:name “IBM Corp.”

「名前」または「名前相当の文字列」が含まれている場合には述語の値を探し，型がリテラルの場合にはその目的語の型に着目する。主語と述語が複数の目的語リテラル値を持つ場合には，以下のように推論可能である。

“International Business Machines Corporation” similar to “IBM Corp.”

この手法を用いると，実際には名前が同一のエンティティを表しているにも関わらず，類似性スコアが0.98未満の場合のシナリオを対象として含んでいる。このため，単に1:m-sプロセッサを用いるのと比較して，リコンシリエーション結果は更に改善可能となる。

上記のシナリオに加えて，キー値によって識別されるエンティティに属するデータ値のラベルを，エンティティ間の等価性を示すラベルの定義済みリストと比較することや，その定義済みリストにエンティティの名前の値の別名として含まれているラベルでラベル付けされたデータ値を抽出することも可能である。この機能によって，データソース構文の知識からコンテンツに特有な結果を生成するメカニズムが得られる。

このようなラベルの例には「dbpedia-owl:wikiPageRedirects」プロパティがある。「dbpedia-owl:wikiPageRedirects」が構成済みであれば，この特別な述語によってリンクされている目的語には，わずかに異なる名称ラベルで生成されたほかの項目/ページに対するリンクが含まれていることが知られている。

・1:m-uプロセッサ

このコンポーネントは，テキストブロックのような非構造化データで，複数の名前情報の値を含むものを処理するために設計されている。企業名が複数言語で記述・翻訳されている多言語テキストのシナリオにおいて，特に有効である。このプロセッサが処理可能な基本となる言語は英語であるが，技術としてはほかの言語にも適用可能である。

二つ以上の名前のマッチングを検出するためのコア技術はパターンマッチングであり，正規表現を用いてシーケンスパターン（テキスト文字列）の記述を行う。シーケンスパターンとは，ターゲットとなるテキスト内での検索パターンを構成する文字シーケンスである。次に，正規表現プロセッサがこの正規表現を非決定性有限オートマトン（NFA：Non deterministic finite automata）で処理し，正規表現にマッチする部分文字列を検出する。多くのプログラム言語には，例えばテキストのブロックから検索対象の部分文字列の抽出処理を行う正規表現機能が備わっている。Java，Python，C++などの主要言語で1:m-uプロセッサが開発されている限りは，正規表現ステートメントの構文解析は問題とはならない。以下のテキスト文字列の場合，

“Fujitsu Ltd. (富士通株式会社 Fujitsu Kabushiki-Kaisha) is a Japanese multinational information technology equipment and services company headquartered in Tokyo, Japan.[5] It is the world's third-largest IT services provider measured by revenues (after IBM and HP)”

以下のような正規表現の例を構成可能である。

Pattern="^(.)(\(.*\)) is a|an.* company|corporation"*

この正規表現に対する戻り値は次のようになる。

Fujitsu Ltd. (富士通株式会社 Fujitsu Kabushiki-Kaisha）

更に解析を行って「Fujitsu Ltd.」と「富士通株式会社」を再抽出し，パターンマッチング結果の精度を上げることも可能である。しかし，これは単純な部分文字列操作のみを必要とする比較的簡単で局所的な操作である。

・ファイナルコンバイナー

コンバイナーは，上記のステップにおいてリコンシリエーターにより生成されたリコンシリエーション結果に対するUNION手法の処理と見なすことができる。しかし，これはあらゆる演算を単純にマージしたものではなく，別の相互参照型リコンシリエーションが実行されている。この処理ロジックは，以下のとおりである。

結果集合$_1$（1:1プロセッサにより生成），結果集合$_2$（1:m-sプロセッサにより生成），および結果集合$_3$（1:m-uプロセッサにより生成）とする。結果集合$_1$の集合に含まれる名前のどれかが，結果集合$_2$の集合に含まれる一つ以上のマッチング名で検出された場合には，結果集合$_1$は結果集合$_2$にリコンサ

イル可能なものと判断する。同じアルゴリズムを結果集合$_1$と結果集合$_3$のリコンサイルに適用して，最終結果を生成する。

一旦名前のリコンシリエーションを行うと，識別子のリコンシリエーションは容易となり，データ統合モジュールであるオープンデータの連携フレームワークを用いるだけで簡単に実行可能となる。最終的なリコンシリエーション結果の例を**表-2**に示す。

連携・フレームワーク

データ連携は新しい研究対象ではなく，ここ40年にわたって様々な試みが行われてきた。その例として，1980年代初頭のデータウェアハウスや，2009年の間接スキーマ手法がある。また，2010年から2011年にかけて，セマンティック統合やデータモデリング手法により生じる問題への取組みという別の新たな研究テーマが出現した。

データソースの大半が加工されていないフォーマットで構成されており，適切に定義されたスキーマ，データモデル，およびセマンティックが欠けている現在のオープンデータ時代において，構造化されていないデータソース内に埋め込まれた情報の連携は，言うまでもなく困難である。

これらの問題への対処として，FLEはマッピングファイルに基づく連携フレームワークを開発した。単純なマッピングファイルの構築に始まり，エンティティリコンシリエーションプロセスに続いて，最後はリコンシリエーション結果に基づきマッシュアップデータの作成を行う。この連携フレームワークのアーキテクチャーを**図-5**に示す。

このアーキテクチャーのデータエクストラクターおよびストレージは，前章で説明したものと同じである。マッピングファイルエディタは，ユーザーがマッピングファイルを簡単に作成できるようにするためのコンポーネントである。これは，作成プロセス全体を通じてユーザーをガイドする，GUIウィザード形式のアプリケーションでも良い。この発想は，プロセスを可能な限り直観的なものとするとともに，ユーザーの不必要な構文エラーをなくす効果もある。マッピングファイルレジストリーには，リコンシリエーションマネージャー

表-2　リコンシリエーション結果の例

企業名	CIKコード	LEIコード	証券コードに割り当てられている企業名
International Business Machines Corp	International Business Machine Corp.	International Business Machine Corporation	International Business Machines Co.

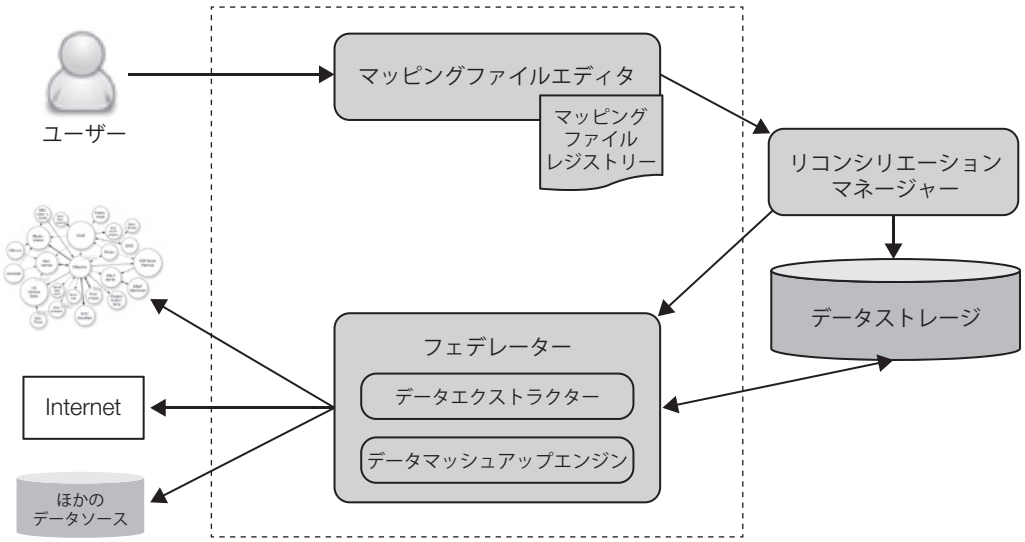

図-5　フェデレーション・フレームワークのアーキテクチャー

およびフェデレーターが使用するマッピングファイルが全て格納されている。リコンシリエーションマネージャーおよびフェデレーターは，この創案の中核部であるマッピングファイルを提供する必須コンポーネントである。以下で詳しい紹介を行う。

● マッピングファイル

ユーザーが，CICIユーティリティが発行する取引主体識別子LEIや，EDGAR企業ファイリングシステムが発行するCIKから，各種会計監督機関（詳細については後述）が発行する企業識別子（例：IBM，Microsoft，Apple）を，株式市場が発行するティッカーシンボルとともに探して，その結果を一つのデータベースに統合したいというシナリオを考える。その場合，データは単独のCSVファイルか，RDBMSのような形式で結合されなければならない。このソリューションでは，様々なWebサイトからLEI，CIK，およびティッカーシンボルを手作業で検索するたびに，一つひとつの企業名を用いてその結果を手動でつなぎ合わせることもできる。しかし，この方法は，企業数が少なければ実行可能であるが，100企業を超えると非常に時間のかかる作業となる。ユーザーが様々な報告書，もしくはデータベースは同じであっても異なるシステムから，複雑な会計データを必要とした場合は言うまでもない。

連携フレームワークおよびそのマッピングファイルの発想は，オープンデータソースのスキーマやデータモデルを解析して媒介スキーマを見つけ出すことなく，データの統合を実現するという要件を満たすためのものである。ユーザーに対して「何を持っているか」ではなく「何を必要としているか」を問いかけることから，マッピングファイルによる連携プロセスが開始される。マッピングファイルはユーザーに対するテンプレートとしての役割を果たし，ユーザーがデータをいかに統合したいのかをシステムに伝える。上記と同じシナリオを用いた，企業名，CIK識別子，LEI識別子，およびティッカーシンボルが含まれているマッピングファイルの例を**図-6**に示す。

このマッピングファイルには，以下の4種類の概念が定義されている。なお，マッピングファイルの中でそれぞれの概念の値はセミコロンで区切られており，それぞれの行もやはりセミコロンで終端されている。ここで，マッピングファイルの出力はCSVファイルとした。その理由は，CSVファイルはプレーンテキスト形式であり，不必要なエラーを発生させることなくどのシステムでも容易に処理可能であるからである。また，CSVファイルは最終出力として用いることも中間フォーマットとして用いることも可能であり，後者の場合には，後でそれ以外のどのようなフォーマットにも変換可能である。出力CSVファイルの各カラムは，マッピングファイルの各行に対応している。

1. *type*

ここでは，キーおよびマップの二つの種類が定義されている。一つのデータソースは，マッピングファイルのキーと定義済みキーのマップのどちらでも良いことを示している。この定義方法は，RDBのテーブルにおけるプライマリキーに似ているが，RDBのものより柔軟なものとなっている。

2. *name*

行の名前を定義する。この行の名前は，生成されたCSVファイルのベースとなる。

3. *pattern*

データソースの中でどのような種類のデータを探すかをシステムに理解させるためのものである。プリミティブ型やデータが埋め込まれた文字列パターンであっても良い。例えば，あらゆる5桁の数値を表すためには00000を用いる。これは，特に非

```
type=key; name=Company_Name; pattern=String; source=file:///company_name.data;
type=map; name=CIK_Code; pattern=String; source=http://www.sec.gov/edgar/searchedgar/cik.htm;
type=map; name=LEI_Code; pattern=String; source=https://www.ciciutility.org/search.jsp;
```

図-6　企業識別子マッピングファイルの例

構造化データの検索では強力な概念となる。

4. *source*

データの読出し可能な場所を示す。ソースの値は，http/https形式のURLやファイルシステムのパスとなる。ファイルシステムにおけるパスの取得には，system.propertiesファイルを構成するか，ソフトウェアから与えられるデフォルトパスを用いる。

表形式のデータソースには，特別な表記法も定義されており，フレームワークが各セルの正確な値を見つけることが可能となっている。以下のマッピング情報を例にとって説明する。

[type=key; name=LEI_Code; pattern=String; source=file:///query1.csv#Identifier@byColumn<LINK>file:///generated/csv/company-name_cik_lei_ticker-symbol.csv#LEI_Code@bySearch;]

この情報は，**表-3**のように整理される。ここで留意すべきは，上記の四つの概念は非常に基本的な設計であることである。今後，type，name，pattern，およびsourceで十分でない場合，新たに概念を拡張し，マッシュアップエンジンを強化していくことで，より充実したマッシュアップを実現していく。

● シーケンスファイル

シーケンスファイルは，マッピングファイル間の依存問題を解決するために設計され，複数のマッピングファイルを必要とする場合は，データワークフロープロセスと同様の動きを行う。例えば，識別子マッピングに加えて，EDGARおよびNASDAQが作成した2014年度版の財務諸表に関する報告書もユーザーが必要とすることがある。この場合，以下の二つのマッピングファイルを順に用いることでこの処理が可能となる。

1. 図-2に示す手順で企業識別子マッピングファイルを定義する。
2. 2014年度財務諸表マッピングファイルを定義する。会計数値は，ステップ1により生成された識別子を用いて取得する。

そしてシーケンスファイルの中に，これらの二つのマッピングファイルを以下のように簡潔に記述する。

Company-identity.map

Fiscal-year-2014-statement.map

フレームワークがこのシーケンスファイルを処理する際には，マッピングファイルの順番に従ってデータの処理が行われる。

表-3　マッピング情報

#	「#」キーの後の文字列は，この表形式ファイルでのコラムの名前を表す。
@byColumn	この値検索は，このコラムの値を使って実行することをシステムに伝える。
@bySearch	この値検索は，基本キー値を与えてCSVファイル全体を検索することにより実行することをシステムに伝える。
<LINK>	これは，二つの異なったCSVファイルを通じてある値を検索する状況である。これにより，データが1か所では検出が容易ではない場合にトンネリングが可能となる。

● データマッシュアップエンジン

マッシュアップエンジンは，最初のコラム（type＝keyであるコラム）の値をメモリに読み込む。図-6を例にとると，これは「company_name.data」からの企業名のリストとなる。.dataファイルは，連携フレームワークとともに考案されたプロプライエタリなファイル種別であることに注意が必要である。ここには，ほかのデータソースからそれ以外の値を検索するインデックスとして使用可能な値のリストが含まれている。マッシュアップエンジンが「International Business Machine Corp.」という値を読み込むと，表-2のリコンシリエーション結果を参照して，SECのサイト[(4)]からCIKコード（EDGARが発行する識別子）を抽出・検索する名称として「International Business Machine Corp.」を見つける。同様に，LEIコードを抽出・検索する名称として「International Business Machine Corporation」を，*file:///company_ticker_symbol_list.csv#Symbol@bySearch*に存在するファイルからティッカーシンボルを検索する名称として「International Business Machines Co.」をそれぞれ見つける。この処理の例を**表-4**に示す。ここで留意すべきは，上記の表に含まれている内容は，図示のための非常に簡単な例であり，実際には同じ処理ロジックでより複雑な連携の実行が可能であるという点である。

表-4　フェデレーション結果

企業名	CIKコード	LEIコード	証券コードに割り当てられている企業名
Microsoft Corp	0000789019	INR2EJN1ERAN0W5ZP974	MSFT
International Business Machines Corp	0000051143	VGRQXHF3J8VDLUA7XE92	IBM
…	…	…	…

想定される用途

FLEは，初期プロトタイプをベースとして，このソリューションを金融と旅行の二つの異なる用途に適用した。

●　金融用途

2009年のリーマンショック危機における反省として，現在世界中のデータをいかに容易に統合してデータ分析ができるかがグローバル経済の安定をもたらすとして，各種標準化を始め各国で様々な取組みが進んでいる。金融用途については，この流れに乗って世界中の企業に関する情報の収集を目標とした。収集する情報は，名前（公式に登記された名前，およびほかのオープンデータソースからの別名），住所，説明，各種の組織や監督機関から発行された様々な識別子，合併および買収，関連のある会社（競合会社，パートナー，親会社，子会社など）といった内容が含まれるが，これに限定されるものではない。

そこでFLEは，最初に企業名に関する最も信頼性の高いソースとして，LEI[5]データの選択を行った。

例えば，米国に主要な拠点を置く企業については，GMEIユーティリティ[6]からダウンロードしたデータが主要なソースとなる。ある企業にLEIが存在しない場合には，二次ソースとして取引主体の法的登記名を用いる。英国の場合にはCompanies House（会社登記所）[7]が取引主体に当たるが，米国では州により異なる。米国にある企業であれば，それ以外にも米国証券取引委員会[4]が発行するCIKという重要な識別子を持っている。それに加えて，全ての大手証券取引所では，そこで取引を行う企業に対してティッカーシンボルを付けている。その好例としては，NASDAQ，ニューヨーク証券取引所，ロンドン証券取引所，東京証券取引所などがある。また，DBPedia endpoint[8]でも，世界中の85 583企業に関する情報提供を行っている。

これらのデータを信頼できる情報として，そのほかのオープンデータの企業名を開発したリコンシリエーション手法でリコンサイルすることにより，多様な形式のデータソース全体の名前マッピングを生成できる。このため，それぞれのデータセットに合わせて同一の企業エンティティを参照する正確な識別子を取得可能である。これによって，同一企業に関するオープンデータをより正確な方法で自動的に連携できるようになり，これまでとは違った新たな分析手法を容易に実現できるようになる。

●　旅行用途

もう一つの例として，旅行情報の中でも特に遺跡のデータの連携にEIDERプラットフォームを適用した。ここでは，スペインのアンダルシア地方の遺跡情報の収集に焦点を合わせた。スペインをはじめとするヨーロッパ各国は観光が産業として確立しており，膨大なオープンデータに間違った情報があると観光産業へのインパクトが懸念される。そこで，本技術を用いて公開されているデータを統合することでこのような不具合を容易に見つけ出し，適切な処置を迅速に行えるようになる。

FLEが収集した情報には，各種ソースからの様々な名前，地元の識別子，緯度経度による位置，芸術の種類，建築年などといった内容が含まれる。データソースには，Mosaico,[9]スペイン版DBPedia[10]およびYelp[11]が含まれる。YelpデータはREST APIから抽出し，JSONファイルとしてダウンロードする。スペイン版DBPediaデータは，公開エンドポイントに対してSPARQLクエリを送信することにより取得するが，合計で1351件のモニュメントの情報が返送されてくる。

FLEのリコンシリエーションロジックを適用することにより，単なる文字列によるマッチングに比べ，2倍のマッチング結果をもたらした。同時に，精度の高い方法で同一モニュメントに関する多様

なデータを連携することが可能となった。

むすび

オープンデータのエンティティ識別問題は，まだ比較的新しいトピックである。これまでに欧州富士通研究所が認識している参考文献は，JamesとNigelによって2014年に発表されたホワイトペーパー[12]が唯一のものである。しかし，2種類の用途におけるデータ統合結果には，従来手法との比較でかなりの進歩が既に示されており，技術的見地とビジネス的見地の両面から大きな関心を集めている。

今後も，リコンシリエーションマネージャーを通じてより多様なアルゴリズムを組み込むことにより，研究成果を更に進展させ，定期リフレッシュとオンデマンドリフレッシュ技術によりデータ鮮度を確保していく所存である。

参考文献

(1) S. R. Auer et al. ：DBPedia：A Nucleus for a Web of Open Data. Lecture Notes in Computer Science 4825, p.722-735, Springer-Verlag (2007).

(2) Socrata：2014 Open Data Benchmark Report. http://discover.socrata.com/2014-benchmark-report-email-thank-you.html?aliId=7973252

(3) OpenRefine. http://openrefine.org

(4) U. S. Securities and Exchange Commission：EDGAR Company Filings｜CIK Lookup. http://www.sec.gov/edgar/searchedgar/cik.htm

(5) LEI ROC. http://www.leiroc.org

(6) GMEI Utility. https://www.gmeiutility.org/

(7) Companies House. https://www.gov.uk/government/organisations/companies-house

(8) DBPedia SPARQL Endpoint. http://dbpedia.org/sparql

(9) The database of the Intangible Cultural Heritage of Andalusia (MOSAICO). http://www.iaph.es/patrimonio-inmaterial-andalucia/

(10) Spanish DBPedia SPARQL Endpoint. http://es.dbpedia.org/sparql

(11) Yelp. https://www.yelp.com/developers/documentation/v2/search_api

(12) J. Powell et al. ：Creating Value with Identifiers in an Open Data World. Open Data Institute and Thomson Reuters, March 2014.

著者紹介

Vivian Lee
欧州富士通研究所 所属
現在，ビッグデータとオープンデータのインテグレーション技術，フェデレーション技術，リコンシリエーション技術の研究開発に従事。

伊豆哲也（いず　てつや）
欧州富士通研究所 所属
現在，ビッグデータ利用技術の研究開発に従事。

後藤正智（ごとう　まさとも）
欧州富士通研究所 所属
現在，ビッグデータとオープンデータの解析研究およびXBRLの標準化に従事。XBRLインターナショナルのXBRL標準化委員会のメンバーであると同時に，IFRSFのIFRS分類法協議グループのメンバーでもある。