情 報 学 基 礎 39−10
（1995. 9. 14）

文献情報を対象としたスキーマ統合の一手法とシステムの構築

西澤 格　　高須淳宏　安達 淳
東京大学大学院工学系研究科　学術情報センター研究開発部

本稿では、自律して動作している複数のデータベースを統合的に利用するための一手法を提案する。システム上に要素データベースを仮想的に統合した「仮想データベース」を構成し、仮想データベースの統合スキーマ上で定義された「仮想関数従属性」に着目することにより、ユーザの発行した問い合わせの意味を損うことなく、要素データベースに対して統合的なアクセス、および質問処理を実現する。本手法の有効性を検証するために、プロトタイプシステムとして個人的に管理されている比較的小さな文献データベースである BIBTEX を対象とし、クライアント-サーバモデルに基づいた文献情報の統合利用を可能とするシステムを構成したので、この紹介を行う。

Schema Integration Method for Bibliographic Information and System Implementation

Itaru Nishizawa　　Atsuhiro Takasu　Jun Adachi
Graduate School of Engineering, The University of Tokyo　　Research & Development Department, NACSIS

In this paper, we propose a new method to achieve integrated access to multiple autonomous databases having similar schema. The method is based on the construction of a "virtual database (VDB)" and the use of "virtual functional dependency (VFD)," which are formed over physically distributed component databases and applied to the query processing process. We subsequently developed and implemented a bibliographic information retrieval system based on client-server model employing the presented schema integration method such that integrated access is obtained for bibliographic information written in BIBTEX.

1 はじめに

近年のネットワーク技術の発展と、増え続ける情報を集中的に管理することの限界から、分散管理されているさまざまな情報をネットワークを介して利用するということが一般的に行われるようになっているが、ユーザ側からはこれらの情報資源を統合的に利用できるかどうかが最大の関心事となる。ここでの「情報資源の統合的利用」とは分散する複数の情報資源に対して、ユーザがその位置、異種性を意識せずにアクセスできるという意味である。具体的には、ユーザは複数の情報源が存在することを意識せずに問い合わせを発行すれば、何らかのメカニズムによって実際は複数の情報源からその答えが集められるということを意味する。

複数の情報資源を統合的に取り扱おうとする際には、解決しなければならない大きな二つの問題がある。一つは情報資源の管理のしかたとアクセス方法に関する問題、もう一つは各情報資源の間に存在するさまざまな異種性に関する問題である。本稿では特に異種性に焦点をあて、データベースにおけるスキーマ統合の研究という観点から、文献情報を統合的に取り扱うための手法を考察し、プロトタイプシステムの紹介を行う。

2 データベースへの統合的アクセス

2.1 想定する環境

本稿では図1に示すように属性が異なる複数のデータベースが存在し、これにユーザが統合的にアクセスを行うという環境を想定する。各要素データベースは独立した主体によって自律的に構成され、運用されているため、そのスキーマはその構造や実体の表現に違いがある。本稿では各データベースのスキーマは属性集合に限定されている場合を考えるが、これは異種データベース [3] 環境の一つの特殊な場合と考えることができる。各データベースのスキーマが属性集合に限定されているという仮定は、その対象を文献情報に限定した場合にはそれほど厳しいものではない。各々のデータベースを要素データベースと呼ぶ。本稿ではスキーマの構造および属性の意味 (semantics) に関しては対応表を与えるという仮定をおいているが、データベーススキーマの構造および属性の意味はその設計者によって異なるため、その知識を持つ人間が写像を与えるのは自然であると考えられる。また本稿においては、実体の同一性に関しては、意味の対応表を与えるという仮定から、その属性値の一致によってその同一性を規定するものとする。スキーマの構造および属性の意味の写像を与えるという仮定の下では、意味の異なる属性は異なる名前の属性に写像される。つまり、この仮定の下ではスキーマにおける属性の有無の問題がスキーマ統合の本質的な問題となる。異なる要素データベースのスキーマに上述の写像を施した後の例を図 2(a) に示す。

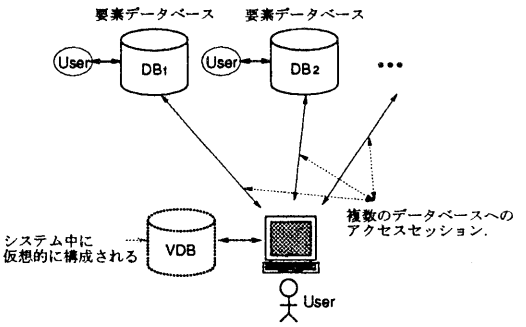

図 1: 複数データベースへの統合的アクセス

問い合わせは仮想的に構成される仮想データベース (VDB) に対して発行される。VDB の定義は 3.2 節でなされる。VDB に対して発行された問い合わせは 3.1節で定義される仮想関数従属性 (VFD) を用いることによって変形され、各要素データベースに対して発行される。システムは各要素データベースから得られた解を併合し、ユーザに示す。

2.2 問い合わせ処理の例

図 2(a) に示す二つの要素データベース DB_1、DB_2 を考える。この DB_1, DB_2 は内容はよく似たスキーマの文献データベースではあるが、独立した主体によって運用されている。ユーザ側の観点からは、それぞれ別々にアクセスするよりも、図 1 のような何らかのメカニズムによって、両者の区別を意識せずあたかも一つのデータベースにアクセスするようにして、問い合わせができると都合がよい。図 2(a) からもわかるように DB_1, DB_2のスキーマは似ているが、DB_1にのみ関数従属性 “著者 → 国籍” が存在する。この例の場合、VDB は図 2(b) のように構成され、VDB のスキーマの属性、「著者′」、「国籍′」の

DB_1

国籍	著者	タイトル
英国	Thomas	データベース
英国	Smith	情報検索
米国	Robert	人工知能

DB_2

著者	タイトル
Thomas	OS
Smith	信号処理
William	データベース

(a) 要素データベース

VDB

タプルID	国籍′	著者′	タイトル′
1	英国	Thomas	データベース
2	英国	Smith	情報検索
3	米国	Robert	人工知能
4	ω_1	Thomas	OS
5	ω_2	Smith	信号処理
6	ω_3	William	データベース

(b) 仮想データベース (VDB)

図 2: 例における要素データベースと VDB

間にはVFD として“著者′ → 国籍′”なる関係が成立する。ここで「著者′」、「タイトル′」などはそれぞれ要素データベース DB_1, DB_2上の「著者」、「タイトル」を VDB 上に写像したものであり、$\omega_1, \omega_2, \omega_3$は空値を表す。以下の議論で便利なように VDB にはタプルID を付加してある。

ここで「英国籍の著者の文献のタイトルを求めよ」という問い合わせ処理を考える。まず、VDB 上のタプルID1 および 2 に対応するタプルが DB_1から得られる。VFD と DB_1からの検索結果を用いることにより、Thomas および Smith の国籍が英国であることがわかり、DB_2からタプルID4 および 5 に対応する解が得られる。これは VDB において空値ω_1およびω_2が英国と演繹されることに相当する。この結果、解は“データベース”、“情報検索”、“OS”、“信号処理”となる。もし、同じ問い合わせを DB_1および DB_2に対して個別に発行したとすると、“データベース”、“情報検索”という二つの解しか得られないのに対し、本手法では論理的に正しくより大きな解集合を求めることができる。ユーザにとってはこちらの方が一層便利であると思われ、本稿ではこのような問い合わせ処理の実現法について論じる。

3 問い合わせ処理手法

2節で、想定する環境について述べ、問い合わせの処理の例を示した。本節では本アプローチの核となる仮想データベース (VDB, Virtual Database) と仮想関数従属性 (VFD, Virtual Functional Dependency) を定義し、問い合わせ処理のアルゴリズムについて述べる。

3.1 VFD の定義

本稿で述べる問い合わせ処理手法では、VFD を用いる。関数従属性とは関係データベースの属性間に成立する従属性の一つで、関係スキーマ Rと、Rに含まれる属性集合の部分集合 X, Yに対して Xに含まれる属性の値を全て定めると Yに含まれる属性の値が一意に定まるとき、属性集合 Xから属性集合 Yに対して関数従属性が存在するという。この関数従属性は関係データベースの正規化において重要な役割を果たす。

ここで以下の議論のための記号を定義しておく。要素データベース DB_i上での属性集合を U_i、DB_i上でのある一つの関数従属性を f、DB_i上での関数従属性の集合を FD_i、FD_iによって論理的に含意される関数従属性の集合 (FD_iの閉包 [5]) を FD_i^+で表す。関数従属性は右辺がただ一つの属性しか持たないものに限定する。これは関数従属性に関する和と分解規則 [5] を適用することによって一般性を失わないことは明らかである。

関数従属性が個々の要素データベース上で成立する関係であるのに対して、VFD はそれらを仮想的に統合した仮想データベース上で成立する関係である。VFD は要素データベース上の関数従属性の集合から構成されるが、その定義のしかたによっては本稿で提案する問い合わせ処理において効果がなかったり、構成される VDB 上のスキーマが矛盾を含むものとなってしまう。そこで、本稿では VFD を式 (1) のように定義する。式 (1) による VFD は直観的には注目する属性が存在するすべての要素データベース上で成立する関数従属性の集合である。ここで、関数従属性 fの両辺に現れる属性の集合を $attr(f)$ とした。

$$VFD = \{f \mid (\forall i)(attr(f) \subseteq U_i \Rightarrow f \in FD_i^+)\} \quad (1)$$

3.2 VDB の定義

スキーマ統合における VDB の概念は直観的には物理的に分散しているデータベースを合成するとい

うもので、そのスキーマは各要素データベースのスキーマの和集合となる。なお、本稿ではデータベースのスキーマは各データベース上の属性の集合によって構成されるものとする。以下の議論では、属性を $A_1A_2\cdots A_n$、属性の集合を Xで表すものとする。タプル t、属性集合 Xに対して $t[X]$ は属性集合 X からなる t のサブタプルを表す。例えばタプル $t=(v_1,v_2,\cdots,v_n)$ に対して $t[A_2A_3]$ は (v_2,v_3) である。本稿では対象とするデータベースを関係データベースとしているため、対象とするデータベース DBは スキーマ R、タプルの集合 r、関数従属性の集合 FD を用いて $DB=\langle R,r,FD\rangle$ という3項組で表すものとする。

このとき、$VDB=\langle R',r',VFD\rangle$ である。VFD は式(1) で与えられるものであり、$R'=\bigcup_i R_i$ である。ここで、i 番目の要素データベースを DB_i、DB_i のスキーマを R_i、VDB のスキーマを R'と表した。タプルについては、DB_iのタプルの集合を r_i、そのタプルの集合に対応する VDB のタプルの集合を r'_i と表す。i 番目のデータベース中のタプル集合 r_i中の j番目のタプルを t_{ij} と表すとすると、

$$t'_{ij}[A_k]=\begin{cases} t_{ij}[A_k] & A_k\in R_i\text{の場合} \\ \omega_{ijk}(\text{空値}) & \text{その他の場合}\end{cases} \quad (2)$$

である。この r'_iを用いて、VDB のタプルの集合 r' は $r'=\bigcup_i r'_i$と表される。

3.3 仮想データベースの形式化

VDB 上での VFD を用いた問い合わせ処理法によって得られる解集合は、VDB を構成せずに個々の要素データベースへ個別に問い合わせを発行した結果得られる解集合よりも大きくなる。これは関数従属性が、要素データベースにおいては空値となっている属性値を、他のデータベースから導出するためのルールの役割を果たすためである。そこで、この大きくなった解集合の正しさが問題となるが、VFD を適用して得られた解も論理的に正しく矛盾を含まないことが示されている。これはまず VDB を論理式を用いた公理によって表現し、問い合わせとそれに対する解も論理式で表現することによって、解の正しさを証明するという手続によって行われる [4]。

3.4 分散したデータベースへの問い合わせ処理

Tと FDから T_{FD}をすべて求めれば、問い合わせ処理に Reiter のアルゴリズムを適用することにより正しく、かつ完全な解集合を求めることができる。しかしながら要素データベースのスキーマ統合の過程では T_{FD}は大きくなる可能性があるため、T_{FD}を求める方法は現実的でない。よってここではすべての T_{FD}を求めずに解を求める方法を考える。

要素データベース上の属性と仮想データベース上の属性との関係は直観的には要素データベース上の各属性は属性をノード、VFD を有向リンクとして結合され、VDB 上に写像されていると考えることができる。式(1) の VFD の定義より、要素データベースの交わった部分についてのリンクに矛盾がないことは明らかである。論理的には、完全な解を求めるということは VDB 上で再帰的に VFD のリンクをたどりながらタプルに相当する基底アトムを計算していく必要があり、要素データベース間で推移閉包 [5] を求めることに相当する高価な処理を行う必要があるが、問い合わせによって参照されるノードに関して VFD が閉路を作らない場合は、分散環境下においては VFD のパス毎に各要素データベースで並行処理を行うことによって計算時間を短縮することができる。

3.5 アルゴリズム

関数従属性の左辺および右辺に現れる属性をそれぞれ $lhs(f)$, $rhs(f)$ とする。関数従属性の推移律より、2つの関数従属性 f, gについて、$rhs(f)\in lhs(g)$ ならば $lhs(f)\cup lhs(g)-\{rhs(f)\}$ から $rhs(g)$ への関数従属性が導出される。f,gから推移律によって導出される関数従属性を $f\circ g$で表すことにする。

各要素データベース D_iと D_i中に現れない属性 A に対して、D_iと A に対する *VFD-path* とは、以下の条件を満たすVFD 中の関数従属性の列 $(f_1,f_2,\cdots,f_n)$ である。(1)$rhs(f_1)=A$ (2) 各 iについて、$rhs(f_i)\in lhs(f_1\circ f_2\circ\cdots\circ f_{i-1})$ (3)$lhs(f_1\circ f_2\circ\cdots\circ f_n)$ は D_iの属性集合に含まれる。

図3にアルゴリズムを示す。3.1節で述べたように、U_{l_k}は要素データベース DB_{l_k}上の属性集合、$attr(f_l)$ は関数従属性 f_lの両辺に現れる属性の集合、3.2節で述べたように R_iは要素データベース DB_iのスキーマである。また、$\sigma,\pi,\bowtie$ はそれぞれ選択演算、射影演算、結合演算を表す。また、図3の第6ステップにおける選択演算 $\sigma_{A_j=W}$は、問い合わせ Wで示された条件から得られた解を推移的に適用していくことを表している。本アルゴリズム中では VFD を用い

```
   procedure GetTheory(VDB, VFD,
                ⟨x_1/τ_1, ··· , x_n/τ_n | W(x_1, ··· , x_n)⟩);
1. begin
2.    T := {};
3.    for each 属性τ_1, ··· , τ_n を含む
      要素データベース DB_i
      do begin
4.       for each W に現れ, DB_i に含まれない属性 A_j
         do begin
5.          for each DB_i と A_j に対する VFD-path
            (f_1 ∘ ··· ∘ f_k) do begin
6.             s_j = (⋃_{l_1} {σ_{A_j=W} π_{attr(f_1)} R_{l_1} |
                        attr(f_1) ⊆ U_{l_1}})
                    ··· (⋃_{l_k} {π_{attr(f_k)} R_{l_k} |
                        attr(f_k) ⊆ U_{l_k}});
               (従属性 f_1 ∘ ··· ∘ f_k に現れる属性から
               構成される関係 s_j を求める. )
7.          end;
8.       end;
9.       T := T ∪ π_{τ_1,···,τ_n}(R_i  s_1  ···  s_j);
10.   end;
11.   return T;
12.end;
```

図 3: アルゴリズム

ており、VFD は式 (1) で与えられる。この計算では関数従属性の閉包が式中に現れているが、この計算中ではある関数従属性ある従属性 $f : x \rightarrow y$ が FD_i^+ に属するかどうかの計算を行えばよい。この計算は FD_i に関する x^+ の計算が従属性の長さの和に比例するように実現することができること [5] を用いると $f : x \rightarrow y \in FD_i^+$ の判定は関数従属性の数 N_{FD}、属性の数 N_{AT} に対して $O(N_{FD} \cdot N_{AT})$ で多項式時間での計算が可能である。また、*VFD-path* についても $O(N_{FD} \cdot N_{AT})$ の多項式時間での計算が可能である。

4 問い合わせ処理の応用

4.1 BibTeX への応用

本節では以上に述べた問い合わせ処理方法の応用について考える。本手法の応用には実体を表現する属性が異なるような複数の情報が分散している必要がある。この身近な例として BibTeX を取り上げて、本提案手法の有効性を示す。

BibTeX は LaTeX を用いて論文を作成している研究者の間で広く利用されている文献データベースである。このデータベースは基本的に個人単位で管理されているため、その規模は比較的小さい。ユーザは本システムを介して、これらの文献データベースをネットワークを介して統合的に利用できる。BibTeX では使用者によって、文献を異なる型で管理している場合があり、その型に応じた情報の入力を行うため、情報が欠落している可能性がある。この欠落した情報を VFD を使うことによって「追跡」することができる。

BibTeX では文献情報を表 1に示す 14 個の型に分類する [7]。この分類はユーザの自由裁量であるため、人によって同じ文献の表現形態が異なることがあり得る。文献の型と、その型の文献が持つ属性は予め定められており、これをまとめて表 3に示す。各型の文献は必須、任意の 24 個の属性から構成され [6]、plain, alpha, abbrv, unsrt などの BibTeX の各スタイルに応じて、annote を除いた項目がタイプセットの際に使用される。実際にシステムを構成するに当たり、(1)Annote, Crossref, Key は除く (2) 属性の必須と任意の区別については考慮しないなどの簡単化を行った。ただし、スペースの関係で属性は表 2 で示した略語を使って表現されており、表中の o はその属性が必須であることを示し、x はその属性が任意であることを示す。添字である数字が同じ属性は同一文献情報中で OR の関係であることを示す。

表 3は仮想データベースのスキーマに相当する。問い合わせ処理では VFD を利用するわけだが、この VFD を構成するためには各型の文献の FD のセットが必要となる。BibTeX の 14 個の型の文献については、意味を考えてそれぞれ FD のセットを与えた。

4.2 作成したシステム

図 4にシステムの構成図を示す。プログラムは C 言語で記述し、SUN SPARC station 20, Solaris 2.4 上で動作を確認した。通信部分は RPC を用い、ユーザが入力した問い合わせおよび BibTeX ファイルのパーシングには lex/yacc を用いた。現時点でのプログラムの規模は約 14000 行である。以下サーバ側とクライアント側、通信部分についての説明を行う。

4.2.1 サーバ側

まず BibTeX ファイルを読み込んで字句解析および構文解析を行い、各フィールドとその値を抽出し、データベースを構成する。現在データは学術情報センター研究開発部の 4 名のものを用いており、件数

表 3: 文献の型と属性の関係

Document category	Field																		
	Ad	Au	B	Ch	Ed	H	I	J	M	No	Nu	O	Pa	Pu	Sc	Se	Ti	Ty	V	Y
article		o						o	x	x	x		x				o		x	o
book	x	o			x				x	x	x			o		x	o		x	o
booklet	x	x				x			x	x							o			x
conference	x	o	o						x	x	x	x	x	x		x	o		x	o
inbook	x	o		o	x				x	x	x		o	o		x	o	x	x	o
incollection	x	o	o	x	x				x	x	x		x	o		x	o	x	x	o
inproceedings	x	o	o						x	x	x	x	x	x		x	o		x	o
manual	x	x			x				x	x		x					o			x
mastersthesis	x	o							x	x					o		o	x		o
misc		x				x			x	x							x			x
phdthesis	x	o							x	x					o		o	x		o
proceedings	x								x	x	x	x		x		x	o		x	o
techreport	x	o					o		x	x	x						o	x		o
unpublished		o							x	o							o			x

表 1: 文献の型

文献の型	説　明
article	論文誌など発表された論文
book	出版社の明示された本
booklet	印刷、製本されているが 出版主体が不明なもの
conference	inproceedings と同じ (Scribe との互換性のため)
inbook	書物の一部 (章、節、文など何でも)
incollection	それ自身の表題を持つ、 本の一部分
inproceedings	会議録中の論文
manual	マニュアル
mastersthesis	修士論文
misc	他のどれにも当てはまらない 時に使う
phdthesis	博士論文
proceedings	会議録
techreport	テクニカルレポート
unpublished	正式には出版されていないもの

表 2: 属性名の略語

正式名	略語	正式名	略語
address	Ad	month	M
annote	An	note	No
author	Au	number	Nu
booktitle	B	organization	O
chapter	Ch	pages	Pa
crossref	Cr	publisher	Pu
edition	Ed	school	Sc
editor	Et	series	Se
howpublished	H	title	Ti
institution	I	type	Ty
journal	J	volume	V
key	K	year	Y

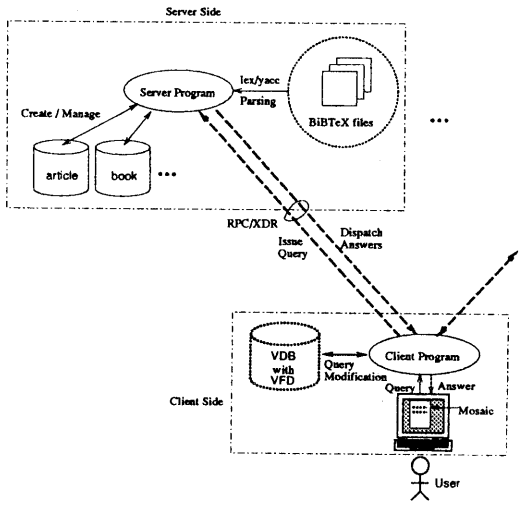

図 4: System 構成

は約 700 件である。BibTeX ファイルの解析処理は lex, yacc を使用して記述しているが、この際 BibTeX ファイルの性質を考慮して、

- 文字列の最初と最後がそれぞれ"{", "}"の場合はそれらを除いたものを値とする。
- 文字列の最初と最後が最初と最後に"“", "”" がある場合はそれらを除いたものを値とする。
- 文字列中の連続する改行、空白文字を一つのスペースに置き換える。

などの処理を行っている。また、ファイル中の文法誤りつまり BibTeX ファイルの記述誤りについては間違ったエントリのみを読みとばすという仕様になっている。

本システムではサーバとクライアント間の通信はSUN RPC (Remote Procedure Call) を用いている。サーバ側はBIBTEXファイルを読み込んでデータベースを構成した後はクライアントからの接続を待ち、クライアントからの接続要求が到着すると、クライアントから送られた問い合わせと、構成したデータベースのマッチングを行い、結果を返す。マッチングは現在Exactマッチングと Substringマッチングをサポートしている。

4.2.2 クライアント側

クライアント側ではNCSA Mosaic[1] を利用して、図5に示すようなユーザインタフェースを作成した。ユーザは左上のウィンドウで検索を行う文献の型を選択することができ、問い合わせを Fieldtype に対する条件の形、あるいは検索式の形で書くことができる。この Fieldtype ボタンはクリックすると全部の属性が表示され、その中から自分の検索を行いたい属性を選択することができるようになっている。検索式には OR を示す'+' および'|'、AND を示す'*' および'&'、優先順位をコントロールする'(' ')' を書くことができる。ここで、Fieldtype に対する条件入力の場合も、少し複雑な検索が可能であるように、入力項目を 2 つ準備し、その間の AND 演算、OR 演算をサポートした。

Mosaic インタフェースに対してのユーザの入力はsubmit ボタンを押すことによって、クライアントプログラムに引数が渡される。このクライアントプログラムは CGI (Common Gateway Interface)[2] プログラムであり、Mosaic インタフェースに入力された文字列を受け取り、選択された文献の型の FD のセットから VFD を構成する。また、インタフェースから渡された問い合わせを lex, yacc を用いて解析し、その問い合わせを選言標準形に変形し、RPC を用いてサーバに問い合わせを発行する。サーバから解が返されると、その解と VFD を用いて問い合わせを再構成し、もし必要であればもう一度サーバに問い合わせを発行する。これらのセッションの後、クライアントプログラムは 得られた解を Mosaic インタフェースに転送しユーザ示す (図6参照)。

4.3 システムの挙動

簡単な問い合わせの例を考えてみる。発行された問い合わせは `SCHOOL = "Univ.of Tokyo"` である。

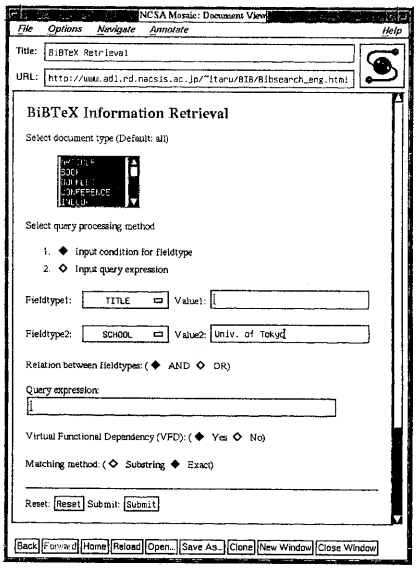

図 5: プロトタイプシステムのユーザインタフェース

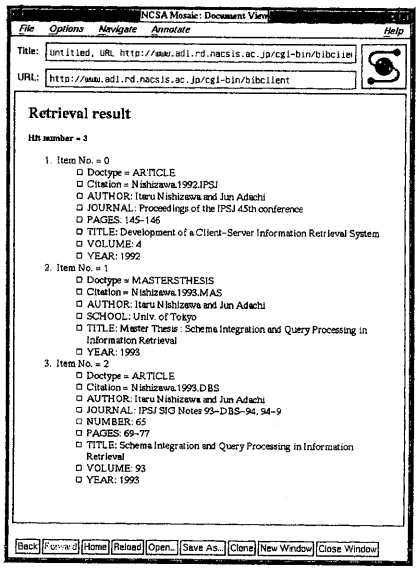

図 6: 検索結果

Answer ID. Document Category	
Attribute	Value
1. masterathesis	
AUTHOR	"Itaru Nishizawa and Jun Adachi"
SCHOOL	"Univ. of Tokyo"
TITLE	"Master Thesis: Schema Integration and Query Processing in Information Retrieval"
YEAR	"1993"

(a) VFD を用いない場合

Answer ID. Document Category	
Attribute	Value
1. article	
AUTHOR	"Itaru Nishizawa and Jun Adachi"
JOURNAL	"Proceedings of the IPSJ 45th conference"
PAGES	"145-146"
TITLE	"Development of a Client-Server Information Retrieval System"
VOLUME	"4"
YEAR	"1992"
2. mastersthesis	
AUTHOR	"Itaru Nishizawa and Jun Adachi"
SCHOOL	"Univ. of Tokyo"
TITLE	"Master Thesis: Schema Integration and Query Processing in Information Retrieval"
YEAR	"1993"
3. article	
AUTHOR	"Itaru Nishizawa and Jun Adachi"
JOURNAL	"IPSJ SIG Notes 93-DBS-94, 94-9"
NUMBER	"65"
PAGES	"69-77"
TITLE	"Schema Integration and Query Processing in Information Retrieval"
VOLUME	"93"
YEAR	"1993"

(b) VFD を用いる場合

図 7: 検索結果

この問い合わせの意味は「東京大学出身あるいは在学中の著者の文献情報を求めよ」である。VFD を用いない問い合わせ処理を行った場合 (クライアント側のインタフェースで VFD を No にした場合) と、VFD を用いた問い合わせ処理を行った場合 (クライアント側のインタフェースで VFD を Yes にした場合) の検索結果を図 7 にまとめる。

VFD を用いない場合には 'mastersthesis' 型の論文である一つの文献のみが解として求まるが、これに対して VFD を用いた場合には 'article' 型の他の二つの論文 (図 7 における ID=1, 3 の論文) も解として得ることができる。これはこの例の場合、VFD 中に $AUTHOR \rightarrow SCHOOL$ が存在するためである。この解は論理的に正しく、ユーザにとってはこのような解も得られる方がより便利であると考えられる。

5 まとめ

本稿では、情報源の異種性に対応するためのシステムコンセプトと、仮想データベースを利用する問い合わせ処理方法について説明し、仮想データベースを用いた問い合わせ処理機構を実装したプロトタイプシステムを作成し、評価を行った。仮想データベースという概念を導入することによって、ユーザは複数の情報資源を意識すること無く、しかもそれぞれの情報資源に対して個別に検索を行う場合よりも多くの情報を得ることが可能となる。本研究では、分散する情報資源を統合的に取り扱うためのシステムの構築を最終的な目標としている。計算機の普及によって、比較的小さなデータベースを個人的に作るということが行われるようになりつつあり、ネットワークを介してこれらのデータベースに対する統合的なアクセスを行うことは、疎結合の情報源に対するアクセス法の一つの可能性を探るものである。

参考文献

[1] "NCSA Mosaic Home Page," URL: http://www.ncsa.uiuc.edu/SDG/Software/Mosaic/Docs/help-about.html (1995).

[2] "The Common Gateway Interface," URL: http://hoohoo.ncsa.uiuc.edu/cgi/overview.html (1995).

[3] W. Litwin, L. Mark, and N. Roussopoulos: "Interoperability of Multiple Autonomous Databases," *ACM Computing Surveys*, Vol. 22, No. 3, pp. 267–293 (1990).

[4] 西澤格, 高須淳宏, 安達淳: "属性集合が異なる複数のデータベースへの統合的問い合わせ処理手法," 電子情報通信学会誌 D-I (投稿中) (1995).

[5] J. D. Ullman: *"Principles of Database and Knowledge-Base Systems Volume I and II,"* Computer Science Press (1988).

[6] 松井正一: "BibTeX の使い方," jbibtex-0.31 のパッケージに添付のドキュメント (January 1991).

[7] 松井正一: "日本語 BibTeX: jBibTeX," jBibTeXC Version キットと共に配布されている文書 (ver.0.20 用, 1989 年) (1991).