データ相互運用問題を支える技術

加藤弘之（国立情報学研究所）

データ相互運用問題とは？

簡潔に説明するとデータ相互運用問題とは，以下のことである．データベースは「雨後の筍」☆1のようにあちらこちらで独立に作成されている現状がある．これら独立に作成されたデータベースには同じ意味を表すデータが異なるデータベースの形式（以下，データベーススキーマと記す）のもとで存在する．このようなデータに対して統一的にアクセスするための枠組みを提供するのがデータ相互運用の目的である．ビジネス界，コンシューマ，データ科学の3つの観点から簡単な具体例をそれぞれ示す[1), 2)]．1つ目はビジネス界から以下の例がある．米国の企業Aが欧州の企業Bを買収する際，企業Aは企業Bがこれまで作成したデータベースを自社のデータベースと統合して使いたいが，一方で，企業Bのデータベーススキーマやデータの作成手法はこれまでの独自の手法があったり，欧州独自のデータの作成手法があるため，引き続き同じデータベースを使用していきたい．このような場合，企業Aは買収効果を得るために，企業Aのデータベースと企業Bのデータベースを統合して，同じ意味を表すデータ（たとえば，顧客売上データ）に統一的にアクセスする必要がある．2つ目はコンシューマからの視点で，以下の例がある．数多くある職探しのWebインタフェースには，多少異なるが同じような項目の入力が要求されている．このような場合も，各Webインタフェースを統合して同じような項目の入力が1回ですまされるようにしたい．3つ目はデータ科学（生物学やエコシステムなど）から以下の例がある．バイオインフォマティクスにおいて，独立に管理されている3つのデータベースGUS（Genomics Unied Schema）☆2とBioSQL☆3，uBio☆4について，これら独立した3つのデータベースを統合して利用することで，生物学者が協調して新たな生物学の知見を得ることができる[2)]．これらに共通する解決手法は，データベースの違いをうまく吸収する手法（後で述べるスキーママッピング）を用いることである．しかしながら，データベースの違いを記述しやすい手法を用いて吸収した場合，必要となるデータの検索には限界があることが分かっている．本稿ではこの問題に対する最近の取り組みについて紹介する．

このように幅広い分野での応用が存在するデータ相互運用問題が議論されるようになったのは古く，1977年にIBMサンノゼ研究所におけるEXPRESS（EXtraction, Processing, and REStructuring System）プロジェクトで，階層データベース間の変換について議論されたように，データベース研究の初期の段階からその重要性は認識されていた．そして現在も，新しい技術や応用とともに発展し続けており，データベース研究分野における著名な国際会議であるACM SIGMOD (Special Interest Group on Management of Data) /PODS (Symposium on Principles of Database Systems)，VLDB (Vary Large Data Base)，CIDR (Conference on Innovative Data System Research) などで毎回この問題に

☆1 北米のある大学の講義では「マッシュルーム」のようにといった表現を使っていたりする．

☆2 http://www.gusdb.org
☆3 http://bioperl.org
☆4 http://www.ubio.org

関するセッションが設けられている．また，これらの研究成果に基づくさまざまな製品が存在するということも強調しておきたい．

★何が難しいのか

データ相互運用問題の難しさは「システムの問題」，「非技術的問題」，「論理的問題」の大きく3つに分類される．本稿では「論理的問題」についてその解決手法について紹介する．

システムの問題

異なるシステム上に独立に作成されたデータベースを統一的に利用するためには異なるシステムを統合する必要がある．たとえ，同じ関係データベースでも異なるベンダによるものはSQLの文法が微妙に異なっていたり，記述能力にも違いがあったりする場合がある．

非技術的問題

非技術的な問題として，有用なデータをいかにデータベースに登録するかという問題がある．データの所有者は自分が提供するデータが無制限にほかの利用者に利用されることに懸念を持っている．特に医療記録や法の執行などのように個人情報を含むデータは匿名化などを用いることで，データの所有者の懸念は払拭可能となる．データの匿名化は最近の計算機科学のホットな話題の1つである．また，データの所有者に適切な評価を与えることで，データ提供のインセンティブとなる枠組みが必要とされており，これに関する議論も始まったばかりである．

論理的問題

同じ意味を持つデータを管理するためのデータベースが独立に作成されているとき，データベーススキーマが一致することはまずあり得ない．複数のデータベースを統合するには，これらスキーマの違いに対応する必要がある．実際，データ相互運用問題においてスキーマの違いが主要なボトルネックである．以下の節ではこの問題に対するアプローチを紹介する．

★2つのアプローチ：データ交換とデータ統合

データ相互運用問題に対するアプローチには，2つのアプローチ，データ交換「data exchange」とデータ統合「data integration」がある[3]．これら2つの大きな違いは，データ交換はデータを重複して持つことで検索の高速化を重視しているのに対して，データ統合ではデータは重複して持たずに問合せの書き換えを通じて最新のデータを検索することを重視している．またこれら2つに共通する点は，データベーススキーマの違いを記述する点（以下，スキーマの違いの記述をスキーママッピングと記す）にある．どのような道具（言語）でスキーママッピングを記述するかについては，以下の観点が挙げられる．

- スキーマの違いを宣言的に記述できること．
- 十分な記述能力を有すること．
- 既知の成果が使えること．

これらの観点からスキーママッピングは一階述語論理で記述されるべきとの合意のもとで研究が進められている．本稿ではこれら2つのアプローチおよびこれら2つを組み合わせたPeer2Peer（以下，P2Pと記す）アプローチについて紹介する．スキーママッピングが記述できればそれで問題が解決されるわけではない．本稿では，適切なスキーママッピングが記述されたとしても解決しなければならない問題に焦点を当てて解説する．なお，本稿ではスキーママッピングの自動生成については記述しない☆5．

データ交換（data exchange）によるアプローチ

前述したように，データ交換では相互運用したいデータを重複して持つことによって検索の高速化に重点を置いている．**図-1**はデータ交換の概念を示している．ソーススキーマSのもとに存在しているデータ（以下，データベースインスタンスと記す）Iはターゲットスキーマ**T**に合うように変換されデータベースインスタンスJとして存在する．したがって，ターゲットスキーマ**T**に対する問合せはJを使って効率良く結果を得ることができる．IをJ

☆5 製品レベルではドメイン知識を入力することでスキーママッピングを半自動化するようなツールも存在する．

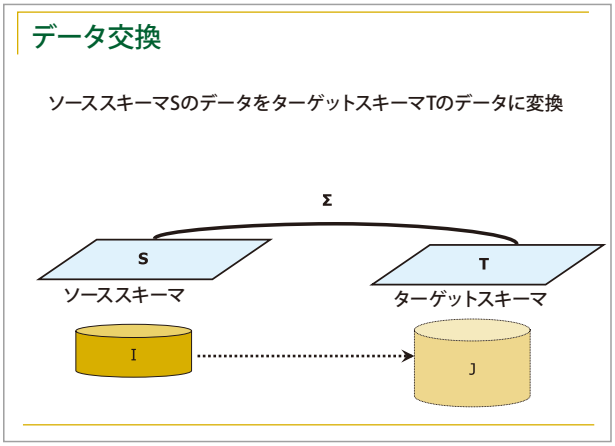

図 -1　データ交換の概念図

に変換する手助けをするのがデータ交換におけるスキーママッピングΣである．データ交換における主な研究課題は，与えられたスキーママッピングからどのようなデータベースインスタンスを生成するかにある．後述するように，生成されるべきデータの性質として，情報の損失がない汎用解（universal solution）と，汎用解の中の最小のデータ集合（core universal solution）がある．

★制約としてのスキーママッピング

以下，説明のための簡単な例を示す☆6．ソースデータベースは1つの関係 Teaches から構成されており，たとえば，Teaches（鈴木，田中）は教授の鈴木は学生の田中をあるクラスで教えていることを表している．関係 Teaches には以下のデータが格納されているものとする．

Teaches

教授（p）	学生（s）
鈴木	田中
佐藤	吉田

一方，ターゲットデータベースは2つの関係 TeachesCourse と Takes から構成されており，TeachesCourse（鈴木，DB）は，教授の鈴木は DB コースを教えていることを，Takes（田中，DB）は，学生の田中は DB コースを履修していることを，それぞれ表現している．このとき，ソースデータベースとターゲットデータベースの間の制約として次のスキーママッピング☆7について考える．Teaches (p, s) →∃ cTeachesCourse (p, c),Takes (c, s) このスキーママッピングは，Teaches に格納されているすべての組 (p, s) について，あるコース c が存在し，TeachesCourse に (p, c) という組が，Takes に (c, s) という組が存在するという制約を表している．

★スキーママッピングによって生成されるデータ

上記のスキーママッピングを満たすデータ集合（以下，解と記す）を以下に示す．

TeachesCourse

教授（p）	コース（c）
鈴木	C1
佐藤	C2

Takes

コース（c）	学生（s）
C1	田中
C2	吉田

ターゲットデータベースのある解 J_1

データ交換の解の中には2種類の値が存在する．1つはソースデータベース I からの値（たとえば，鈴木）であり，もう1つは I にない新しい値（たとえば，C1）である．この新しい値はターゲットデータベースの不確実な情報を表現しており変数とみなすことができる☆8．より正確には，この変数には何らかの代入が存在するが，具体的な代入は分からないということである．たとえば，J_1 中の C1, C2 は，鈴木と佐藤が教えているクラスだが，具体的なクラス名は分からないということを表している．

実は，スキーママッピングを満たす解は複数存在する．この複数の解のうち，より良い解に関する2つの性質，汎用解（universal solutions）と汎用解の核（core universal solutions）が議論されている．汎用解とは情報の損失のない解であり，汎用解中で

☆6　あくまでも説明のための例であり，実際には複雑なものになっている．

☆7　データ交換におけるスキーママッピングは，組生成従属性（tuple-generating dependencies）を用いて記述される．関係データベースにおける従属性は 1970 年代および 1980 年代に精力的に研究された分野であり，組生成従属性もその 1 つである．

☆8　ラベル付き Null 値と呼ばれることもある．

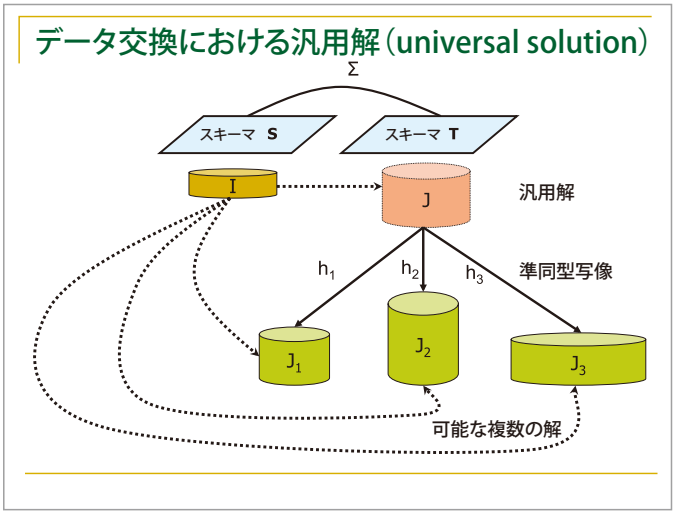

図 -2　汎用解（universal solution）の概念図

最小のものが核（core）と呼ばれている☆9．例を用いて汎用解と核について説明する．上記の解 J_1 以外にも，たとえば以下の J_2 も上記のスキーママッピングの制約を満たしている．

TeachesCourse

教授（p）	コース（c）
鈴木	C1
佐藤	C1

Takes

コース（c）	学生（s）
C1	田中
C1	吉田

ターゲットデータベースのある解 J_2

J_2 では鈴木と田中が教えているコースに同じ値（変数）C1 が用いられているので，鈴木と田中は同じコースを教えていることを表している．しかしながら，ソースデータベースとスキーママッピングにそのような情報はない．したがって，J_2 はスキーママッピングを満たしているが，ほかの解（たとえば，J_1）よりも汎用ではないことが分かる．直感的に，汎用解はソースデータベースとスキーママッピングから見て，いかなる情報の損失も余分な情報の追加もないものとみなすことができる．汎用解の性質として，**図 -2** に示したように，すべてのほかの解に対する準同型写像が存在することが挙げられる．

実は汎用解自身たくさん存在することが知られている．たとえば，以下の J_k は任意のサイズを持っている汎用解である．

TeachesCourse

教授（p）	コース（c）
鈴木	C_1
佐藤	C_2
…	
鈴木	C_{2k-1}
佐藤	C_{2k}

Takes

コース（c）	学生（s）
C_1	田中
C_2	吉田
…	
C_{2k-1}	田中
C_{2k}	吉田

任意のサイズを持つ汎用解 J_k

実用上，汎用解のうち最小の汎用解が望ましい．この最小の汎用解は核（core）と呼ばれており，ある同型のもとで唯一（unique up to an isomorphism）であることが知られている[1]．先に示した J_1 はこの例における汎用解の核である．

データ統合（data integration）によるアプローチ

データ統合では，データを重複して持たずに問合せの書き換えを通じて最新のデータを検索することを重視している．**図 -3** はデータ統合の概念図を示している．ソースデータベースのスキーマをスキーママッピングを用いて仮想的に統合しスキーマ **T** とする．そして，このスキーマ **T** に対する問合せは，スキーママッピングを通じて各ソースデータベースに対する問合せに書き換えられる．データ統合における主な課題は，統合データベースに対する問合せをソースデータベースに対する問合せにどのように書き換えるかにある．データ統合ではスキーママッピングは変換として記述されているため，変換の方

☆9 汎用解と汎用解の核を求めるための効率的なアルゴリズムが提案されているが本稿では触れない．興味のある方は文献 3）などを参照されたい．

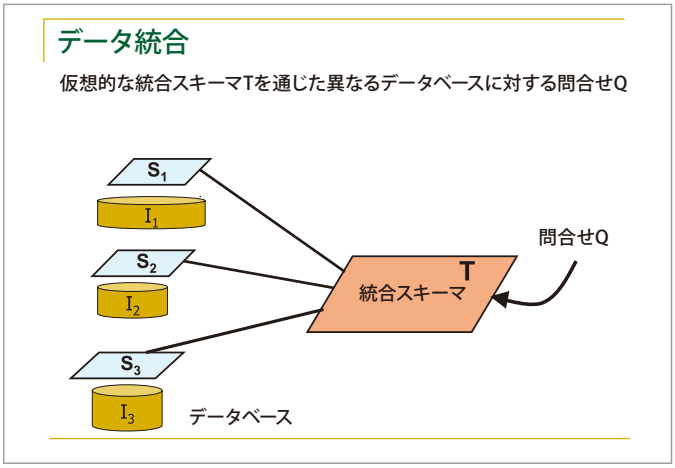

図 -3　データ統合の概念図

向によって 2 つの異なる問合せの書き換え手法がある．ソースデータベースのデータを統合スキーマの構造を持つデータに変換するようなスキーママッピングを用いた手法は Global-As-View（GAV）アプローチと呼ばれるのに対して，逆方向つまり，統合スキーマの構造を持つデータをソースデータベースのスキーマ構造のデータに変換するスキーママッピングを用いた手法は Local-As-View（LAV）アプローチと呼ばれ，それぞれ以下のような特徴がある．

	GAV	LAV
モジュール性	×	○
ソースの情報無損失性	△	○
問合せ書き換えの容易性	○	△

このような特徴を踏まえ，GAV はソースデータベースの種類の数が少なくかつ，ソースデータベースの種類の追加削除が起こらない安定した状況で使われる場合に向いている．これに対して，LAV はソースデータベースの種類の数が多くかつ，ソースデータベースの種類の追加削除が頻繁に起こるような状況で使われる場合に向いている．以下，GAV と LAV について簡単な例を用いてその特徴を述べる．

★Global-As-View（GAV）アプローチ

GAV アプローチは，ソースデータベースを統合スキーマの構造に変換するスキーママッピングを用いた手法である．特徴として，問合せの書き換えは，単にこのスキーママッピングを展開することで得られる．一方で，ソースデータベースの情報が損失される場合がある．

以下，簡単な例を用いて説明する．映画に関する統合スキーマとして 2 つの関係，Movie（title, dir, year, genre）と Schedule（cinema, title, time）を考える．関係 Movie は，題名（title）と監督（dir），公開年（year），ジャンル（genre）から構成され，関係 Schedule は，劇場（cinema），題名（title），上映時刻（time）から構成されているものとする．今，ソースデータベースが 2 つの関係 S1（title, dir）と S2（title, year, genre）から構成されているとすると，S1, S2 から統合スキーマ Movie（以下，問合せ式中では M と記す）へのスキーママッピングは，以下のデータログ問合せ[☆10] で表現できる．

$$M(t, d, y, g) \leftarrow S1(t, d), S2(t, y, g)$$

この問合せは，S1 に格納されているすべての組 (t_1, d) と S2 に格納されているすべての組 (t_2, y, g) について，タイトルが同じデータ（$t_1 = t_2$）を関係 Movie の (t_1, d, y, g) に変換していることを示している．このとき，統合スキーマに対して，公開年が 2000 年以降の映画の題名と監督を検索する以下の問合せを考える．

$$A(t, d) \leftarrow M(t, d, y, g), y > 2000$$

この問合せは，スキーママッピングを展開することで，以下のようにソースデータベースに対する問合せに簡単に書き換えることができる．

$$A(t, d) \leftarrow S1(t, d), S2(t, y, g), y > 2000$$

次に，劇場と上映している映画のジャンルに関する情報を保持しているソースデータベース S3 (cinema, genre) が追加された場合のことを考える．S3 から統合スキーマ Movie と Schedule（以下，問合せ式中では S と記す）へのスキーママッピングはそ

☆10 データログは SQL 問合せの論理的な表現手法として知られている．論理に詳しい方は，単なるホーン節と見てほしい．

れぞれ以下のようになる．

$$M(null, null, null, g) \leftarrow S3(c, g)$$
$$S(c, null, null) \leftarrow S3(c, g)$$

このとき，統合スキーマに対する問合せとして，コメディを上映している劇場の検索ができなくなる．S3 をそのまま使えば検索できるのに統合スキーマに変換してしまったためにできなくなってしまう．つまり，統合スキーマに変換することでソースデータベースの情報を失ってしまっていることが分かる．新しいソースデータベースの追加に適応して，統合スキーマを修正すると，今度はこれまで記述したすべてのスキーママッピングを書き換える必要がある．このように，GAV では問合せ処理は単純であるが，ソースデータベースの追加（削除も同様）に対するモジュール性が欠落している．

★Local-As-View（LAV）アプローチ

LAV アプローチは，統合スキーマの構造を持つデータをソースデータベースのスキーマの構造のデータに変換するスキーママッピングを用いた手法である．その特徴は，モジュール性があり，ソースデータベースの情報を損失することはないが，問合せの書き換えが難しいことが知られている．以下，GAV の説明で使用した映画に関する統合スキーマ Movie，Schedule とソースデータベース S3 を用いて簡単に説明する．まず，S3 が新たに追加された場合，以下のようなスキーママッピングを記述すれば良い．

$$S3(c, g) \leftarrow M(t, d, y, g), S(c, t, t')$$

このとき，上記と同様に，統合スキーマに対して，コメディを上映している劇場を検索する以下の問合せについて考える．

$$Q(c) \leftarrow M(t, -, -, \text{"コメディ"}), S(c, t, -, -)$$

なお，上記の問合せ中の "－" は問合せ式の中で一度しか現れない変数を置換したものであり，匿名変数（anonymous variables）と呼ばれる．この問合せは，M と S に対する問合せを S3 を用いて検索することになる．実はこのような問合せ処理は，本質的には「ビューを用いた問合せ処理（Answering queryusing materialized views）」である．なぜならば，S3 は M と S を用いたビューとみなすことができ[☆11]，Movie と Schedule に対する問合せをビューである S3 を用いて処理するからである．このビューを用いた問合せ処理に関してはすでに数多くの研究が存在している[4]．たとえば，スキーママッピングの逆変換を用いた手法を用いると，上記の問合せは次のように処理することができる．

簡単のため，匿名変数を省いた以下の問合せについて考える．

$$Q(c) \leftarrow M(t, \text{"コメディ"}), S(c, t)$$

同様に一度しか現れない変数を省略した以下のようなスキーママッピングを考える．

$$S3(c, g) \leftarrow M(t, g), S(c, t)$$

このスキーママッピングについて，S3 に組，たとえば（"A", " コメディ "）が存在するということは，Movie 中にある題名 T1 の組（T1, "コメディ"）が存在し，Schedule 中に同じ題名 T1 で組（"A", T1）が存在することを表している．S3 中のほかのデータについても同様のことがいえ，このことを利用すると，S3 から Movie と Schedule に**図 -4** に示すようなデータが存在することが分かる．この同じ題名 T1, T2, T3, ... を表すための関数 f[☆12] が存在しこれを用いることで，以下のようなスキーママッピングの逆変換を得ることができる．

$$M(f(X, g), g) \leftarrow S3(X, g)$$
$$S(c, f(c, Y)) \leftarrow S3(c, Y)$$

このスキーママッピングを使うことで，先の問合せである「コメディを上映している劇場」を検索することができる．

☆11 なので，Local-As-View と呼ばれている．
☆12 スコーレム化関数をを用いて 9 を取り除くという標準的な手法を用いている．

S3

劇場（c）	ジャンル（g）
A	コメディ
B	SF
B	コメディ

Movie

題名（t）	ジャンル（g）
T1（=f（A, コメディ））	コメディ
T2（=f（B, SF））	SF
T3（=f（B, コメディ））	コメディ

Schedule

劇場（c）	題名（t）
A	T1（=f (A, コメディ)）
B	T2（=f (B, SF)）
C	T3（=f (B, コメディ)）

図 -4　スキーママッピングの逆変換を用いた S3 から M と S へのデータ変換

このように，LAV アプローチではソースデータベースの情報損失は起きないが，問合せ処理は困難で，上記の例ではうまく処理できたが，一般には難しいことが知られている．

P2P (peer2peer) アプローチ

これまで述べてきたデータ交換にしてもデータ統合 (GAV, LAV) にしても，システム全体を把握する管理者の存在を仮定しており，システム全体を考慮した統合スキーマの存在や制約も仮定している．しかしながら，実際問題としてシステム全体を把握する管理者を仮定することや，1 つの統合スキーマを設計することは困難である．また，独立した異なるソースデータベースからのデータには矛盾するような内容が含まれる場合がある．このような実用上の問題点に対応したのが，P2P に基づくアプローチである．P2P アプローチには，単に P2P アプローチを採用した PDMS（Peer Data Management System）と，独立した異なるソースデータベースからの矛盾に対応した CDSS（Collaborative Data Sharing System）がある．

★PDMS（Peer Data Management System）

PDMS は，データ統合に基づき，各ピア（各ソースデータベース）が，ほかのピアとの違いをスキーママッピングを用いて記述する手法である．これにより，全体として 1 つの統合スキーマを設計せずにすむため，より現実の問題に適している．PDMS の概念図を**図 -5** に示す．問合せは自分がよく知っているピアに対してなされ，そのピアにスキーママッピングで繋がっているすべてのピアからデータを検索することが可能となる．今，ピア（ソースデータベース）p1 を p2 に変換しているスキーママッピングを $f_{1\to 2}$ とすると，p2 に対する問合せは GAV アプローチにより p1 のデータも検索可能となるし，p1 に対する問合せは LAV アプローチにより p2 のデータも検索可能となる．このように，PDMS ではあるピアに対する問合せは，スキーママッピングの向きに応じて GAV と LAV を組み合わせることで，そのピアにスキーママッピングで繋がっているすべてのピアのデータも検索可能となる．PDMS のもう 1 つの利点は，モジュール性にある．あるピアが PDMS に参加する場合，自分と最も似ているスキーマを持ち熟知しているピアとの間にスキーママッピングを記述すれば良いし，PDMS からの離脱も単に自分と繋がっているスキーママッピングを削除するだけで良い．

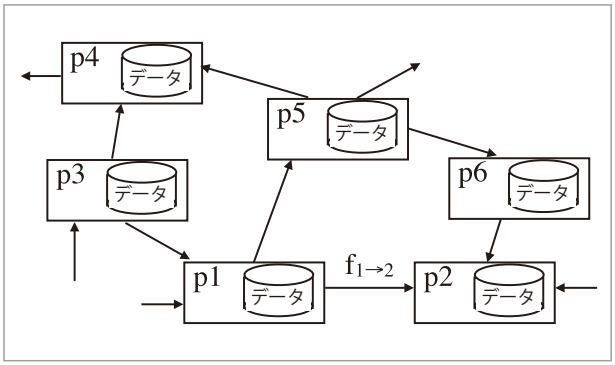

図 -5　PDMS の概念図

PDMS における主な研究課題は，問合せの最適化にある．p2 に対する問合せを例に説明する．この問合せは，p1 を通じて p5 のデータ検索と p6 を通じた p5 のデータ検索，さらには p1, p3, p4 を通じた p5 のデータ検索の 3 通りの検索が考えられる．これら 3 通りの問合せをそのまま実行するのは冗長である．これら 3 通りの問合せを分析し効率的な問合せ

を求めたい．一般に，与えられた2つのSQL問合せの包含関係の判定は困難であることが知られており，さまざまなアプローチによる研究が存在する．

★CDSS（Collaborative Data Sharing System）

かつては信頼できる専門家によって作成されることでその質が保証されていたデータベースは，近年インターネット上でさまざまな人たちによって作成，コピー，移動されている．このような状況では，データの質は保証されず，さまざまな質のデータが混在している．先に述べたP2Pアプローチの利点であるモジュール性により，自分が参加しているP2Pのシステムにどのような質のデータが混ざっているか事前に知ることは困難である．CDSS[2]では，データの出所情報と来歴情報を用いて検索結果のデータがそもそもどのピアからきたのか（出所情報），どのようにしてきたのか（来歴情報）が分かるような枠組みを提供している．この枠組みにより，データの信頼度が計算できるようになっている．特筆すべきは，SQLに代数的基礎を与えている関係代数を，代数構造の一種である半環構造（semiring）を用いて抽象化し，この半環構造を用いてデータの来歴情報（Provenance）の計算の枠組みを構築した点にある．さらに，この半環構造は，従来の集合に基づく関係代数だけでなく，重複を許すバッグや確率データの計算など，これまでに関係データベースが対象とするデータとして拡張してきたあらゆるデータ構造に適用可能なものになっている．CDSSの論文[2]は，2017年のACM PODSにおいて，10年前の論文のベストペーパーに与えられる，「test of time award」の有力候補といわれている．

今後の展望

「データ相互運用問題は，古くて新しい問題である．」と，データベース研究の第一人者であるPilip Bernstein[☆13]が言及しているように，今後も新しい応用と技術とともに，さらなる研究が進むことが予想される．たとえば，2016年の4月に開催されたダグストゥールセミナー[☆14]では，ビッグデータのデータ相互運用問題のために，ビッグデータの来歴情報をどのように定義すれば良いかについての議論があった[5]．また，これまではあまり扱われていなかった「更新」の扱いも重要である．現実世界は常に変化し続けており，この変化に迅速に対応できるような，データ相互運用システムの構築も重要な課題である．

参考文献

1) Doan, A., et al. : Principles of Data Integration, Morgan Kaufman, ISBN:978-0-12-416044-6 (2012).
2) Green, T. J., et al. : Provenance Semirings, In ACM PODS (2007).
3) Kolaitis, P. G. : Schema Mappings, Data Exchange, and Metadata Management, Invited talk in ACM PODS 2005.
4) Halevy, A. : Answering Queries Using Views : A Survey, The VLDB Journal 10 : pp.270-294 (2001).
5) Deutch, D. : Towards Big Data Provenance, Foundations of Data Management,Dagstuhl Perspectives Workshop 16151 (2016).

(2017年1月8日受付)

☆13 GAV, LAVと名付けたのも彼である．
☆14 計算機科学の分野で著名なセミナーで，ある課題について専門家が合宿形式で議論することで互いの交流をはかりつつ，その課題についての方向性を見出すことが目的のセミナー．日本でも国立情報学研究所が「湘南会議」として同様のものを開催している．

加藤弘之（正会員） kato@nii.ac.jp

国立情報学研究所コンテンツ科学研究系助教．XQueryの最適化，MapReduceの最適化など，データベースプログラミング言語の分野を中心に研究に従事．博士（工学）．