

## 情報検索におけるスキーマの統合と質問処理

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情報源である要素データベースを統合的に利用するための手法を提案する。要素データベースに対する統合的なアクセスを実現するために、システムの構成要素としてリソースオブジェクトという要素データベースのアクセスに必要な情報を管理するモジュールを用意し、これを利用することにより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」を構成する。仮想データベース上の統合スキーマは要素データベースのスキーマ情報から構成される。プロトタイプシステム ROBIN は仮想データベースの統合スキーマ上で定義された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。

## A Schema Integration and Query Processing in Information Retrieval

Itaru Nishizawa

Atsuhiko Takasu Jun Adachi

Graduate School of Engineering, Research & Development Department,  
The University of Tokyo NACSIS

The purpose of this paper is to propose a method to achieve an integrated access to multiple autonomous databases for information retrieval. We handle information resources as 'Resource Object' which is defined in order to form 'Virtual Database (VD)' over physically distributed databases. The VD has a virtual schema which is derived from schema of each component database and realizes an integrated access to the component databases.

ROBIN, a prototype system described in this paper, uses functional dependency on VD for the decomposition of a query issued to the VD. Therefore, it is possible to process a user's query without losing the semantics the user holds.

## 1 はじめに

1980年代以降の計算機の進歩には目を見張るものがあり、高性能のワークステーションが比較的簡単に利用可能となった。また、時を同じくしてコンピュータネットワークの技術もめざましい発展をとげ、現在では大学や企業の研究施設は高速のコンピュータネットワークによって結ばれつつある。これにともない各種の情報をネットワークを通じて利用したいというような要求が一層強くなってきた。ネットワークを介して各種の情報源データベースにアクセスを行なおうとする場合に重要となるのは、各データベースを統合的に扱うための方法である。このデータベースの統合問題は分散しているデータベースを取り扱おうとする際には避けられない問題である。

従来からの分散データベースの研究では、データが物理的に分散しているだけで、論理的には一つのデータベースとして扱い得るものを対象とする場合が多い。すなわち、データの可用性、信頼性、およびアクセス時間の向上などを目的としてデータが分散されるのであり、分散されている要素データベースにはデータベースシステムとしての異種性は存在せず、個々の要素データベースにおける自律性も存在しないというものである。しかし、既に稼働しているデータベースを統合的に利用しようとするアプローチも研究が進みつつある。これらのアプローチは連邦型データベースシステム (Federated Database System[3])、多データベースシステム (Multi Database System[1]) などと呼ばれる。連邦型データベースシステムは各要素データベースを統合的に取り扱うために大域的なスキーマを構成しこれを利用するというアプローチを取るのに対して、多データベースシステムでは各要素データベースの自律性を重視するため、大域的なスキーマの構成は行なわれないのが一般的である。

情報検索システムは既存の多数のシステムが独立に稼働しているという意味で、多データベースシステムの良い例である。そこで、本論文では分散して自律的に動作している異種データベースを統合的に扱おうとする場合の一つの例として、複数の情報源データベースを統合的に取り扱う情報検索システムの構築法について考える。

## 2 情報検索システム ROBIN

自律的に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に、システムは各種の情報を必要

とする。この必要となる各種の情報資源は本システムではリソースオブジェクト (以下 RO とする) という概念によって管理される。情報検索を行なおうとするユーザシステムはこの RO と通信を行なうことにより、RO の情報を用いながら情報源である要素データベースにアクセスを行なう。この情報検索システムのモデルを ROBIN (Resource-Object Based Information Retrieval System) と呼ぶ。

ROBIN はシステム上に構成された仮想データベースという概念を利用することにより、複数の情報源に対する統合的なアクセスを実現するが<sup>1</sup>、この仮想データベース上のスキーマは各要素データベースのスキーマ情報から生成される。ここではこの仮想データベース上のスキーマを統合スキーマと呼ぶことにする。統合スキーマの構成の際には、関数従属性というデータベースの持つ論理性に注目し、各要素データベース上の関数従属性を利用することによって意味的に矛盾の少ないスキーマ統合が行なわれる。

ROBIN では生成された統合スキーマを利用して、仮想データベースに対して発行された問い合わせを各要素データベース用の問い合わせに変換するが、この際の問い合わせの発行の判断を行なうアルゴリズムについては情報検索システムであることを考慮し、ユーザの発行した検索の意味を保持しながら、得られるべき解を含む、より小さな集合を得ることを目標にした。

### 2.1 従来の情報検索システムの問題点と ROBIN の特徴

従来の情報検索システムはそのほとんどが情報の蓄えられているデータベースにログインし、その上に実装されているデータ操作言語や組み込みコマンドを用いて検索を行うという、いわゆる TSS ベースのものである。このような情報検索システムは集中管理型のデータベースを前提としており、必要な情報が複数のデータベース上に分かれて蓄積されている場合には、以下のような問題が生じる。

1. 各システムにデータにアクセスする際のコマンド体系が異なるため、ユーザは異なるデータベースにアクセスするたびに、そのシステム固有のコマンドを使用しなければならない。

<sup>1</sup>ここでいう統合的という意味は、複数のデータベースに蓄積されている情報に対してユーザが明示的にデータベースを指定し、それぞれのデータベースから得た検索結果を合成する必要なしに情報検索が行なえることを指している。

- 2.各データベースによって、蓄えられているデータの構造および意味が異なるため、ユーザはそれぞれの意味・構造を把握しづらい。
- 3.2の問題があるにもかかわらず、SQLなどのデータ操作言語ではデータ構造を明示的に記述しなければならない。情報検索ではユーザがデータ操作言語を明示的に使用することは少ないが、ユーザは問い合わせを書くにあたり、そのデータベースのデータ構造を十分に理解しておく必要がある。

このような問題は、特にデータベースの構造などを理解していない一般ユーザにとっては、情報検索システムを利用する際の重大な障壁となる。そこでROBINではこのようなユーザにも容易に利用が可能な情報検索システムを構築するため、その構成にあたり従来のシステムと比較して以下のような特徴を持たせた。

#### 1.データの総称化機能

複数のデータベースにおけるデータの意味と呼称の違いをユーザができるだけ気にすることなく情報検索を行なうことができること<sup>2</sup>。

#### 2.情報源の位置透明性の実現

ユーザが、情報の蓄積されているデータベースの所在場所を認識する必要なく、情報検索を行なうことができること。

#### 3.利用者インタフェースのユーザ支援機能

ユーザができるだけ具体的なデータ操作言語、およびデータ構造を認識する必要なしに情報検索を行なうことができること。

### 2.2 システムの概要

2.1節でROBINの3つの特徴を述べたが、この特徴を実現するためのシステムのモデル、その構成要素について説明する。

分散環境下において自律して動作している各データベースを利用することを考えた場合、それらのスキーマをユーザ側の都合で変更することは実際上不可能であるので、ユーザ側に各データベーススキーマの異種性を吸収する機構が必要となる。そこでここでは情報源となる各データベースをサーバ、デー

<sup>2</sup>各情報源データベースとROBINとの通信におけるプリミティブなコマンドについてはこれを統一できると仮定した。このようなコマンドのセットとしては情報通信プロトコルANSI Z39.50[4]などがある。

タベースを利用しようとするユーザをクライアントとみなすものとする[6]。

ROBINのシステム全体の構成は図1のようになる。システム全体としては、検索を行なうために必要となる各種情報を管理するROと、システム上の構成要素であるマネージャが通信を行なうことにより、システム上に仮想データベースを構成する。この仮想データベースの情報をを用いることにより、複数の要素データベースに対して同時に検索が行なわれることになる。ROBINとはROを含めたシステム全体についての名前であり、以下で使用する‘クライアント’という言葉は、ROとセッションを行なうマネージャを実装した部分(図1参照)を指すものとする。マネージャはクライアントの構成要素であり、これと4.1節で説明するROが通信を行なうことにより、各要素データベースへの統合的なアクセスを実現する。ROBINは五つのROおよび、三つのマネージャをその構成要素として持つ。これらがどのような情報を管理し、どのような働きを行なうかを説明するにあたり、まずROBINがそのシステムの内部に構成する仮想的なデータベースと、それを利用する問い合わせ処理のアルゴリズムについて説明する。

### 3 ROBINにおける問い合わせ処理

#### 3.1 仮想データベース

ROBINは複数の情報源データベースを統合するために、システム上にアクセスの対象となる要素データベースを統合した仮想的なデータベースを構成する。ここではこの仮想的なデータベースを仮想データベースと呼ぶ。

#### 3.2 問い合わせ処理のなげれ

ROBINにおける問い合わせ処理のなげれについて簡単にその概略を述べる。ROBINによる情報検索過程において、まずユーザは3.1節で説明した仮想データベースに対して問い合わせを発行する。ROBINはその内部で問い合わせを解析・分解し、各要素データベースにその問い合わせを発行するかどうかの判断を行なう。問い合わせを各要素データベースに発行するかどうかの判断基準には、仮想データベース上の関数従属性に関する情報を利用する[7]。以下、仮想データベース上における統合スキーマの構成法と関数従属性の定義方法について説明する。

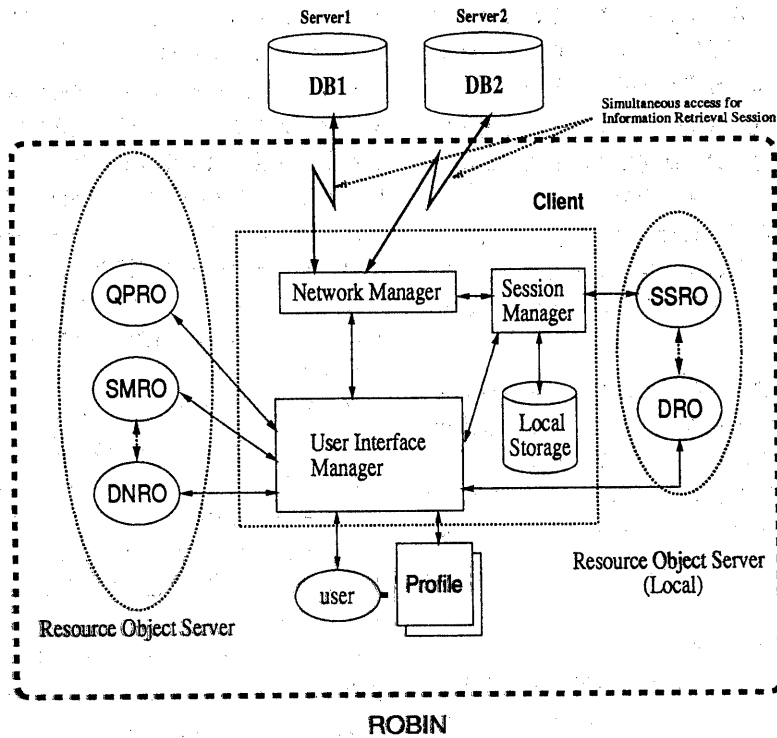


図 1: システム構成

### 3.3 仮想データベース上のスキーマ

要素データベース  $DB_i$  上の属性の集合を  $U_i$  ( $i = 1, 2, \dots, N$ ) とした時, 仮想データベース上での属性の集合を統合スキーマを  $\cup_i U_i$  と定義する. ここで問題となるのは各要素データベース上の属性名の違いだが, これについては写像関係を与え, その属性名を統一する. この統一された属性を概念属性と呼ぶ. なお, 本論文では以下で要素データベース上の関係が一つであるとの前提をおいている. データベースは複数の関係から構成されるのが普通だが, 本稿では普遍関係 (Universal Relation[5]) の考え方によりあらかじめ仮想的に複数の関係の Join が行われていると考える.

### 3.4 仮想データベース上の関数従属性

仮想データベース (以下 VD と表す) 上の関数従属性について考えると, その関数従属性の定め方としては以下の 3 通りが考えられる.

1.  $D_{valid}$ : すべての要素データベース上で成立している関数従属性を, VD 上の関数従属性とする.
2.  $D_{all}$ : その属性が存在する, 全ての要素データベース上で成立する関数従属性を, VD 上の関数従属性とする.
3.  $D_{any}$ : いずれかの要素データベース上で成立する関数従属性を, VD 上の関数従属性とする.

これら 3 つの関数従属性の定式化を行なう. いま要素データベースを  $DB_i$ ,  $DB_i$  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  $D_i$ ,  $D_i$  によって論理的に含意される関数従属性の集合 ( $D_i$  の閉包) を  $D_i^+$  とした時, 全ての要素データベースで成立する関数従属性の集合  $D_{valid}$ , およびある要素データベースで成立する関数従属性の集合  $D_{any}$  は以下のように表すことができる.

$$D_{valid} = \{f \mid (\forall i)(f \in D_i^+)\} \quad (1)$$

$$D_{any} = \{f \mid (\exists i)(f \in D_i^+)\} \quad (2)$$

次に、その属性が存在する全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  $D_{all}$  は、上で定義した記号の他に要素データベース  $DB_i$  上での属性集合を  $U_i$ 、関数従属性  $f$  の両辺に現れる属性の集合を  $attr(f)$  と表すものとする、以下のように書くことができる。

$$D_{all} = \{f \mid (\forall i)(attr(f) \subset U_i \rightarrow f \in D_i^+)\} \quad (3)$$

### 3.5 問い合わせの解析と分解

まず実際のデータベースに発行される問い合わせについて考える。この時、存在しない属性についての条件はその要素データベースでは評価することは不可能であるので、実際にその要素データベース上に存在する属性に対する条件のみを含んだ問い合わせを発行することになる。VD に発行された問い合わせ  $Q$  を選言標準形に分解したうちの一つの項  $Q_i$  を、要素データベース  $DB$  に対する問い合わせに変換したものを  $Q_{i(DB)}$  と表すものとする、

$$Q = Q_1 \vee Q_2 \vee \dots \vee Q_i \vee \dots \vee Q_N \quad (4)$$

$$Q_{i(DB)} = \{q_{ij} \mid q_{ij} \in Q_i, \text{Item}(q_{ij}) \in \text{Items}(DB)\} \quad (5)$$

である。但し、 $\text{Item}(q_{ij})$  はリテラル  $q_{ij}$  が参照する属性、 $\text{Items}(DB)$  は要素データベース  $DB$  上に存在する属性の和集合を表すものとする。ここで  $Q_i$  を考えたのは、検索の意味を考えた場合、ユーザから与えられた検索式を選言標準形に変形したとき、OR で結合された条件式は別々に評価してよいと考えられるからである。

### 3.6 問い合わせ発行の判断

3.5節で実際に発行される問い合わせの定式化を行なったが、この問い合わせを全ての要素データベースに対して発行するのは意味がないので、次に問題となるのはその生成された問い合わせを実際に発行するかどうかの判断である。ここではこの判断基準について考える。

#### 3.6.1 関数従属性を考慮しない定式化

仮想データベース VD に対して発行された問い合わせを実際に各要素データベースに問い合わせるか

どうかを考えた時、まず考えられる判断基準は選言標準形に分解された問い合わせ中で参照する属性が全て存在する要素データベースにのみ、問い合わせを発行するというものである。

要素データベース  $DB$  に対して選言標準形に分解された一つの問い合わせ  $Q_i$  を発行するかどうかについての真偽値を  $J_{DB}(Q_i)$  で表すものとする以下のように書ける。

$$J_{DB}(Q_i) = \bigwedge_{q_{ij} \in Q_i} \{\text{Item}(q_{ij}) \in \text{Items}(DB)\} \quad (6)$$

#### 3.6.2 関数従属性を考慮した定式化

前節の (6) 式による定式化は論理的に正しい解<sup>3</sup>しか返ってこないという点では厳密であるといえるが、その応用を情報検索システムとして考えた場合は問題もある。つまり、(6) 式による評価によって発行された問い合わせはユーザが発行した問い合わせに対して論理的に正しい解が返すが、それは実際にユーザが得たい解の部分集合となっているため、ユーザが検索を行なおうとした情報を見失う可能性がある。これは情報検索システムとしては不十分であると考えられる。

情報検索システムとしてはこのような事態は避けるべきであり、ユーザの発行した検索の意味を保ちながら、ユーザが得たい解を含んだ、なるべく小さな解集合を得ようとするのが最良の戦略であると考えられる。そのためにここでは、各要素データベース上に存在する関数従属性を利用し、それを判断基準とすることにより、(6) 式よりも大きな解集合を得るための判断式の定式化を行なう。

記号の意味は (6) 式に従うものとする。このとき検索発行の判断を行なう真偽値  $J_{DB}(Q_i)$  を以下のように定義する。

$$J_{DB}(Q_i) = \bigwedge_{q_{ij} \in Q_i} \left( \{\text{Item}(q_{ij}) \in \text{Items}(DB)\} \vee (X \rightarrow \text{Item}(q_{ij}) \in D) \right) \quad (7)$$

ただし、

$$X = \{\text{Item}(q_{ik}) \mid \text{Item}(q_{ik}) \in \text{Items}(DB), q_{ik} \in Q_i, k \neq j\} \quad (8)$$

である。

<sup>3</sup>ここで論理的に正しい解とは、返ってくる解は必ず発行した条件を完全に満たしているということである。

## 4 ROBINの構成法とその構成要素

3節でROBINにおける問い合わせ処理の方法について述べたが、本節では実際にこれらの機構を実際にシステムとして実装するためのシステムの構成要素について説明する。

### 4.1 リソースオブジェクトサーバ

リソースオブジェクトサーバはその中にROを保持している。ROはクライアント上の各マネージャと通信を行ない、クライアントが、サーバである要素データベースにアクセスを行なう際に必要な各種情報を返す。ROは各要素データベースのスキーマ情報を用いて構成されるもので、情報のカテゴリ別に要素データベースを統合した仮想データベース(3.1節参照)を構成する。ROBINではシステム全体で五つのROが存在する。この五つのROについてその内部構造、および働きについて説明する。DNRO, SMRO, QPROについてはその例も示す。この例においては、その取り扱う情報は文献情報である。

#### 1. DNRO(Database Name RO)

DNROは、要素データベース名とそのデータベースの持つ情報のカテゴリ、およびそのカテゴリにおける仮想データベース上の概念属性(3.3節参照)に関する情報を保持する(図2, 3参照)。DNROにおける概念属性は、同一カテゴリの情報を蓄積する全ての要素データベースのスーパーセットとなっている。

#### 2. SMRO(Semantics Mapping RO)

SMROは要素データベース名と、その情報のカテゴリに対応する仮想データベース上の概念属性と要素データベース上の属性とのマッピング情報を保持する(図4参照)。このマッピングについては、DNRO上で定義された概念属性に対する要素データベースの固有属性が記述されるが、概念属性は固有属性のスーパーセットとなっているため、仮想データベース上の概念属性に対応する固有属性が要素データベース上には存在しないという問題が現れる。この場合、固有属性のフィールドにはNull値を書き込んでおくものとする。

#### 3. QPRO(Query Processing RO)

QPROは要素データベース名と、そのデータベース上での関数従属性についての情報を保持

している(図5参照)。ROBINでは検索を行なう際に、クライアント上で仮想データベースの関数従属性についての情報を利用するが、これにはQPROの情報が用いられる。この関数従属性の生成は3.1節で説明した関数従属性取得のアルゴリズムを適用している。

### 4. SSRO(Search Strategy RO)

SSROは検索方針についての情報を保持する。ここでいう検索方針に関する情報とは、次に説明するDRO中の辞書の利用法に関する情報、情報のカテゴリに応じた属性に関する統計情報、仮想データベースの構成方法に関するストラテジーなどである。

### 5. DRO(Dictionary RO)

DROはシソーラス辞書、同義語辞書などの辞書情報を保持する。この辞書の情報はSSROの検索方針によってその使用法が決定される。ある情報のカテゴリにはそのカテゴリ内で頻繁に使用される語の集合が存在するため、辞書の利用は効果的であると考えられる。

### 4.2 マネージャ

マネージャはROBINのシステム上にあり、ROと通信を行ない、その情報を利用することによりシステム上に仮想データベースを構成する。

#### 1. ユーザインタフェースマネージャ(UIM)

ROBINにおける情報検索過程では、ユーザはシステムから示される概念属性への検索条件として、問い合わせを表の形式で記述する。UIMはこのユーザからの入力を、DNRO, SMRO, QPROなどの各ROとセッションを行ない、その情報を利用することにより特定の要素データベースへの問い合わせに変換し、ネットワークマネージャに送る。また、ネットワークマネージャから送られてきた検索結果をユーザに示すなどの働きを行なう。

#### 2. ネットワークマネージャ(NM)

NMはUIMから得た各要素データベース向けの問い合わせに従って、所望の要素データベースとのセッションを確立し、これを管理する。このセッションにおいて、NMは要素データベースごとにプロセスを生成し、その内部に専用のタイマを持つことにより、ネットワークの障害、

情報のカテゴリ名 (仮想データベース名)	要素データベース名
DOC	COMP,NORTH,INSP

図 2: DNRO における情報のカテゴリ

DOC における概念属性
AUTHOR
TITLE
LANGUAGE
CITATION
PUB-YEAR
ABSTRACT
KEYWORD
CONFERENCE
ISSN
ISBN
LIBRARY
PUB-COUNTRY

図 3: DNRO における概念属性

	データベース名		
	COMP	NORTH	INSP
概念属性	要素データベース上の属性		
AUTHOR	AUTH	LN	AUTHOR
TITLE	TITL	TI	TITLE
LANGUAGE	LANG	-	LANGUAGE
CITATION	CITN	JO	-
PUB-YEAR	YEAR	YR	PUB_DATE
ABSTRACT	ABST	AB	ABSTRACT
KEYWORD	FLTM	KW	KEYWORD
CONFERENCE	CONF	-	-
ISSN	ISSN	SN	ISSN
ISBN	ISBN	-	-
LIBRARY	-	LI	LIBRARY
PUB-COUNTRY	-	-	COUNTRY

図 4: SMRO によるマッピング

概念属性	左の概念属性が関数従属である属性集合
LANGUAGE	{AUTHOR, TITLE}, ISSN, ISBN
...	...

図 5: QPRO によるマッピング

サーバでの検索の失敗に対処する。

### 3. セッションマネージャ (SM)

システムがユーザに検索結果に応じたフィードバックを行なうにあたり、それまでの入力、およびその入力に対する検索結果を管理する必要がある。これを行なうのが SM である。SM は SSRO の情報を用いながら、検索結果に応じて次の検索のストラテジーを決定し、そのストラテジーに従ってメッセージを他の 2 つのマネージャに送る。

## 5 プロトタイプの実装

提案した情報検索システム ROBIN のプロトタイプを SUN ワークステーション上に実装した。プロトタイプでの実現範囲は

1. 要素データベース上での関数従属性の集合から仮想データベース上での関数従属性情報を生成する機構
2. 仮想データベースに対して発行された問い合わせを分解・変形し、情報源データベースに対して発行する機構

である。仮想データベース上における関数従属性については 3.4 節で説明した三通りの関数従属性をユーザが状況に応じて選択できるような機構を用意した。また、実際に問い合わせを変形・発行する機構についてもその関数従属性を利用するかしないかはユーザに選択が可能である。仮想データベース上では三通りの関数従属性の集合  $D_{valid}$ ,  $D_{all}$ ,  $D_{any}$  を定義したが、 $D_{valid}$  を用いた場合は論理的にも正しい解のみが返される。これに対して  $D_{all}$ ,  $D_{any}$  を用いた場合は解の集合の大きさを検索の意味を損なわずに大きくすることができる。これは ROBIN がユーザの意図に応じて解の大きさを変更することができることを意味する。

検索発行の判断基準として、 $D_{all}$ ,  $D_{any}$  を用いた場合には、ユーザが発行した問い合わせに対して論理的には正しくないものが含まれてしまう。また、厳密に言えば正しい解を全て含んでいるともいえない。しかしながら、もし論理的に正しい解集合のみを得ようとした場合には、その解集合は必ずユーザが意図した解集合の部分集合となってしまう。また、正しい解を全て含んだような解集合を得ようとした場合、全ての問い合わせを要素データベース上に存

在しない属性は除いた上で発行する必要があるが、これはユーザが発行した検索の意味を著しく損なうと考えられる。本稿で提案した検索発行の判断基準  $D_{all}$ ,  $D_{any}$  は、論理的な解の正当性と、ユーザが発行した検索の意味の保存のトレードオフとして位置付けられる。

プログラムは C 言語で書かれており、ユーザインタフェースマネージャ (UIM) についてはそのインタフェースを XVIEW を用いることによって実装を行なった。要素データベースとのセッションに関しては、要素データベースをそれぞれ ORACLE リレーショナルデータベースシステム上に仮想的に実現し、ユーザから UIM に対して発行された問い合わせを 3 節で説明したアルゴリズムに従って処理し、それを SQL に変換して ORACLE に送り、結果を得るというエミュレーションを行なった。

## 6 考察

従来の情報検索システムは、それぞれの情報源データベースに密に結合する形で構築されていたため、その利用に際しては、1) どの情報検索システムにどのような情報があるのかわからない。2) それぞれの情報検索システムのコマンドが異なる。3) 情報検索システム上の属性が異なる。4) 複数の情報検索システムで検索を行なった場合、その結果をユーザが合成しなければならない。などの問題があった。ROBIN はこのような従来の情報検索システムに対する問題点を改善し、しかも既存の情報検索システムに対する変更を比較的少なくし、さらに各情報検索システムの自律性を損なわないように設計されている。

これは多データベースシステムによるデータベースの統合と共通する部分があるが、多データベースシステムが大域的なスキーマを構成しないのに対して、ROBIN では情報のカテゴリごとの統合スキーマを構成し、仮想データベースに対して発行された問い合わせをその関数従属性に関する情報を用いることにより、発行された検索の意味を保持しながら各情報源データベースに対して動的に問い合わせを生成・発行するという点に特徴がある。

この機構は情報検索システムというアプリケーションを利用するユーザが、その操作に不慣れである時より一層その効果を発揮すると思われる。



## 7 むすび

本論文では分散した情報検索システムを統合的に扱う手法についての提案を行ない、その手法をとりいれた統合的情報検索システム ROBIN を提案し、そのプロトタイプを作成した。

統合的な情報検索システムを構築するにあたっては二つの大きな課題が存在する。一つは複数の情報源の管理をいかにして行なうかということ、もう一つは発行された問い合わせをいかにして評価し、複数の情報源に対して発行するかということである。

まず一つめの課題である情報源の管理とアクセスの問題については、本研究では各情報源へのアクセスのために必要な情報を、その機能要素ごとにリソースオブジェクトという情報管理のための情報資源によって管理するというモデルを提案し、その情報を用いることによって柔軟な情報の管理機構を実現し、情報源の位置透明性を確立することができた。これにより、ユーザは従来の情報検索システムの利用の際の障壁となっていた情報源データベースの選択、検索システム上の属性名の相違、およびコマンドの相違を気にすることなく情報検索が行なえるようになる。

二つめの課題である問い合わせの評価の問題については、まず複数の情報源データベースを仮想的に統合したデータベースとして、仮想データベースという概念を導入し、このデータベースに対して情報検索を行なうことによって、各要素データベース上の情報検索システムの属性の総称化を行なうことができた。この仮想データベース上の属性と、各要素データベース上の属性は、そのマッピングが前述の RO によって与えられる。

要素データベースの、仮想データベースへの統合の際に発生する属性値の有無によって発生する問題と、その際の検索発行の判断については、要素データベース上の関数従属性から仮想データベース上の関数従属性を生成することにより、それを判断基準としてユーザの発行した検索の意味を保持しながら、得られるべき解を含んだ解集合を得るための手法を提案し、プロトタイプの動作からそのアルゴリズムの有効性を確認できた。

## 参考文献

[1] Witold Litwin, Leo Mark and Nick Roussopoulos: "Interoperability of Multiple Autonomous

Databases", *ACM Computing Surveys*, Vol.22, No.3, pp267-293, 1990.

- [2] RAYMOND REITER: "A Sound and Sometimes Complete Query Evaluation Algorithm for Relational Databases with Null Values", *Journal of the ACM*, Vol.33, No.2, pp349-370, 1986.
- [3] Amit P.Sheth, James A.Larson: "Federated Database Systems for Managing Distributed, Heterogeneous, and Autonomous Databases", *ACM Computing Surveys*, Vol.22, No.3, pp183-236, 1990.
- [4] American National Standard Z39.50-1988, "Information Retrieval Service Definition and Protocol Specifications for Library Applications", developed by The National Information Standards Organization.
- [5] JEFFREY D. ULLMAN 著, 國井, 大保 訳: 「データベースシステムの原理」, 日本コンピュータ協会, (1985).
- [6] 西澤 格, 安達 淳: "クライアント-サーバモデルによる情報検索システムの提案", 第 45 回情報処理学会全国大会講演論文集 (4), pp145-146, 1992.
- [7] 西澤 格, 高須 淳宏, 安達 淳: "異種スキーマをもつデータベースへの統合的なアクセス手法", 第 46 回情報処理学会全国大会講演論文集 (4), pp91-92, 1993.