

情報源記述を用いた空間情報統合モデルの提案

柳 奇亨[†] 石川 佳治* 北川 博之*

[†]筑波大学 理工学研究科

* 筑波大学 電子・情報工学系

{nonl,ishikawa,kitagawa}@is.tsukuba.ac.jp

概要

GPSや携帯端末の発達により、携帯端末を保持したユーザや自動車などの移動オブジェクトの位置情報を得ることが容易となってきた。一方、オンライン、特にWeb上で利用可能な情報源の数も日々増えているが、これらの中にはある特定の地域に関する情報を提供する情報源や、提供する各情報に位置情報も付与して提供するような情報源が存在する。GPSにより取得した位置情報をもとに各時点のユーザの位置に応じた情報提供を行うような情報提供サービスはすでに存在するが、このようなサービスは異種の複数の情報源の存在を考慮しているわけではない。よって本稿では、多種多様な情報源を統合し、位置情報に応じた情報を提供するための枠組みを提案する。本アプローチの特徴は、各情報源が提供する情報の内容や各情報源の間合せ能力を宣言的にメタ情報として記述する情報源記述のアプローチを採用し、情報を統合するメディアータにおける間合せプラン作成に利用するという点である。本稿では特に、Open GISにおける地理データ型や地理演算の規格を考慮して空間情報統合について議論する。

キーワード：異種情報源の統合、地理データベース、情報源記述、位置情報、Open GIS、間合せ処理

A Proposal of Spatial Information Integration Model Based on Source Description Framework

Gihyong Ryu[†] Yoshiharu Ishikawa* Hiroyuki Kitagawa*

[†]Master's Program in Sciences and Engineering, University of Tsukuba

*Institute of Information Sciences and Electronics, University of Tsukuba

Abstract

Rapid development of information technology such as mobile terminals and GPS systems enabled information services that provide location-oriented information based on users' positions. In this paper, we propose an information integration framework to incorporate spatial information sources and provide appropriate information to mobile users. First, we provide a source description method to represent the content and the query capability of each spatial information source. Then we describe a query processing scheme that finds a combination of information sources to satisfy a given query then evaluates it efficiently. In particular, we focus on spatial data types and spatial operations proposed by Open GIS Consortium to provide a general information integration framework. **keywords:** heterogeneous information integration, geographic databases, source description, location-oriented information, Open GIS, query processing

1 はじめに

GPS, 通信技術, 携帯端末の発達により, 携帯端末を保持したユーザや自動車などの移動オブジェクトの位置情報を得ることが容易となってきている。一方, インターネットの発達により, 利用可能なネットワーク上の情報源の数が爆発的に増えてきているが, これらの中にはある特定の地域に関する情報を提供する情報源や, 提供する各情報に位置情報も付与して提供するような情報源が存在する。本稿ではこれらの情報源を総称して空間情報源 (spatial information source) と呼ぶことにする。多種多様な空間情報源が今後数多く出現してくることが考えられるため, それらの統合利用技術の開発が求められている。

GPS や PHS の機能により取得された位置情報をもとに各時点のユーザの位置に応じた情報提供を行うような情報提供サービスはすでに存在するが, このようなサービスは異種の複数の空間情報源の存在やそれらの統合処理を考慮しているわけではない。これに対し本稿では, ささまざまな情報源を統合し, 利用者の位置に応じた適切な情報を提供するための枠組みを提案する。まずはじめに, 指定された位置条件に基づいて情報を提供する情報源や, ある特定の地域に限定して情報を提供するような空間情報源について, それらが提供する情報の内容や, それらの問合せ能力を記述するための情報源記述の枠組みを提案する。続いて, 与えられた問合せに対し, 適切な情報源の組合せを選択し問合せを処理するための問合せ処理方式について述べる。問合せ処理において特に着目するのは, 距離に基づくフィルタリング処理など, 地理情報を利用する際にしばしば求められるような問合せの処理である。

本稿の構成は以下になる。まず, 2 節では背景と関連研究について述べる。3 節では, 情報源記述を行うための統合データモデルを導入し, 4 節では, 問合せと情報源記述の例を挙げ, 5 節では, 与えられた問合せが, 情報源記述に基づいて各情報源への問合せ要求にどのように変換され, また, どのように統合処理が行われるかを例を用いて示す。最後に 6 節では, まとめと今後の課題について述べる。

2 背景と関連研究

2.1 異種情報源の統合

リレーショナルデータベース, Web サーバなど, ネットワーク上には多種多様な情報源が存在する。異種の情報源を統合し, ユーザの要求に応じて適切な情報を提供するためには, 情報源の異種性を吸収し, 与えられた問合せに対して適切な情報源を選択し, 問合せの発行と結果の統合を行う機構が必要となる。このために広く用いられているアプローチは, 与えられた問合せに対して情報源を選択し問合せ処理を行うメディアータ (mediator) と, 各情報源をメディアータの処理機能に応じて抽象化するラッパー (wrapper) からなる情報統合のアプローチであり, これはしばしばラッパー/メディアータモデル (wrapper/mediator model) と呼ばれる [KMM00]。本研究でもこのアプローチを採用する。

メディアータにおける情報統合のための問合せ処理を実現するためには, 各情報源が有する情報を適切に表現するための情

報源を記述する枠組みが必要となる。このような枠組みには, 多様な情報源を表現するための柔軟性や, 効率的な問合せ処理プランを作成できるための簡潔かつ十分な表現能力が求められる。このような情報源に関するメタ情報を表現するためのアプローチの 1 つとして, Information Manifold [LRO96] などとられた情報源記述 (source description) のアプローチがある。このアプローチでは, まず対象となる異種情報源の集まりに対し大域的なスキーマを定義する。ユーザからメディアータに与えられる問合せは, この大域的なスキーマに対し記述されることになる。情報源記述には, 各情報源がこの大域的なスキーマのどの部分に相当するデータを提供するかという情報源の内容の記述, また, 情報源に対してどのような引数を与え, どのような結果を得ることができるかといった問合せ能力の記述などが含まれる。各情報源に対する情報源記述が与えられると, メディアータは問合せに対してどの情報源を組み合わせてよいか, また, どのように問合せ処理を行えばよいかといった問合せプラン作成を行うことができることになる。

2.2 地理情報の統合利用

GPS による位置情報取得機能, 携帯端末などの移動体通信技術, 地図の電子化などの各種技術の発展により, 地理情報を統合し提供するための基盤が着々と整備されつつある。たとえば, Open GIS コンソーシアムにおける活動 [OGC] や, XML に基づく地理情報の記述を与えるプロトコルである G-XML [GXML] 規格, XML による位置情報提供のための記述言語 POIX [POIX] に関する検討, 道路情報提供のための XML に基づく言語 RWML [RWML] への取り組みなどがあげられる。

このような基盤技術の発展や標準化の進展を背景として, モバイル環境においてユーザの位置に応じた情報提供を行うための研究が行われている [大石 00, 寺田 00, 倉林 00, 森下 00]。ActiveGIS [寺田 00] では, アクティブデータベースの概念を移動体環境に適用し, ECA ルールによる位置依存の処理記述の機能を提供している。これにより, たとえば「現在地の近辺に本屋があるときに, 探している本がその本屋にあるかどうかを本屋のデータベースから検索して結果を表示する」といった処理要求が記述・実行できる。[倉林 00] でも同様に, ECA ルールに基づく位置依存の処理記述方式について提案している。

本研究はこれらの研究と同様, モバイル環境における位置に応じた情報の提供のための技術に焦点を当てているが, 特に, 情報源の異種性に着目していることが特徴であり, メディアータにおいて有用な空間情報源をどのようにして選び出すか, また, 選び出した空間情報源にどのように問合せを発行するかという問題を具体的に考慮している点が特徴である。2.1 節で述べたように, 異種分散情報の統合では情報源が提供する情報の内容や情報源の問合せ能力がそれぞれ異なるため, 一種のメタ情報である情報源記述に基づいて情報統合処理を実現する。

3 統合データモデル

空間情報統合を行うため, 本稿ではリレーショナルデータベースに空間データ型を導入した統合データモデルを採用する。

以下では、まず統合要求例をもとに基礎的な概念を説明し、その後で空間データ型などの説明を行う。

3.1 大域的スキーマと統合要求の例

本研究では、Information Manifold [LRO96] などで採用された情報源記述のアプローチに基づいて情報統合を行う。情報源を統合するために、まず大域的なスキーマを設定する。大域的なスキーマの構築はメディアエータに対する統合要求に基づいて行われ、一方、各情報源の情報を大域的なスキーマにマッピングする処理は、各情報源上に構築されたラッパーにより行われる。

ここでは簡単な例として、レストランに関する各種情報源を統合して、ユーザが現在いる場所の近くにあり、評判のよい店を探すことを考える。レストランの情報は図1のような大域的スキーマで表現されるものとする。

```

1 relation Restaurant { // レストラン情報
2   name string;       // レストラン名
3   category string;   // レストランの種類
4   address string;    // レストランの住所
5   position Point;    // 所在地の座標
6 };
7
8 relation Scoring {   // レストランの評価
9   name string;       // レストラン名
10  score real;        // 評価値
11 };

```

図 1: 統合スキーマの例

Restaurant リレーションはレストランに関する情報を保持しており、たとえば name 属性はレストラン名、category 属性はレストランの種類別 (例: イタリアン, 和食) を表している。レストラン所在地の座標は Point 型の position 属性に保持されるものとする。一方、Scoring リレーションはレストランへの評価の情報を保持するリレーションであり、レストラン名と評価値を属性とする。

図1の大域的スキーマに対してユーザの現在の位置から1000メートル以内で、評価値が2.5以上のレストランを近い順に20件探し出し、その名前と住所をリストするような問合せは、SQL風に表現すると図2のようになる。ここで mypos は、ユーザの現在の位置を表す Point 型の値であるものとする。通常のSQLの機能に加え、ここでは点の間の距離を求める述語 Distance や、[CK97] で提案されている STOP AFTER 句を用いて要求を記述している。

3.2 空間データ型と空間演算

本研究では、空間情報を扱うため、通常のリレーショナルデータモデルに、Point 型などの空間データ型 (spatial data type) を導入したデータモデルを情報源のデータ構造の記述のために採用する。このようなデータモデルの採用は、単純・明

```

SELECT r.name, r.address
FROM Restaurant as r, Scoring as s
WHERE r.name = s.name, s.score >= 2.5,
      Distance(r.position, mypos) <= 1000
ORDER BY Distance(r.position, mypos)
STOP AFTER 20

```

図 2: 問合せの例

快かつ空間情報の表現能力を有するデータモデルが求められるという本研究の要求に基づくものである。空間データ型、空間演算については、OpenGIS [OGC] により仕様策定されている空間データの規格に基づいて議論を行う。特に、ODBC API を通じて空間情報にアクセスするための SQL に関する規格案 [OGC99] を参考にする。

まず、空間データ型について述べる。OpenGIS ではトップレベルのクラスとして Geometry クラスを定義している。その直下のサブクラスとしては、Point, Curve, Surface, GeometryCollection が存在する。Point クラスは点データ、Curve は点の集まりとして定義される曲線、Surface は点の集まりとして定義される面、GeometryCollection は Geometry オブジェクトのコレクションに対応している。

ただし本稿では、単純化のため使用可能な空間データ型を限定する。点データは Point クラスをそのまま利用するが、曲線に関しては点の集まりとしての Curve クラスではなく、そのサブクラスで曲線を線分の集まりで近似した OpenGIS の LineString クラスを対象として扱う。また、面に関しても、点の集まりとして定義されている Surface ではなく、そのサブクラスであり、線分の集まりにより定義される Polygon クラスを考慮の対象とする。また、MultiSurface, MultiCurve, MultiPoint のサブクラスを有する GeometryCollection クラスについては、議論の単純化のため今回の議論の対象からは外す。よって、以下で対象とする空間データ型は Point, LineString, Polygon の3種類となる。

問合せにおいて利用可能な空間的関連を表す述語としては、OpenGIS の SQL に関する仕様をもとに、以下のものを考慮する。ここで、 g_1, g_2 はすべて Point, LineString, Polygon のいずれかのクラスに属する Geometry オブジェクトを表す¹。

- $equals(g_1, g_2)$: g_1 と g_2 の領域が等しいとき真となる。
- $disjoint(g_1, g_2)$: g_1 と g_2 に交わりがないとき真となる。
- $intersects(g_1, g_2)$: g_1 と g_2 に交わりがあるとき真となる。 $intersects(g_1, g_2) \Leftrightarrow \neg disjoint(g_1, g_2)$ が成り立つ。
- $touches(g_1, g_2)$: g_1 と g_2 が接する。なお、 g_1, g_2 がともに Point オブジェクトである場合には適用できない。
- $crosses(g_1, g_2)$: g_1 と g_2 が交わりを持つとき真となる。 $intersects$ の場合とは異なり、この述語は (LineString, LineString), (LineString, Polygon) という組合せの場合にのみ適用でき、前者は2つの LineString オブジェク

¹本稿では GeometryCollection クラスのオブジェクトを考慮していないため、ここに示した述語のセマンティクスは OpenGIS の仕様よりも簡略化されたものとなっている。

トが 1 回以上点による交わりを有する場合、後者は LineString オブジェクトと Polygon オブジェクトが点または線の交わりを有する場合に相当する。

- $within(g_1, g_2)$: g_1 が g_2 に含まれるとき真となる。なお、 $equals(g_1, g_2)$ の場合には $within(g_1, g_2)$ は偽となる。
- $contains(g_1, g_2)$: g_1 が g_2 を含むとき真となる。これは $within(g_2, g_1)$ と等価である。
- $overlaps(g_1, g_2)$: g_1 と g_2 にオーバーラップがあるとき真となる。これは g_1, g_2 が (LineString, LineString), (Polygon, Polygon) の場合に定義され、前者は 2 つの LineString オブジェクトが一部線分を共有するとき、後者は 2 つの Polygon の交わりが Polygon となる場合 (つまり点や線ではない場合) に相当する。なお、 $equals(g_1, g_2)$, $within(g_1, g_2)$, $contains(g_1, g_2)$ のいずれかが成り立つ場合には $overlaps(g_1, g_2)$ は偽となる。

OpenGIS の仕様では、この他にパターン行列を用いて空間的な関連を判定する *Relate* 述語があるが、これに関しては今後の課題とし以下では考慮しない。

空間的関連に加え、以下のような空間オブジェクトに対する関数も利用する。

- $envelope(g)$: g の最小包含矩形 (Minimum Bounding Box; MBB) であるような Polygon オブジェクトを返す。
- $distance(g_1, g_2)$: g_1 と g_2 の最小距離を返す。
- $isempty(g)$: g が空である場合に真となる。
- $intersection(g_1, g_2)$: g_1, g_2 の交わった領域を返す。
- $union(g_1, g_2)$: g_1, g_2 の領域の和を返す。

ただし、以降の議論では、問合せ中では $distance$ 関数のみが利用可能であるとし、他の関数は問合せ処理においてのみ使用する。

4 問合せと情報源記述の例

以下では、前節で述べたような空間情報統合要求に対し、特に以下の 2 つの問題に焦点を当てて議論を行う。

- さまざまな情報源のうち、与えられた問合せに対し有用な情報を有している情報源をどのように特定するか。
- 有用な情報を保持していると判断された情報源から必要な情報を取り出すための問合せをどのように構築するか。

Information Manifold [LRO96] などの研究では、各情報源の入出力属性が必ずしもすべて問合せに利用できるとは限らないという状況における問合せプラン作成について議論していた。これに対し本研究では、簡単のため、各情報源中のリレーション中の属性へのアクセスには特に制限がないことを想定する。一方、情報源が処理可能な空間演算の種類には制限があることを想定する。また、図 2 では ORDER BY や STOP AFTER などといった順序付け機能や top- N 問合せ能力を想定しているが、これらについても今回の議論の内容からは外すことにする。

4.1 問合せの記述

ここでは、論理積問合せ (conjunctive query; 連言問合せ) の形式で問合せを記述することにする。ただし、空間述語 (例: $contains$) や空間関数 $distance$ 、および不等号による比較を記述可能としているため、通常のリレーショナルモデルにおける論理積問合せを拡張した形式を用いる。図 2 に示した統合要求の問合せを記述した例は以下ようになる。評価の高いレストランである地点 p から 1000 メートル以内にあるものを列挙する問合せであり、前述のとおり、距離の近い順に 20 件のみを検索するという点については省いている。

$$ans(n, a) \leftarrow Restaurant(n, c, a, p), Scoring(m, s), n = m, s \geq 2.5, distance(p, p) \leq 1000$$

ここで、 p は Point クラスの変数、 p は現在位置を表す Point クラスの定数オブジェクトを表す。以降では、このように変数と定数をフォントの字体で区別する。

問合せの一般形は以下ようになる。

$$ans(u) \leftarrow \mathcal{R}_1(u_1), \dots, \mathcal{R}_n(u_n), c_1, \dots, c_m \quad (1)$$

$\mathcal{R}_1, \dots, \mathcal{R}_n$ ($n \geq 1$) は大域リレーション、 u, u_1, \dots, u_n はそれぞれ変数の並びであり、 c_1, \dots, c_m ($m \geq 0$) はそれぞれ条件である。条件としては、

- $x \theta y$ または $x \theta c$ の形式の条件: ただし、 x, y は単純値属性に束縛されている変数で、 c は単純値をとる定数、 $\theta \in \{=, \neq, <, >, \leq, \geq\}$ は算術比較演算である。
- 空間述語に基づく条件: g, g_1, g_2 を Geometry クラスの値をとる属性に束縛されている変数、 g を定数の Geometry オブジェクトとしたとき、たとえば $contains(g_1, g_2)$, $contains(g, g)$, $contains(g, g)$ がこの例となる。
- $distance$ 関数に基づく条件: g, g_1, g_2 を Geometry クラスの値をとる属性に束縛されている変数、 g を定数の Geometry オブジェクト、 d を実数値の変数、 d を実数値の定数とする。 $\theta \in \{=, \neq, <, >, \leq, \geq\}$ を算術比較演算とすると、 $distance(g_1, g_2) \theta d$, $distance(g_1, g_2) \theta d$, $distance(g, g) \theta d$, $distance(g, g) \theta d$ の条件指定を許す。

各条件に現れる各変数は定数か大域リレーションのある属性に束縛されているものとする。なお、 $distance(g_1, g_2) \theta d$, $distance(g, g) \theta d$ という形式の条件については、 p, q が上記の条件を満たせば d は自由変数であってもよいものとする。

4.2 空間的な包含制約を伴う条件

この節では、以降の議論のため、空間的な包含制約を伴う条件の概念を導入する。例として、

$$ans(n) \leftarrow Restaurant(n, c, a, p), contains(g, p)$$

という問合せを考える。この問合せは、その位置が定数領域 g に含まれるようなレストランの名前をすべて求めるものである。この場合、変数 p がとることができる位置は定数 g の内部に制限されている。このようなとき、 $contains(g, p)$ を

空間的な包含制約を伴う条件と呼び、 p を領域制限された変数という。contains に関していえば、引数の両者が変数である場合 ($\text{contains}(g_1, g_2)$) や、包含する側が変数である場合 ($\text{contains}(g, p)$) の場合には該当しないことに注意する。本稿で導入した空間述語について、空間的な包含制約を伴う条件となるのは以下の3つの場合である。

- $\text{equals}(g, g)$ (および $\text{equals}(g, g)$)
- $\text{within}(g, g)$
- $\text{contains}(g, g)$

ただし $\text{equals}(g, g)$ については、変数 g を定数 g と置き換えることで変数 g を消去できるため以下では考慮しない。また、 $\text{within}(g, g)$ は $\text{contains}(g, g)$ と等価であるため、以降ではこのような空間述語としては contains のみを考慮の対象とする。空間的な包含制約を伴う条件として、距離関数を用いた条件も考慮の対象とする。たとえば、

$$\text{ans}(n) \leftarrow \text{Restaurant}(n, c, a, p), \text{distance}(p, p) \leq 1000$$

といった問合せがこれに相当する。具体的には、関数 distance に関する条件が空間的な包含制約を伴う述語となるのは、

$$\text{distance}(g, g) \theta d \quad (2)$$

という場合である。ここで、 g は変数、 g は定数の空間オブジェクト、 d は実定数、 θ は $<, \leq$ のいずれかである。

4.3 情報源記述の例

ここで情報源記述 (source description) について述べる。情報源記述とは、各情報源に対して記述されるもので、以下の2つから構成される。

- 内容記述 (contents description): その情報源がどのような内容を保持しているかを表す。
- 問合せ能力記述 (capability description): その情報源がどのような問合せ処理能力を有しているかを表す。

情報源記述の例を図3に示す。ここでは大域的なリレーションとして、先にスキーマを示した Restaurant , Scoring が存在すると考えている。情報源 A, B, C は Restaurant に対応し、情報源 D は Scoring に対応する情報を有している。大域的なリレーション \mathcal{R} に対応する情報源の集合を $\text{sources}(\mathcal{R})$ で表すことにする。

4.3.1 内容記述

記述の各項目について説明する。contents の項目では、その情報源が保有する情報が大域的なスキーマのどのような部分にあたるかを記述する。情報源 A を例にとって説明する。情報源 A は、領域 r の範囲内のレストラン情報を提供しているとする。つまり、大域的スキーマ上の Restaurant リレーション (実際には仮想的にしか存在しない) に対する問合せ

$$\text{ans}(n, c, a, p) \leftarrow \text{Restaurant}(n, c, a, p), \text{contains}(r, p)$$

<p>Source A: 地域限定のレストラン情報。領域 r 内の情報を提供。レストラン名、住所による検索が可能。</p> <p>contents: $S_A(n, c, a, p) \subseteq \text{Restaurant}(n, c, a, p), \text{contains}(r, p)$</p> <p>filters: $\langle n: \text{string} \rangle \Rightarrow n = n, \langle a: \text{string} \rangle \Rightarrow a = a$</p>
<p>Source B: 問合せ地点と距離の上限を指定して問合せが行えるレストラン情報源。カテゴリによる検索も可能。</p> <p>contents: $S_B(n, c, a, p) \subseteq \text{Restaurant}(n, c, a, p)$</p> <p>filters: $\langle p: \text{Point}, d: \text{real} \rangle \Rightarrow \text{distance}(p, p) \leq d, \langle c: \text{string} \rangle \Rightarrow c = c$</p>
<p>Source C: 名前により検索可能なレストラン情報源。Polygon の領域を指定すると、その内部に限定してレストラン情報を出力する。</p> <p>contents: $S_C(n, c, a, p) \subseteq \text{Restaurant}(n, c, a, p)$</p> <p>filters: $\langle n: \text{string} \rangle: n = n, \langle g: \text{Polygon} \rangle \Rightarrow \text{contains}(g, p)$</p>
<p>Source D: レストラン評価の情報源。名前、スコアによる検索が可能。</p> <p>contents: $S_D(m, s) \subseteq \text{Scoring}(m, s)$</p> <p>filters: $\langle n: \text{string} \rangle \Rightarrow n = n, \langle s: \text{real} \rangle \Rightarrow s \theta s (\theta \in \{=, \leq, \geq\})$</p>

図3: 情報源記述の例

の結果であるタブルの集合 $\text{ans}(n, c, a, p)$ について、

$$S_A(n, c, a, p) \subseteq \text{ans}(n, c, a, p)$$

という関係が成立する。このような意味で、情報源 A の contents の項目には

$$S_A(n, c, a, p) \subseteq \text{Restaurant}(n, c, a, p), \text{contains}(r, p)$$

と記述する。

contents の項目の記述形式の一般形を以下のように定める。

$$\text{contents}: S(u) \subseteq \mathcal{R}(u), c_1, \dots, c_n \quad (3)$$

u は変数の並びであり、 \mathcal{R} は大域リレーション名、 c_1, \dots, c_n ($n \geq 0$) は0個以上の条件の並びである。本稿では、ここに記述可能な条件として、以下の2種類の条件を許す。

- $x \theta c$ の形式の条件: ただし x は \mathcal{R} の単純値属性に束縛された変数であり、 c は単純値の定数である。
- 空間的な包含制約を伴う条件: すなわち、 $\text{contains}(g, g)$ あるいは $\text{distance}(g, g) \theta d$ ($\theta \in \{<, \leq\}$) の形式の条件で、 g は \mathcal{R} の Geometry クラスの属性に束縛されている変数である。

これにより、たとえば、領域 g に含まれ、しかも点 p から半径 5000 メートル以内にあるレストランを保持する情報源の内容記述は

$$S(n, c, a, p) \subseteq \text{Restaurant}(n, c, a, p), \text{contains}(g, p), \text{distance}(p, p) \leq 5000$$

となり、イタリアンレストランの情報を提供する情報源の内容記述は

$$S(n, c, a, p) \subseteq \text{Restaurant}(n, c, a, p), c = \text{"Italian"}$$

となる。

4.3.2 問合せ能力記述

filters の項目で、問合せにおいて指定可能な条件式のパターンを列挙する。 $pat \Rightarrow out$ という形式において、 pat は必須である入力引数の組 (入力パターン)、 out は入力パターン pat が与えられたときに適用される条件式を示している。情報源 A についていえば、引数 a に "Shibuya" という値を代入すれば、

$$ans(n, c, a, p) \leftarrow S_A(n, c, a, p), a = \text{"Shibuya"}$$

という問合せの結果が得られるということになる。複数の条件が適用される場合には、論理積 (conjunction) の形で適用されるものとする。

異種分散情報統合に関する研究では、入力として必ず指定しなければならない条件があったり、出力として大域スキーマの一部の属性しか得られないといった、さまざまな制約が存在する場合の問合せ処理方式が研究開発されている。しかし、ここでは簡単化のため、**filters** に記述されたパターンは 0 個以上任意に論理積の形で組み合わせて利用できるものとし、まったく入力条件を指定しない場合には全数検索により情報の抽出が可能であるものとする。問合せ結果の出力についても、一部の属性値が入手できないといった制約はおかず、指定した条件に合致するタプルがすべて情報源から入手できることを想定する。

5 問合せ処理

本節では、ユーザからの問合せがどのようなステップを経て処理されるかを述べ、本研究の中心となる情報源の選択問題と、選択された情報源にどのような問合せを発行すればよいかという問題に対する本研究のアプローチについて述べる。

5.1 問合せ処理の概要

ユーザから寄せられた問合せを処理するため、メタデータは問合せプランを作成するが、これは大まかに以下のステップからなる。

1. まず、大域スキーマに基づいて問合せが正当なものをチェックする。続いて、冗長な変数の除去、定数式の定数への変換を行う。後者の例としては、たとえば定数の点 p, q について、 $distance(p, q) \leq d$ という式を、 p, q の実際の距離をもとに $100 \leq d$ のような形式に変換する処理が含まれる。
2. 次に、情報源の選択を行う。問合せ中に指定された各条件に基づいて、各情報源が問合せ条件を満たす可能性があるかどうかを判定し、満たす可能性がないものはあらかじめ排除することで問合せ処理の効率化を図る。そのために、情報源の内容記述を利用する。

3. 第 3 のステップでは、各情報源の問合せ能力記述に基づいて、問合せ条件を情報源にプッシュできるかどうかの判定を行う。より多くの条件を情報源にプッシュできれば、一般にはより効率的な問合せ処理が行えると考えられるため、これは一種の最適化処理となる。

4. 最後のステップは、統合問合せプランの作成である。情報源からの問合せ結果を統合し効率的にメタデータ内部で評価するためのプラン作成を行う。この部分については本稿では省略する。

5.2 情報源の選択

ここでは、ユーザからの問合せ中で指定されたそれぞれの条件に対応する情報源の内容記述と比較検討することにより、問合せ処理に有用な情報源の選択を行う。

5.2.1 空間的な包含制約を伴う条件について

空間的な包含制約を伴う条件を各情報源の内容記述と照合することにより、与えられた問合せに対し情報源が有用な情報を有している可能性があるかどうかの判断をすることができる。これにより、有用でないことが明らかな情報源を問合せ対象から排除することができる。

例 1 大域的スキーマ上の問合せとして

$$ans(n) \leftarrow \text{Restaurant}(n, c, a, p), dist(p, p) \leq 1000$$

が与えられたとする。 $dist(p, p) \leq 1000$ は大域リレーション Restaurant の空間属性に束縛された変数 p を一つだけ含んでいる。ここで、 $A \in sources(\text{Restaurant})$ であり、情報源 A の情報源記述が

$$S_A(n, c, a, p) \subseteq \text{Restaurant}(n, c, a, p), c = \text{"Italian"}, contains(r, p)$$

であつたとする。このとき、情報源 A が問合せを満たす可能性があるのは、

$$\exists p(contains(r, p) \wedge dist(p, p) \leq 1000)$$

が真となる場合である。これは、領域 r と、 p を中心とする半径 1000 の円 ($circle(p, 1000)$ と表記する) の間に交わりがある場合、つまり、 $intersects(r, circle(p, 1000)) = true$ の場合に相当する。 ■

以上の考察に基づいてよりフォーマルに議論する。与えられた問合せ

$$ans(u) \leftarrow \mathcal{R}_1, \dots, \mathcal{R}_n, c_1, \dots, c_m \quad (4)$$

に含まれているある大域リレーション \mathcal{R}_i ($1 \leq i \leq n$) に着目する。ここで、 \mathcal{R}_i に対応する情報源の集合を、 $sources(\mathcal{R}_i) = \{R_{i1}, \dots, R_{il}\}$ と表記する。また、 R_{ij} ($1 \leq j \leq l$) の内容記述が

$$S_{R_{ij}}(v) \subseteq \mathcal{R}_i(v), e_1, \dots, e_s \quad (5)$$

であるとする。 \mathcal{R}_i のある空間属性 a について、 c_1, \dots, c_m のうち a に束縛されている変数 x を含んでおり、かつ、空間的な

包含制約を伴う条件であるもの²を $\tilde{c}_1, \dots, \tilde{c}_{m'}$ ($m' \leq m$) とする。また、 e_1, \dots, e_s のうち a に束縛されている変数を含んでいるものを $\tilde{e}_1, \dots, \tilde{e}_{s'}$ ($s' \leq s$) とする。ただし、 $\tilde{e}_1, \dots, \tilde{e}_{s'}$ において属性 a に束縛されている変数を x に変更しておく。このとき、

$$\exists x(\tilde{c}_1 \wedge \dots \wedge \tilde{c}_{m'} \wedge \tilde{e}_1 \wedge \dots \wedge \tilde{e}_{s'}) \quad (6)$$

という式が偽になる場合には、情報源 R_{ij} が問合せ中の条件 $\tilde{c}_1, \dots, \tilde{c}_{m'}$ を満たす可能性はないことになり、これらの条件に関しては情報源 R_{ij} を評価の対象から排除することが可能となる。この判定は、条件が表す領域の交わりを *intersection* 演算で求め、これが空であるかどうかを *isempty* 関数を用いて判定することで決定できる。このような判定処理は、Open GIS の SQL 規格案に準拠した SQL 処理系が存在すれば記述可能である。

ただし、任意の形状を有する Geometry オブジェクトの交わりを求めることは一般的にはコストが大きい。そこで、*envelope* 演算を利用してそれぞれの空間オブジェクトの MBB を求めその交わりを求めることで、問合せプラン作成時の処理コストを低減させることも可能である。ただし、その場合には、実際には有用な情報を保持していない情報源を探索してしまう可能性があるため、問合せプラン作成コストと実行時のコストのトレードオフが存在する。

単純値属性についても、同様の考え方で有用でない情報源を条件式の評価から排除することが可能である。これは、日本食レストランを求める

$$ans(n) \leftarrow Restaurant(n, c, a, p), c = \text{"Japanese"}$$

といった問合せに対し、イタリアンレストランの情報のみを有する情報源を処理対象から排除する場合に相当する。この処理については詳細は省略する。

5.2.2 空間的関連に基づく結合処理について

空間的関連に基づく結合条件を含んだ問合せに関して、内容記述を考慮することにより無駄な結合処理を避けることが可能となる。

例 2 バス停の名前 m と位置 l を保持している大域的リレーション $BusStop(m, l)$ があるとす。このとき、

$$ans(n, m) \leftarrow Restaurant(n, c, a, p), BusStop(m, l), distance(p, l) \leq 200$$

という問合せを考える。ここで、 $A \in sources(Restaurant)$, $B \in sources(BusStop)$ を満たす情報源 A, B の内容記述が

$$S_A(n, c, a, p) \subseteq Restaurant(n, c, a, p), contains(r, p) \\ S_B(m, l) \subseteq BusStop(m, l), contains(s, l)$$

であったとする。このとき、 $distance(r, s) \leq 200$ の場合には、 A, B の結合条件が満たされる場合があるため、問合せを発行する必要がある。しかし、 $distance(r, s) > 200$ の場合は条件が満たされる可能性がないため、処理を省略できる。■

²先の定義では空間的な包含制約を含む条件には含めていなかったが、 $distance(g, g) = d$ という形式の条件については $\tilde{c}_1, \dots, \tilde{c}_{m'}$ の中に含めても議論が成立する。

より具体的には次のようになる。問合せ

$$ans(u) \leftarrow \mathcal{R}, S, c_1, \dots, c_n \quad (7)$$

に対し、 $R_i \in sources(\mathcal{R})$, $S_j \in sources(S)$ の情報源記述がそれぞれ

$$S_{R_i}(v) \subseteq \mathcal{R}(v), e_1, \dots, e_m \quad (8)$$

$$S_{S_j}(w) \subseteq S(w), f_1, \dots, f_l \quad (9)$$

であったとする。 r, s をそれぞれ、 R_i, S_j の空間属性に束縛されている変数とする。このとき、 e_1, \dots, e_m 中で R_i のその空間属性に束縛されている条件を $\tilde{e}_1, \dots, \tilde{e}_{m'}$ ($m' \leq m$)、同様に S_j の対応する空間属性に束縛されている f_1, \dots, f_l の条件を $\tilde{f}_1, \dots, \tilde{f}_{l'}$ ($l' \leq l$) とする。 $\tilde{e}_1, \dots, \tilde{e}_{m'}$ に対応する各領域の和を *union* でとった結果を e とし、同様に $\tilde{f}_1, \dots, \tilde{f}_{l'}$ についても領域の和をとった結果を f とする。このとき、

- 問合せ中に *equals*(r, s) という条件が含まれていた場合、*intersects*(e, f) の評価結果が偽であれば、情報源 R_i と情報源 S_j を *equals*(r, s) で結合した結果は明らかに空となる。よって問合せ処理の中で考慮する必要はない。同様に考えて、*intersects*(r, s), *touches*(r, s), *crosses*(r, s), *within*(r, s), *contains*(r, s) についても *intersects*(e, f) による判定に帰着できる。
- *distance*(r, s) θ d ($\theta \in \{=, <, \leq\}$) という条件については、*distance*(e, f) $\leq d$ という条件の判定に帰着できる。

以上のようなアプローチにより、問合せ対象とする情報源を削減することが可能となる。

5.3 問合せ能力記述に基づく問合せ条件のプッシュ処理

与えられた問合せを効率的に処理するためには、各情報源が有するフィルタリング機能を有効に活用することが必要となる。この節では、特に空間属性に着目して、与えられた問合せ条件をどのように情報源へのフィルタリング条件に変換するかについて述べる。

基本的なアプローチとしては、与えられた条件をできるだけ情報源にプッシュすることになる。たとえば、

$$ans(n) \leftarrow Restaurant(n, c, a, p), contain(r, p)$$

の場合、情報源記述の問合せ能力指定に

$$\langle g: Geometry \rangle: contain(g, p)$$

のような、引数と述語が一致するものがあればそれを利用し、情報源に条件 *contain*(r, p) をプッシュする。*within* と *contains* のように互いに等価な演算に関する条件についてもこのような適用が可能である。

一方、完全にマッチするフィルタリング条件が存在しない場合には、求める結果を含むより緩和されたフィルタリング条件を探す。たとえば、*equals*(g, g) という条件に対しては、対象の情報源において直接 *equals* がサポートされていない場合

には $intersects(g, g)$ という条件が利用可能かを調べ、利用できる場合にはこの条件を情報源にプッシュする。ただし、この場合には、得られた結果が実際に $equals(g, g)$ を満たすかどうかをあらためて検証する必要がある。このような、条件をより緩める方針については、 $intersects$ 演算が有用である。 $equals, touches, crosses, within, contains, overlaps$ はすべて $intersects$ 演算により支援できる。距離関数を含んだ問合せ、たとえば

$$ans(n) \leftarrow Restaurant(n, c, a, p), dist(p, p) \leq 1000$$

についても、 p を中心とする半径 1000 の円と $Restaurant$ の位置 p の $intersects$ を求める処理に帰着できる。

以上述べた内容をまとめると、問合せプラン作成における問合せ条件の処理の方針は以下ようになる。

1. 情報源が処理可能な条件はできるだけ情報源にプッシュする: プッシュされた条件を情報源が利用するかどうかは、情報源上のラッパーの判断に任される。
2. その際、以下のような方針をとる。
 - (a) 着目している条件と等価なフィルタ条件が情報源において利用できる場合には、条件を情報源に直接プッシュする。
 - (b) 等価なフィルタリング条件は存在しないが、より緩いフィルタリングが利用できる場合は、より緩い条件に変換して情報源にプッシュする。この場合、情報源から得られた問合せ結果を検査し、条件を実際に満たしているかの検討を行う必要がある。

6 まとめと今後の課題

本稿では、地理情報を扱う異種情報源を統合し、ユーザの位置に応じた情報提供を行うための枠組みを提案した。その特徴は、各情報源が保有する情報やその問合せ能力を記述する情報源記述のアプローチを採用したことであり、情報源記述は問合せを発行すべき情報源の選択、各情報源に対する問合せの生成、問合せ結果の統合処理のプラン作成のために用いられる。

今後の課題であるが、情報源の選択方法、問合せ能力記述に基づく情報源への問合せ発行の方式、情報源への問合せした結果を統合するための問合せ処理方式などが課題として挙げられる。特に、今回述べた方式では、メディアータの問合せ処理において Open GIS において仕様化されている空間演算が自由に利用可能としているが、実際にはこれらの演算にも処理コストが存在するため、問合せプラン作成のコストと作成されたプランの良さのトレードオフを考慮したプラン作成手法が必要であると考えられる。また、具体的な応用として、冒頭で述べたような GPS や携帯機器を有して連続的に移動するユーザを考慮して、ユーザの近辺のデータを継続的に提示するための連続的問合せ (continual query) をどのように記述・処理するかについても見当する必要がある。

参考文献

- [CK97] M.J. Carey and D. Kossmann. Processing Top N and Bottom N queries. *IEEE Data Engineering Bulletin*, Vol. 20, No. 3, pp. 12–19, Sept. 1997.
- [GXML] G-XML ホームページ
<http://gisclh.dpc.or.jp/xml/contents/>
- [KMM00] H. Kitagawa, A. Morishima, and H. Mizuguchi. Integration of Heterogeneous Information Sources in InfoWeaver. In Y. Masunaga and S. Spaccapietra (eds.), *Advances in Multimedia and Databases for the Next Century—A Swiss/Japanese Perspective—*, pp. 124–137. World Scientific, 2000.
- [LRO96] A.Y. Levy, A. Rajaraman, and J.J. Ordille. Querying Heterogeneous Information Sources Using Source Descriptions. In *Proc. of VLDB*, pp. 251–262, Mumbai, India, Sept. 1996.
- [OGC] Open GIS Consortium Home Page
<http://www.opengis.org/>
- [OGC99] Open GIS Consortium, Inc. Open GIS Simple Features Specification for SQL, Revision 1.1, May 1999. OpenGIS Project Document 99-049.
- [POIX] POIX: Point Of Interest Exchange Language
<http://www.w3.org/TR/poix/>
- [RWML] 道路用 Web 記述言語 RWML の提案
<http://www2.ceri.go.jp/its-win/RWML.html>
- [寺田 00] 寺田努, 塚本昌彦, 西尾章治郎. アクティブデータベースを用いた地理情報システム. 情報処理学会論文誌, Vol. 41, No. 11, pp. 3103–3113, 2000 年 11 月.
- [森下 00] 森下健, 中尾恵, 垂水浩幸, 上林弥彦. 時空間限定オブジェクトシステム SpaceTag: プロトタイプシステムの設計と実装. 情報処理学会論文誌, Vol. 41, No. 10, pp. 2689–2698, 2000 年 10 月.
- [倉林 00] 倉林修一, 石橋直樹, 清水康. モバイル・コンピューティング環境におけるアクティブ型マルチデータベースの実現方式. 情報処理学会研究報告, Vol. 2000, No. 69, pp. 463–470, 2000 年 7 月. 2000-DBS-122-60.
- [大石 00] 大石晴夫, 上林悟, 石山正樹, 大窪政範, 桑木伸夫. XML と WIDL 技術を利用した位置情報提供システムの検討. 2000 年電子情報通信学会総合大会, 2000 年. D-9-29.