

分散環境における仮説生成による問合せ機能の拡張

萬上裕 黒田崇 横田一正

京都大学工学部 京都大学大学院工学研究科

〒606-01 京都市左京区吉田本町

{banjou,tkuroda,yokota}@kuis.kyoto-u.ac.jp

広域ネットワーク上に分散したさまざまな自律的な情報源を統合 / 融合することによってより有用な情報を得ることが必要とされている。このためには各情報源の異種性の解消、対象とする情報源の融合、情報の表現と問合せ処理のための高い表現能力をもった言語などが必要とされている。本稿では、そのための代表的なメディーエータシステムを取り上げ、その拡張を行なう。そのために、情報源の表現とそれらを統合するメディーエータの仕様記述を統一的な言語として、演繹オブジェクト指向データベース言語 *QUIXOTE* を拡張した *QUIK* (*QUIXOTE* in Kyoto) を提案し、包摂関係のマージ、仮説生成による情報源の探索、オブジェクトの同定などについて議論する。

Extensions of Query Processing in Mediator Systems

Yutaka Banjou Takashi Kuroda Kazumasa Yokota

Department of Information Science, Kyoto University

Yoshida-Honmachi, Sakyo, Kyoto 606-01, Japan

For advanced data-oriented applications in wide area networks, effective information is frequently obtained by integrating or fusing various autonomous information sources. There are many problems: how to resolve their heterogeneity, how to integrate target sources, how to represent information sources with a common protocol, and how to process queries. In this paper, we propose a new language, *QUIK*, as an extension of a deductive object-oriented database language, *QUIXOTE*, and extend typical mediator systems. We discuss various features of *QUIK*: integrating an exchange model and an mediator specification language, merging subsumption relations, searching alternative information sources by hypothesis generation, and identifying objects.

1 はじめに

近年コンピュータネットワーク上に存在する情報は加速度的に増加している。また、WWWなどを介して一般の利用者がその情報を扱うことが容易になった。ネットワークを介して利用可能なそれらデータベース等の情報源が持つデータ量が増加する一方で、複数の異なる情報源を統合的にアクセスする手法については未だ充分なもの存在しない。利用者は目的に応じた情報を格納する情報源を選択し、その情報源に対して必要な検索を行うという手法が一般的である。しかし、ネットワーク上に散在するファイルシステムやWWW情報、データベースなどの分散した情報源を統合的に利用可能とするようなシステムが実現されると、利用者はネットワーク上に存在する様々な情報に対してその情報がどこに存在するかを意識することなく統一的に利用することが可能となる。このことにより、利用者は目的に応じた情報をより容易に入手することが出来ると考えられる。

ネットワーク上に分散する情報源から得られるデータを利用者の求める形に合成するシステムを実現するためには、それらの情報源間の矛盾や情報の異種性が問題となる。このような問題をメディエータアーキテクチャを用いて解決するシステムにTSIMMIS [5]がある。TSIMMISはスキーマを持たない半構造化情報を統合するシステムであるが、これは共通プロトコルとして、オブジェクト識別子を持った複合オブジェクト程度しか定義できないという問題がある。

本稿では広域ネットワーク上に分散する情報源が持つ情報を統合するメディエータアーキテクチャを拡張することを考える。各情報源は統合環境内部ではカプセル化され、環境内で共通のプロトコルを持たせる。共通のプロトコルとして、演繹オブジェクト指向データベース言語／知識表現言語QUIXOTEを拡張したものを用い、環境内部での大域的な包摂関係の維持のための対話的な情報統合を提案する。また、メディエータ記述言語に仮説生成機能を導入することによる新たな情報源の探索機構を提案し、メディエータに対する問い合わせを拡張する。

本稿では、2節で従来のメディエータアーキテクチャにつ

いて簡単に概観し、3節でQUIXOTEを拡張した言語QUIKについて簡単に説明する。その後、4節で包摂関係のマージ、5節で仮説生成機構を用いた問い合わせの拡張について述べる。

2 メディエータアーキテクチャ

分散環境における情報源の統合に関連する研究として、メディエータシステムの説明を簡単に行ない、その後、このシステムにおける問題点について考察する。

2.1 TSIMMIS システムの機能

TSIMMISはメディエータアーキテクチャに基づく分散情報統合システムで、全体のアーキテクチャは図1のようになっている。ラッパー (Wrapper) は情報源のデータを共通モデルに変換する。メディエータ (Mediator) がラッパーからのオブジェクトを融合し、統合ビューを定義する。TSIMMISでは、データベースのみならずファイルシステムなども含めた異種分散情報源に対する統合アクセスを与える。

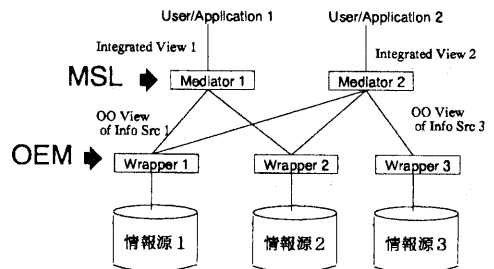


図 1: メディエータ アーキテクチャ

文献 [5] では、共通モデルとして OEM モデル (Object Exchange Model) を以下のように定義している。

```
<&rl,report,set,{&rln,&rla,&rlt,&rlr}>
<&rln,rn,string,'AB-123'>
<&rla,authors,set,{&rlal}>
<&rlal,author,string,'John Patriot'>
```

```
<&rlt,title,string,'UN Conspiracies'>
<&rlr,rel,set,{&r2}>
<&r2,report,set,{&r2n,&r2a,&r2t,&r1r}>
```

ここで第1引数の $\&rxx$ はオブジェクト識別子であり、第2引数は属性名、第3引数は属性値の型、第4引数は属性値である。OEM は情報源に対する一種のビューであり、半構造化データを含む情報源に対する共通モデルを定義している。

ところが演繹オブジェクト指向モデルの観点から考えれば、単一の情報源に閉じたビューだけでは以下の点で不十分である。

- 同一ヘッド / オブジェクトを持つルールをマージすることによってより多くの解を得ることができる。
- 各情報源が包摂関係を持っている場合は複数の情報源間での一貫性を保証する必要がある。

したがってこれを解決するために共通モデルを拡張し、それをメディアータに反映することを考える。そのため共通モデルとして *Quixote*[6, 7] を使用する (3節以降でこれを拡張する)。上の例は *Quixote* では以下のように書くことができる。

```
&rl/[report={&rln,&rla,&rlt,&rlr}]
&rln/[rn='AB-123']
&rla/[authors={&rla1}]
&rla1/[author='John Patriot']
&rlt/[title='UN Conspiracies']
&rlr/[rel={&r2}]
&r2/[report={&r2n,&r2a,&r2t,&r1r}]
```

あるいは

```
&rl/[%rn='AB-123',
    %authors=('John Patriot'),
    %title='UN Conspiracies',
    %rel={&r2}]
&r2/[%rn=...,%authors={...},%title=...,%rel=...]
```

さらに *Quixote* ではルールと包摂関係を記述できるが、モジュールについてはそれ自体が局所的であるため、メディア

ータに輸出することは考えにくいので、共通モデルとしての *Quixote* にはモジュール指定はないものとする。

2.2 共通モデルとその融合

メディアータは、複数の情報源のそれぞれの共通モデルを用いて、統合された一種のビューを定義する。文献 [5] のメディアータ仕様記述言語 MLS (Mediator Specification Language) はルールで以下のような表現を取っている。

```
<trep(RN) tr {<title T>}@m :-
    <report {<rn RN> <title T>}>@s1
<trep(RN) tr {<postscript T>}@m :-
    <report {<rn RN> <postscript T>}>@s2
```

ここでヘッドの “@m” はその前の情報がメディアータで得られることを示し、ボディ部の “@s1” “@s2” は対応する情報が情報源 s_1 , s_2 からそれぞれ得られることを示している。ここで考えなければならない問題点がある。

- 一階論理だけで、再帰関係を含む一般的なルールが扱えない。
- 使用する情報源の包摂関係を考えるための拡張の余地がない。
- 情報源の指定が固定で、その探索機能を持たせ得ない。
- メディアータも一種の (複合) 情報源であるが、ラッパーの共通モデルと互換性がない。
- メディアータ仕様記述の作成支援環境が考えられていない。

などである。

本稿では、情報源の共通モデルを *Quixote* としたが、上記のようにメディアータの仕様記述と共通モデル記述を共通の言語にすることによって、図 2 のような階層的なメディアータ記述が可能になる。階層化の利点はあるメディアータで見つからなかった情報を上位のメディアータを使うことによってより多くの情報源を保持することが可能になることである。*Quixote* 自体は情報源の記述や包摂関係のマージ機能を持っていないので、*Quixote* を拡張した新しい知識表現言語 QUIK (*Quixote* in Kyoto) を設計することにし

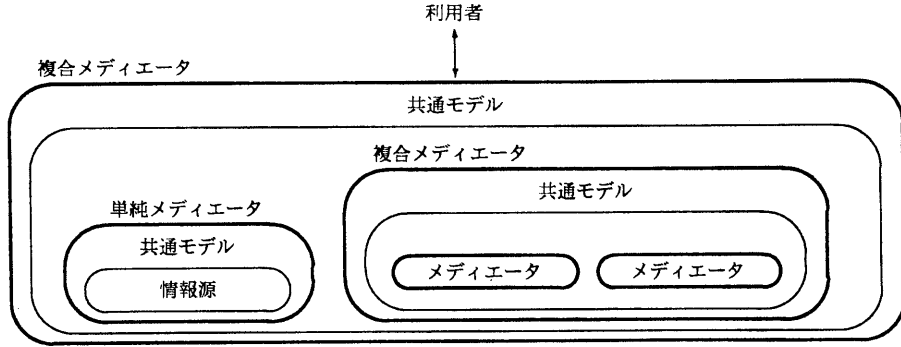


図 2: メディエータの階層構造

た。したがって情報源の共通モデルも QUIK での記述とする。また、これ以後は“情報源”という語はメディエータも含めた一般的な用語として用いる。

3 情報源／メディエータ記述言語 QUIK

WWW 上のファイルのような一定のスキーマを持たない情報源の情報を記述する必要があるために、共通モデルには高い柔軟性が要求される。たとえば、ファイル上の情報は表の形式で記述されていても、データによって欄の数が変わったり、欄が追加されたりする。共通モデルはこのようなスキーマを持たない半構造化された情報を表現できる必要がある。また、高度な知識表現を可能にするためには、単に複合オブジェクトを表現できるだけでなくルールや包摂関係を表現できた方がよい。

3.1 QUIK プログラム

QUIK プログラム P は、データディクショナリ / ディレクトリ (DD/D) D 、情報源関係 I 、包摂関係 S 、ルールの集合 R からなっている。つまり

$$P = (D, I, S, R)$$

の四つ組である。一つの情報源に対し、一つの QUIK プログラムが定義される。

S は概念の束である。すなわち、この情報源を形作る構成要素であり、それらの要素の間には包摂関係 (subsumption

relation) “ $\sqsubseteq$ ” が付けられている。 $a \sqsubseteq b$ は a が b より詳細化されていることを示すが、直感的には “is-a 関係” に相当する概念間の半順序関係である。半順序集合から束への変換はよく知られている [4] ので、 S は一般性を損なうことなく束であることを仮定する。 $QUIKOTE$ と異なっているのは同値関係を持っていることである。これは後述するメディエータ内の大域的な束生成の際に生成されたものと、情報源間のオブジェクト同定の結果導入されるものであるが、これは 6 節で議論する。

DD/D は情報源の一種のメタデータであり、環境内で定義されているデータを保存している。DD/D D は、情報源識別子、オブジェクト識別子、属性名から以下のように構成されている。

$$D = (ID, \{ID_i, \{O_{ij}, \{L_{ijk}\}\}\})$$

ID は自己識別子、 ID_i は自己を含む情報源識別子、 O_{ij} は ID_i に含まれるオブジェクト識別子、 L_{ijk} はオブジェクト O_{ij} が持つ属性名である。 D はオブジェクト識別子、属性名からも参照可能であるとする。

各ルールは以下の形式をしている。

$$a \leftarrow s_1 : a_1, \dots, s_n : a_n | C$$

s_i は情報源識別子で、 a, a_i は $QUIKOTE$ と同じ属性項、 C は情報源識別子を含む包摂関係を集めた制約である。 $n=0$ のときそのルールはファクトと呼ばれる。情報源識別子を $QUIKOTE$ のモジュール識別子と同様に構造化することに

よって、ひとつの情報源に複数の共通モデルが記述されていたときそれらを区別することができる。

情報源の手続きの意味は *QUICKOTE* におけるモジュールの外部参照と同じである。情報源識別子が省略されると自情報源を探索する。情報源関係は、この情報源 / メディエータがどの情報源を参照するか、どの情報源のルールをダウンロードするかを指定する。

参照関係: $I_1 \subseteq_R I_2$

ダウンロード関係: $I_1 \subseteq_D I_2$

$I_1 \subseteq I_2$ で I_1 が I_2 を参照 / ダウンロードすることを表す。一般的にルール型言語では単なる外部参照か、ダウンロードかで意味が異なっているので 2 種類の情報源関係を導入した。例えば

s1: $p(X) \Leftarrow r(X, Y), p(Y)$
 $p(a)$
 s2: $p(b)$

では s1 と s2 をマージすることで p は終了条件が二つに増える。ダウンロードされる情報源の場合、ルール中で指定される情報源識別子は無視される。ルールでの情報源の外部参照は、ここで規定する参照関係にしたがっていなければならない。これは包摂関係の一貫性を保つためである。

メディエータはラッパーからの共通モデルオブジェクトを融合し、情報源に対する統合ビューを定義する。その際、複数の情報源からの冗長な情報や、情報源間の矛盾を取り除く。

3.2 ルールによるオブジェクトの融合

メディエータがどのように分散した情報源からの情報を融合するかを例を用いて示す。情報源 s1 が論文番号とタイトル、参照文献などの文献情報を持ち、情報源 s2 が論文番号、著者、出版年の情報を持つとして、メディエータでこれらの情報を融合する。ここで問題になるのが、情報源に定まったスキーマがないかも知れないことである。たとえば半構造化されている情報源 s2 では、オブジェクトによって出版年の属性が無い場合や、掲載ページという属性がある場合がある

とする。さらに半構造化された情報源ではメディエータ管理者の知らない間に属性が追加されたり削られていることがあり、それらのラッパーレベルでのデータ構造の変化にメディエータは自動的に対応できることが望ましい。そのために、ラベル変数を導入している。メディエータの記述は次のようになる。

論文[論文番号 = RN]/
 [タイトル = T,
 著者 = {A},
 LABEL = X] $\Leftarrow$
 s1: 論文[論文番号 = RN]/
 [タイトル = T],
 s2: 論文[論文番号 = RN]/
 [著者 = {A}, LABEL = X].

メディエータ記述は情報源からのオブジェクトをほかのオブジェクトを定義する規則からなる。規則には本体の各情報源で変数 (RN や T など大文字で表される) が充足されると頭部で対応する融合オブジェクトが生成される。

メディエータは問い合わせを受け取ると、メディエータ記述に従い各情報源に対して問い合わせ、変数を束縛していく。LABEL はラベル変数で、任意のラベルに対して束縛される。これにより、情報源の変化に対し、メディエータを書き換える必要がなくなる。

4 概念階層のマージ

複数の情報源に対し問い合わせを行うにあたり、問い合わせの一貫性を保つために、使用する情報源の包摂関係をあらかじめマージしておかなくてはならない。情報源 I が情報源 $I_1, I_2, \dots, I_n$ を参照しているとき、それぞれの包摂関係を $L, L_1, L_2, \dots, L_n$ とすると、 I の包摂関係は $L \cup L_1 \cup L_2 \cup \dots \cup L_n$ から得られる包摂関係である。一般的に束 L_1 と L_2 の合併 $L_1 \cup L_2$ は必ずしも束とはならない。この束の生成アルゴリズムはよく知られている [4, 2] が、実際には $\top \sqsubseteq \perp$ が得られる危険性があるので、QUICK ではメディエータ作成時の支援環境と位置づけ、ここでは

利用者の対話的構築によって可能としている [3]。この構築の過程で生じる同値関係が新たに DD/D に貯えられる。束 L_1 と L_2 から生成できる束を $\overline{L_1 \cup L_2}$ と書くことにする。 $L = (D, I, S, R)$, $L_i = (D_i, I_i, S_i, R_i)$ ($1 \leq i \leq n$)、そして

$$L' = \overline{L \cup L_1 \cup L_2 \cup \dots \cup L_n}$$

のとき、 L' を $I, I_1, L_2, \dots, I_n$ に共通の包摂関係として定義する。これによって関係するすべての情報源はメディアータによって定義されることによって、メディアータ内で一貫性をもった大域的な包摂関係をもつことになる。メディアータで生成されたこの包摂関係は関係する情報源に輸出され、各情報源での制約解消や属性継承は、対応するメディアータから呼び出されたときにはこの包摂制約に従うことになる。

5 仮説生成を用いた問い合わせ処理

仮説生成と条件付き問合せは部分情報をもった環境での問合せ処理で有効である [7]。仮説生成は *QUIXOTE* では制約論理型言語と異なり導出列を

$$(G_0, A_0, \emptyset) \Rightarrow (G_1, A_1, C_1) \Rightarrow \dots \Rightarrow (\emptyset, A_n, C_n)$$

の3つ組として考えた。 G_i は解決すべきオブジェクト項、 A_i は解決すべき制約、 C_i は解決された制約である。オブジェクトのゴールが充足されさえすれば、充足されない制約 A_n のみを仮説として生成し、解のひとつとして $(\emptyset, A_n, C_n)$ を生成してきた。これはオブジェクトの存在は仮説としては生成しないが、その属性は仮説とすることに対応していた。これはまた解の生成の爆発を防ぐことにも役立っていた。

分散環境ではあらかじめ指定された情報源にオブジェクトが存在しないとしても、他の情報源には存在するかもしれない。したがって上記の制限を緩和することが有効が多い。しかし一般に開放系の中で必要なオブジェクトを探索するのは事実上不可能なので、メディアータという閉鎖系の中でオブジェクトを探索することを考える。

QUIK プログラム (D, I, S, R) 、問合せ $?-G$ に対する失敗した導出列の最終ノードが $N_i = (G_i, A_i, C_i)$ であるとする。DD/D D が、オブジェクトに対して情報源識別子を返

すとする。 $G_i = G'_i \cup \{o@s_j\}$, $o@s_j \notin G'_i$ のとき、 D が o に対して $I_C = \{s_1, s_2, \dots, s_n\}$ を返すとすれば、 N_i から

$$N_{i_1} = (G'_i \cup \{o@s_1\}, A_i, C_i)$$

$$N_{i_2} = (G'_i \cup \{o@s_2\}, A_i, C_i)$$

⋮

$$N_{i_n} = (G'_i \cup \{o@s_n\}, A_i, C_i)$$

を生成することができる。さらにオブジェクトの存在が解消された後、属性に関する制約解消にも DD/D を用いることができる。情報源に対応する変数が実行時に束縛されていない場合、この DD/D によって情報源を見つける。

たとえば

$$\begin{aligned} m : & (m, \{(s1, \{q, r\}), (s2, \{r, s\}), \text{DD/D} \\ & (m, \{p, \dots\}), \{\}\}) \\ \{m \sqsubseteq_{s1} s1, m \sqsubseteq_{s1} s2\} & \quad \text{情報源関係} \\ p \leftarrow s1 : q, s2 : r. & \\ s1 : q ; r. & \\ s2 : r \leftarrow s. & \end{aligned}$$

のとき、メディアータ m に対する問合せ $?-p$ はこのままでは失敗するが、DD/D により、 r が $s1$ にあることを知り、 $s1 : r$ の代替サブゴールにより、候補解が得られる。

1節で述べたように、ネットワーク環境における情報源は自律性が高く、不安定な場合を考えなければならない。メディアータ仕様記述での動的要素として以下のものがある。

- 情報源関係にあるオブジェクト、属性名を DD/D にいつ取得するか？
- 必要なルールをいつダウンロードするか？
- 大域的包摂関係 (束) をいつ生成するか？

そのために QUIK では、必要に応じたりフレッシュ機能をもつとともに、情報源の更新時にメディアータに対する自動反映機能、メディアータの実体化の3つを検討している。さらに共通モデルとメディアータ仕様記述を QUIK で共通化することによって、特定のメディアータで得られない解を上位のメディアータを用いることによって、求めることができる可能性がある。そのための対話環境も検討している。

6 地理情報システムでのオブジェクト同定

地理情報処理は、単に地図情報を扱うだけではなく、それに関連したイベント、行政区画、道路情報、気象情報などさまざまな情報を対象とする。最近マルチメディア情報が容易に計算機で扱うことができるようになったため、カーナビを始めとして急速に応用が拡大している。

地理情報の大きな特徴としてオブジェクトの連続性がある。たとえば河川は、流れが分岐したり、合流しながら、最後は海に流れ込む。それは支流や本流、あるいは流れている地域に依存して、名前を変えることが多く、それらの境界も必ずしもはっきりしていない。道路も同様な例である。別の例としてイベントがある。イベントの開催される地域は必ずしも行政区画とは一致せず、恣意的な要素もある。

もうひとつの大きな特徴としては、地理情報の中心をなす空間データは、それがたとえ複雑なものであったとしても、地球という空間座標の中で絶対的な位置情報を持ちうる。たとえば地図上で「銀閣寺」という名前であらわされる地理実体は、寺そのものを表したり、境内を表したり、周辺地区をも含む場合があるが、それぞれの空間実体は幾何学的には明確に表現できる。

2次元空間データを表現する空間オブジェクトを g_1, g_2 とする。具体的な幾何学的実体を表現するのは複雑なので、ここではその表現には触れない。これは空間的な包含関係が明確に設定できるので、その順序を $\sqsubseteq_G$ で定義する。集合論的な操作、合併 $\cup_G$ 、共通部分 $\cap_G$ 、差 $\setminus_G$ も定義できることを仮定する。この空間オブジェクトを使って地理オブジェクトを $o[\text{loc} = g]$ のように表現することにしよう。 o はアトム、 loc は空間オブジェクトを指す特殊な属性名、 g は空間オブジェクトである。このような表現をとることによって、地理情報の連続性は空間オブジェクトの中で幾何学的に表現することが可能となる。

QUIK オブジェクト o_1, o_2, o_3 、空間オブジェクト g_1, g_2 の間に

$$o_1 \sqsubseteq o_2 \sqsubseteq o_3, \quad g_1 \sqsubseteq_G g_2$$

という関係があったとき

$$o_1[\text{loc} = g_1] \sqsubseteq o_2[\text{loc} = g_2] \sqsubseteq o_3$$

と定義する。

地理的な名称では別名を付けることが多い。地理オブジェクト $o_1[\text{loc} = g_1]$ に別の地理オブジェクト o_2 を $o_2 = o_1[\text{loc} = g_1]$ と対応付けたとき、さらに o_2 の一部 $o_2[\text{loc} = g_2]$ を o_3 と対応させたとする。すると

$$o_3 = o_2[\text{loc} = g_2] = o_1[\text{loc} = g_1][\text{loc} = g_2]$$

が得られる。*QUIXOTE* ではこれは構文的に禁止しているが、空間データの特殊性を考え、

$$o[\text{loc} = g_1][\text{loc} = g_2] = o[\text{loc} = g_1 \cap_G g_2]$$

と定義する。上の例では、 g_2 は g_1 の部分空間であるので、 $o_3 = o_1[\text{loc} = g_2]$ が得られる。また $o_1 = o_2[\text{loc} = g_2]$, $o_2 = o_1[\text{loc} = g_1]$ という定義が与えられたとき、

$$o_1 = o_1[\text{loc} = g_1 \cap_G g_2] = o_2[\text{loc} = g_1 \cap_G g_2] = o_2$$

が得られ、これは矛盾ではない。

この地理オブジェクト関係を QUIK の包摂関係に組み込むためには、以下について考察しなければならない。

- オブジェクト項の簡約形

これは $[\text{loc} = g_1][\text{loc} = g_2] \rightarrow [\text{loc} = g_1 \cap_G g_2]$ という書き換えで求まる。

- 同値類からの代表元の選択

$\sqsubseteq$ での極大元の中で、より属性数の少ないもの、さらに辞書的順序を導入することで代表元を求める。

- 属性継承の妥当性

地理オブジェクトは一般的に集約的な属性が多い（人口、面積、…）が、これらは *QUIXOTE* (QUIK) の属性の上書き (overriding) や局所性 (locality) で対応する。

複数の情報源を統合しようというとき、人によるオブジェクト同定が行われることもある。したがってメディアータ仕

様記述としての QUIK プログラムの包摂関係の定義部には、境を実装した石田研究室の中西英之氏に感謝致します。

大域的包摂関係を生成する際に生じる同値関係の他に、利用者が自由に上のような同定関係を導入することができる。

このようなオブジェクト同定に基づくオブジェクトの融合だけではなく、履歴情報を持ったさまざまな情報を地理情報に付加することも検討してしている。

7 おわりに

本稿ではメディアータアーキテクチャを QUIK によって拡張した。その特徴は以下のようである。

- 演繹オブジェクト指向データベース言語 *QUIXOTE* を分散環境へ拡張し、新しい知識表現言語 QUIK を提案した。
- 情報源の共通モデルにルールや包摂関係を導入することにより知識表現能力を拡張した。
- 情報源識別子を構造化することにより、ひとつの情報源に複数の共通モデルを定義可能にした。
- メディアータ仕様記述言語に仮説生成能力をもたせることにより、問合せ能力を拡張するだけでなく、情報源の自動探索機能を付加した。
- メディアータ記述言語と共通モデル記述を同一言語化し、メディアータの階層化を可能にし、複数メディアータに対する問合せも可能にする環境を提供した。
- 分散地理データベースのためのオブジェクト同定の可能性を提案した。

今後の研究課題は、QUIK の言語仕様をさらに詳細に詰め、(Java による) 処理系の本格的な実装を行ない、地理情報処理の応用を始めとするさまざまな応用への適用可能性の評価である。

8 謝辞

本研究に関して御指導、御助言をいただいた上林彌彦教授に感謝致します。また、有益な御助言、御協力をいただいた上林研究室の地図グループの皆様と包摂関係の対話的統合環

参考文献

- [1] 横田一正, “マルチエージェントによるマルチデータベースの拡張”, 情報処理学会データベースシステム研究会 & 電子情報通信学会データ工学研究会合同ワークショップ, July 20-22, 1994.
- [2] 近藤秀文, 平井千秋, 横田一正, “束を用いた概念間の包摂関係に関する表現と利用方式”, 情報処理学会第 46 回全国大会, Mar., 1993.
- [3] 中西英之, “Java によるデータベースの動的再構築機構の実装”, 京都大学上林研インターン報告, 7 月, 1996.
- [4] Hassan Ait-Kaci, Robert Boyer, Patrik Lincoln, and Roger Nasr, “The Efficient Implementation of Object Inheritance”, *MCC Tech. Report AI-102-87*, 1987.
- [5] Y. Papakonstantinou, S. Abiteboul, and H. Garcia-Molina, “Object Fusion in Mediator Systems”, In *Proc. VLDB Conf., India*, 1996.
- [6] Kazumasa Yokota and Hideki Yasukawa, “Towards an Integrated Knowledge-Base Management System — Overview of R&D on Databases and Knowledge-Bases in the FGCS Project”, *Proc. Int. Conf. on Fifth Generation Computer Systems (FGCS'92)*, pp. 89-112, Tokyo, June 1-5, 1992.
- [7] Kazumasa Yokota, Toshihiro Nishioka, Hiroshi Tsuda, and Satoshi Tojo, “Query Processing for Partial Information Databases in *QUIXOTE*”, *6th IEEE International Conference on Tools with Artificial Intelligence (TAI'94)*, pp.359-365, New Orleans, Nov. 6-9, 1994.