

## 視覚情報を用いた RDB 検索システム:OBAQ

加藤哲朗、鈴木幸市  
NTT DATA 開発本部

視覚情報を用いたリレーショナルデータベース検索システム OBAQ における設計手法及び評価について報告する。OBAQ は、「視覚情報を用いたオブジェクト指向による RDB 検索法」として既に報告した。今回は、その後の改良点を含む OBAQ の概略と設計における考えかたを明らかにし、OBAQ の操作方法と性能の評価を行う。

操作方法評価は、OBAQ と SQL インタプリタの操作方法の修得状況の比較と上記試験の被験者による主観評価により、また、OBAQ の性能評価は応答時間、描画時間、CPU 時間の測定により行った。

操作方法の評価では、簡単なデータベース検索において SQL に対する OBAQ の操作方法修得時間の優位性が示され、主観評価においても OBAQ は高い評価を得た。OBAQ の処理時間は、描画時間や RDBMS 等のその他の処理時間に比べ小さく、視覚表現による性能劣化がほとんど問題にならないことが明らかになった。

## OBAQ: An RDB manipulation system with visual user interface

Akio KATO Kouichi SUZUKI  
Development Headquarters  
NTT DATA Communications Systems Corporation

This paper describes a design concept and an evaluation of performance and operation of OBAQ (Organized Bitmap Assist Query) RDB manipulation system, which was proposed in "A Visual Interface for an Object Oriented Query in Relational Databases". Functional enhancement from the former version is also reported.

The operation method of OBAQ was evaluated by comparing the number of commands learned in 40 minutes lecture and the mean opinion score (MOS) for OBAQ and SQL commands. Performance of OBAQ was estimated by response time, drawing time and CPU time.

OBAQ got higher MOS than SQL commands and it was much appreciated even by operators who are accustomed with SQL commands.

Performance estimation shows that CPU time consumed by OBAQ is much shorter than both query processing time and screen drawing time.

## 1 はじめに

我々は、ビットマップディスプレイ及びポインティングデバイス(マウス)、マルチウィンドウシステムなどのヒューマンインタフェース向上をめざすハードウェア技術の進歩を背景に視覚的データベース検索システム OBAQ を提案した [1]。OBAQ は、リレーショナル・データベースシステム上に構築されており、コンピュータ操作に不慣れた利用者向けに視覚的で、簡単にかつ対話形式でデータベース検索システムを実現することを目的としている。OBAQ の構成図を図 1 に示す。図に示したように、DBカーネルとのインタフェースは SQL ベースでデータベースアクセスを行っているが利用者に関し SQL コマンドを意識させずに OBAQ は実現されている。

本稿では OBAQ の設計の考えかた及びその評価について述べる。

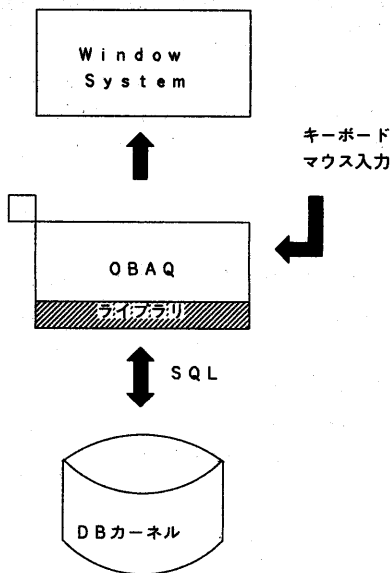


図 1 OBAQ の概略図

## 2 OBAQ の概要

### 2.1 OBAQ の設計方針

OBAQ を設計するにあたり、留意した点について述べる。まず想定したユーザについて考察を行い、その後、ユーザの利用形態について考察する。

データベースシステムのユーザとして、以下の 3 種類が考えられる。すなわち

1. システム設計者 (SYSTEM DESIGNER)
2. データベースシステム管理者 (DBA)
3. エンドユーザ (END USER)

である。

最初に挙げたシステム設計者は、エンドユーザ向けにデータベースシステムの開発をし、サービスを提供する。ライブラリ等を利用してデータベースアクセス用のプログラムを開発したり、必要なテーブル/ビューの設計を行うなどの開発を行う。従って、初心者向けのデータベース操作システムの必要性は設計者自身にはあまりない。

2 番目のデータベースシステム管理者はデータベースのユーザ管理、システムの維持管理、媒体管理、性能管理を行うものであり、システムの管理用コマンドを用いて管理する。これらの仕事を行うものはデータベースシステムに通じており、SQL のようなリレーショナル・データベース操作言語にも習熟していると期待でき新たに簡易な操作を目的としたアプリケーションの必要性は少ない。

3 番目のエンドユーザは初めてコンピュータに触れるものやデータベースを使用したことのないものなどが想定されて、その習熟レベルは様々なものが予想される。データベースシステムには、文字ベースの端末を前提とした種々のアプリケーションが開発されており、データベースアクセスに習熟しているユーザに対しては、適切なアプリケーションが用意されているといえる。しかし、このレベルの利用者のうちデータベースアクセスに関しても習熟していないもの特にそのうちコンピュータ操作にもほとんど初めて接するものに対するアプリケーションのサポートが不十分であるといえる。

OBAQ の利用対象者としてはこの 3 番目のエンドユーザ、にコンピュータ操作に習熟していないものを意識した。従って、ユーザはキーボード操作に対して抵抗感があり、マウスなどのポインティングデバイスもあまり使用したことがないことを OBAQ の設計の際に前提とした。従って、OBAQ は、基本的にマウスを多用することによりキーボード使用を減らし、操作方法及び情報の表現をユーザに視覚的に訴えることとした。また SQL のようなコマンドで行うように複雑な検索条件をすべて満たすような設計を行うと、指定の仕方が難しくなり操作の簡易性が損われることを怖れ、OBAQ におけるデータベース操作コマンドの表現能力よりも視覚的に対象を表現して直観的に操作方法が理解できるインタフェースの実現を優先させた。

さらに、OBAQ は特定のテーブル、ビューを対象としたシステムではなく任意にユーザが定義したテーブルを扱う汎用システムとした。これは、OBAQ のように会話的に行うデータベース操作システムでは定型的な処理よりも非定型的な処理に向いておりそのためには汎用データベースシステムとしての機能を持たせたほうが有利であると判断したためである。

以上で述べてきた目的を達成するため、OBAQ では SQL で代表されるデータベース検索言語を陽に用いることなくビットマップ上に視覚的に表現さ

れたデータベース検索システムを構成し検索、削除等のデータベース操作を会話的に処理することとした。

次に OBAQ の具体的な設計にあたり、まずデータベースにおけるエンドユーザの検索手順を想定し OBAQ をどう設計すればよいか考察した。ユーザは一般に図 2 に示したように、まずデータベースシステムにログオンした後、操作対象であるテーブル/ビューを特定することが必要となる。そのため、名前を覚えていない場合まず操作可能なテーブル/ビュー一覧を表示することとなる。また、この時に名前を知っているユーザにおいても SQL コマンドで行うような名前をキーボード入力するよりもその一覧の中から目的のものを選択し、自らタイプすることなく指定できるようにすれば入力誤りを避けることができる。従って、この一覧表示は OBAQ において必須であるとみなすこととした。

1. データベースにログオンする  
ログオン名/パスワード入力が必要
- ↓
2. 操作対象を探す  
名前はシステムから入手可能  
記憶している : 直接その名前を用いてアクセスする  
記憶していない: 一覧表示をしてその中から探す
- ↓
3. 検索対象の絞り込み  
SELECT コマンドでカラム名を用いてそのカラムの条件を与え検索対象を絞り込む
- ↓
4. UPDATE, INSERT, DELETE  
データの変更等がある場合に行う
- ↓
5. この操作対象に対して作業が残っている → 3へ
- ↓
6. 他のテーブルの操作が残っている → 2へ
- ↓
7. データベースをログオフする

図2 データベースの検索手順

さらに、カラム名もテーブル/ビュー名がわかればデータディクショナリから入手可能であり、カラム名の一覧表示は可能である。この機能も OBAQ で必要と考えた。従って、この一覧表示機能を用いれば、ユーザのキーボード入力なしにカラム名を指定することができる。このように、ユーザがどうしてもキーボードを使用して入力しなければならない場合というのは、操作対象のテーブル/ビュー名の指定と、カラムに対する検索条件、その他に入力/変更のデータのみであり、それ以外は省略させることができる。OBAQ のように視覚的で操作が容易、かつ会話的に処理するためにマウス使用を増したアプリケーションを設計する際にはマウスとキーボ

ードの持ち替えを減少させるためにも必要最低限のキーボード入力以外はマウスで行うように設計すべきであると判断し、この自動的にテーブル/ビュー名やカラム名の表示機能を持たせることとした。

また、テーブル/ビューの名前を探すこととテーブル/ビューのカラム名を使用することは前者は操作対象の探索であり、後者は操作対象の指定のために用いるとの目的が異なることから、OBAQ の持つ機能をデータディクショナリ (DD) 機能とデータ操作 (DM) 機能とに分類した。これは、先程の図 2 で「2. 操作対象をさがす」と「3. 検索対象の絞り込み」にそれぞれ対応する部分である。SQL のようなコマンド言語では、これらはひとつのコマンドとなり分割不可能であるが OBAQ のように会話的に処理することを考慮するとまず目的のテーブル/ビューを決定してからその内容について条件をしぼり込むと考える方が自然でありこの分類はそれを前提に行った。言い換えると、リレーショナル・データベースでのテーブル、ビューといったデータ単位を管理する機能とデータの値そのものを取り扱う機能という観点で分類したことになる。これらは、具体的には、テーブル/ビュー名の管理とテーブル/ビューのデータの操作といった操作を異なった取り扱いをすることを意味し、OBAQ ではこれらを別の機能として取り扱うこととした。これらをそれぞれビジュアル・データ・ディクショナリ (VDD)、ビジュアル・データ・エディタ (VDE) と呼ぶ。

以下の章では、VDD と VDE について述べ、その後 OBAQ の評価について述べる。

### 3 VDD

#### 3.1 VDD の設計

VDD は、リレーショナル・データベースのテーブル/ビュー等を視覚的に管理するための機能であり、一つのテーブル/ビューを一つの図形 (シンボルと呼ぶ) で表示し OBAQ のコマンドの意味は別の図形表現 (アイコンと呼ぶ) で表わした (図 3)。従って、シンボルを用いてユーザがアクセス可能なすべてのテーブル/ビューを一画面に表示しては、目的のテーブル/ビューを探すのが大変となるため、テーブル/ビューを階層的に管理し、必要なものだけ表示できる機能を持たせた。また、図に示されるようにアイコンは具象的なデザインになっており、サンプル及びシンボルは抽象的なデザインにしてあり区別してある。これらのデザインはユーザの好みの形に編集可能である。

VDD は、SQL コマンド相当の機能としては、テーブル/ビューの削除、生成の際の名前の付与といった機能を持つ。

VDD の設計にさいしては、エンドユーザの操作の簡便性、視覚表現の定義法について考慮し、それ

によりユーザの記憶の補助となるようにした。以下の章では操作の簡便性と視覚表現について解説する。



アイコン



シンボル/サンプル

図3 VDDの図形表現

### 3.2 操作の簡便性

ビットマップディスプレイを有するワークステーションにおいては、ユーザインタフェース向上の手段としてキーボード操作を避けて、マウスを使用する方法が有力である。しかしながら、ワークステーションの操作に不慣れなユーザにとってマウス操作といえども必ずしも簡単とはいえない。コマンドベースであれば、そのコマンドの索引等をリファレンスマニュアルで調べるなりして操作方法を見つける方法があるが、マウス操作では、コマンドをマウスの操作で行うためコマンド名がわかりにくく、マニュアルで調べることが困難である。そのため、マウス操作の統一は、視覚表現を多用するアプリケーションでは、重要な課題となる。また、同様な操作を同じ方法で取り扱うことはユーザに対する記憶負荷も小さくできる。例えばメニュー表示は右ボタンで常に行うと約束すれば、いつでも例外なくメニュー表示ができ、ユーザにとって快適な環境といえる。OBAQでは、マウス操作を簡単にするため必要な動作を2種類に絞った。すなわち、「指定」と「移動」である(表1)。表1では、OBAQで使用するすべてのマウス操作を整理した。表より明らかなように画面上の位置を指し示す場合(通常メニュー<sup>1</sup>を含む)に左ボタンを使用しポップアップメニューを表示・選択する場合のみ右ボタンを用いて操作方法の統一をはかっている。また、VDDとVDEでの使用法の異なる部分はどちらかでのみ実現している機能であり共通の部分は、統一されている。

### 3.3 VDDの視覚表現

OBAQにおいては、テーブル/ビューを表現するために図形表現を用いたため数多くのテーブル/ビューを持つユーザはそれらを整理する機能が必要

<sup>1</sup>通常メニューはメニューの一覧表示であり、ウィンドウボタンの押下で表示されマウスの左ボタンでその一覧メニューの中から目的の項目を選択する

操作	意味	使用
左ボタンのクリック	1. ウィンドウ操作枠内の機能ボタン選択	共通
	2. 通常メニューの選択	
	3. ウィンドウ内のフィールド選択	
右ボタン押し	ポップアップメニューの表示	共通
左ボタン押下+移動+	アイコン、シンボル、サンプルの移動	VDD
離す(「移動」)	カラム幅の変更	VDE
左ボタンのダブルクリック	VDE起動(シンボル)	VDD

表1: マウス操作一覧

となる。そのため、VDDにおいては、これらの図形表現を管理するための機能として階層構造を持たせた。具体的には、VDDのトップレベルはルートフォルダと呼ばれるフォルダでありフォルダの中には、テーブル、ビューとフォルダ(まとめてシンボルと呼ぶ)を持つことができる。フォルダは上記で述べたもの以外に、テーブル/ビューに対する機能を表現するアイコンとVDD上でテーブル/ビューと対応するシンボルの定義に用いるサンプルから構成される。VDDの表示例を図4に示す。これらの、図形表現の位置は、マウスを用いて好きなところに移動させることができる。

アイコンはVDDで用意した機能に対応し数式における演算子に相当するものを図形表現したものである。従って、これらは単独で用いられず引数に相当する操作対象のシンボルの指定が必要である。これらの名前は「編集機」、「ごみ箱」、「書類断機」、「複写機」、「台車」の5つである。ただし、「工具箱」は、その他の雑多な機能を表現しており操作対象のシンボルの指定は不要である。

シンボルまたはサンプル(総称してオブジェクトと呼ぶ)は数式における引数に相当しオブジェクトをマウスを用いてアイコンに重ねることでオブジェクトに対してアイコンの機能を実行することを表現する。これらの組み合わせは意味をなさないものが

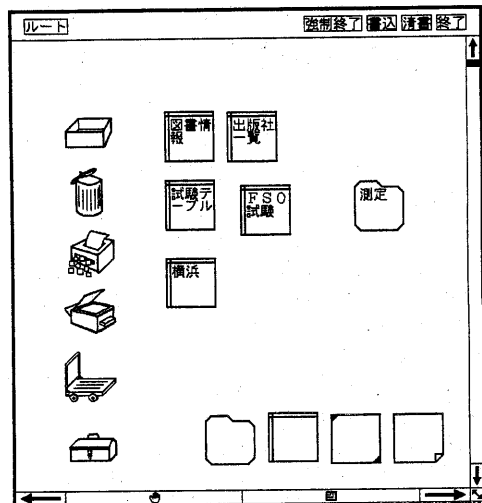


図4 VDDの画面例

あり、すなわち構文的に許されても意味的に許されないものも存在する。アイコンとシンボル、サンプルとの組み合わせで意味をもつものを表2に示す。表2にあるようにアイコンの操作対象は、ほとんどシンボルであり、サンプルを用いるのは「複写機」のみである。また、シンボルでアイコンとの演算が不可能なのは、VDDの階層構造管理のために導入したフォルダに対して対応するSQLコマンドが無い等の無意味な場合である。すなわち、「書類破砕機」では、DROPコマンドを発行しているため「編集箱」に対しては、VDEを起動すること自体フォルダには意味が無いためである。三角は、警告の意味をこめて、中が空でないフォルダは捨てられないようにしたことを意味している。

	シンボル			サンプル		
	テーブル	ビュー	フォルダ	テーブル	ビュー	フォルダ
複写機	○	○	○	○	○	○
編集箱	○	○	×	×	×	×
破砕機	○	○	×	×	×	×
ごみ箱	○	○	△	×	×	×
台車	○	○	○	×	×	×

表2: VDDの演算

また、組み合わせの意味を表3に示す。表には、多くの操作が示されているが、マウスの操作としては「指定」と「移動」の2種類のみで済んでいる。以下では、実例としていくつかのアイコンの機能の説明を行う。

一般に演算子に相当するアイコンはその機能を実行して終了するが、例外としてごみ箱がある。ごみ箱は別な見かたをすれば、どのフォルダからも移動可能な特殊なフォルダでその中には捨てたオブジェクトと台車から構成される。

編集箱は、ビュー生成時などに複数のテーブルを指定する際に用いる。すなわち、ビュー生成の際にもととなるテーブルをまとめて「指定」する場合にこの編集箱にテーブルを「移動」させることで表現する。

台車はフォルダ間の移動に用いるものでこの上にあるオブジェクトはどのフォルダに移っても参照可能であり、目的のフォルダに移ってからそのフォルダにオブジェクトを移動させることでフォルダ間の移動を実現した。

道具箱はその他の機能を示すアイコンであり、これを「指定」したときにメニュー表示し、対応する機能をメニューから選択して使用する。

以上に示してきたように、VDDはメニューを選ぶこと（「指定」+選択）とマウスを用いてオブジェクトを移動させる（「移動」）2つの基本操作を覚えればよい。また、アイコンの示す機能も具体的であり、例えばVDD上のテーブルを消去（VDD上で非表示にする）する場合は、そのオブジェクトをごみ箱の上まで「移動」させることにより、ごみ箱に捨てることを意味し消去コマンドの実行とする。

操作	対象	意味		
指定	アイコン	編集箱	VDEを起動する	
		ごみ箱	ごみ箱の内容を表示する	
		道具箱	次の補助機能メニューを表示する ・データディクショナリの参照 ・ロケーションスキーマの参照 ・アイコンエディタの起動 ・DB転送等	
	シンボル	フォルダ	フォルダの内容を示す画面へ移る	
		テーブル	VDEを起動する	
移動	サンプル	複写機	空のシンボルを生成する 実体との対応を取る	
	シンボル	編集箱	VDEへ渡す	
	テーブル	複写機	実体の内容を含めて複写する	
		破砕機	シンボルと実体を削除する	
		ビュー	ごみ箱	画面からシンボルを消す
	シンボル	台車	一時的な置き場所とする	
		シンボルの移動	表示位置の変更	
	フォルダ	複写機	名前の変更を指示する	

表3: マウス操作とその意味

## 4 VDE

本節では、操作する対象のテーブル/ビューを実際に取り扱うVDEについて解説する。リレーショナル・データベースにおいて取り扱うデータは関係としてタブルの集合で表現される。これらは、一般に表形式で表わされる。表編集処理とRDB検索操作との類似性を表4 [2]に示す。表4にあるように行の追加、削除、変更、カラムの追加、並び変えといった操作とSQLコマンドは対応ととることができ、行の順番のようなリレーショナル・データベースの概念と合わないものが一部あるものの表の編集操作とRDBのコマンドとの対応はきわめてよい。VDEはこれらの類似性に着目しリレーショナル・データベースの操作、検索、削除、変更、挿入をあたかも表の編集を行うように実行することで行うことができるように設計したものである。例えば、タブルの削除は、表の相当する行を削除することに、変更は当該行を書き換えることに対応する。

RDB検索操作	表編集操作
SELECT FROM [WHERE]	パターンマッチング
INSERT INTO	行追加
DELETE FROM [WHERE]	行消去
UPDATE [WHERE]	行消去
ALTER [WHERE]	カラム追加
不可	カラム削除
SELECT カラム名	カラム並び変え
不可	行並び変え
CREATE TABLE+INSERT	表生成

表4: SQLコマンドと表編集処理操作の対応

また、VDEでは、VDDとは異なりデータ投入が多いためデータエントリ時などキーボード入力している場合にはマウス操作をいれずに連続して入力可能となるよう各フィールドの移動などは、キーボードの矢印キーやコントロールキーで操作できるようにしている。

さらに、多量のデータ投入のために同様な入力の軽減をはかりウィンドウ間転送の機能を多用している。すなわち、行単位の複写やVDEの他の部分や他のアプリケーション(ワードプロセッサの文書)からの文字列の複写/移動が可能であり同じ文字列のタイプを避けることができる。

検索における条件の設定方法は、当初SQLにおける条件節の部分に相当するものを会話的に設定する方式としていたが簡単なパターン一致等の条件でも操作が煩わしく検索しにくいとためQBEにおけるような検索条件を表形式に示す方式(図5)を導入した。この場合、条件ウィンドウに検索式の条件に相当するボタンや範囲を条件のカラムの下にあるフィールドに書きこめばよいとため簡単な検索が容易にできるようになった。

面積	人口	世帯数	調査日	区分
	> 200000	< 100000		

図5 検索条件の入力例

区名	面積	人口	世帯数	調査日
旭	32.79	237987	75641	1986-09-01
神奈川	23.38	201999	76357	1986-09-01
港南	19.87	212321	68112	1986-09-01
鶴見	31.35	240147		1986-09-01

図6 VDE画面例

タブ削除は操作が容易で、消したい行にカーソルを移動させメニューを表示させ、メニューの中から削除を選択すればよい。

図6にVDEの画面例を示す。図6は図5とほぼ同じ画面であり、図にあるようにカラム名はテーブルより自動的に入手して表示するためカラム名を記憶しておく必要はない。また、図5のように条件を指定する場合にも対応するカラムの位置にその条件を入力するためカラム名入力の手間が不要となる。このように、VDEはデータベースの操作として、挿入、変更、削除が表を編集する感覚で行うことができるためデータベースの概念を知らないものでも容易にその操作方法を覚えデータ投入等が楽に行うことができる。次の節ではこの効果を評価した結果について述べる。

## 5 OBAQの評価

### 5.1 OBAQの評価

ユーザインタフェースの評価は個人の好みの問題もあり客観的に判定することは難しい。同じユーザインタフェースを持つシステムに対して全く逆の評価が下される場合もあり得る。OBAQでは、操作性の評価として実際の使用者による使いやすさの採点と使用法の学習速度の測定を行った。結果の図を図7、8に示す[3]。測定方法は、5種類のデータベース操作を用意しそれぞれ同等のコマンドをSQL言語とOBAQを用いて被験者に学習してもらいその学習レベルを測定したものである。被験者としては、SQL使用経験のある者(2名)とSQL使用経験のない者(3名)に依頼した。SQL使用経験を有するものに対して測定を行った理由は、OBAQの学習速度にSQLやデータベースの知識が影響を与えるか比較するため行った。図7は、学習速度を測定したもので、初めに20分2度目と3度目に10分のコマンド及び使用法の説明を行い、その直後にその5つのコマンドの使用法をテストし正解した個数を示したものである。図より明らかに操作方法の学習はOBAQの方が速やかに行われており、またSQL

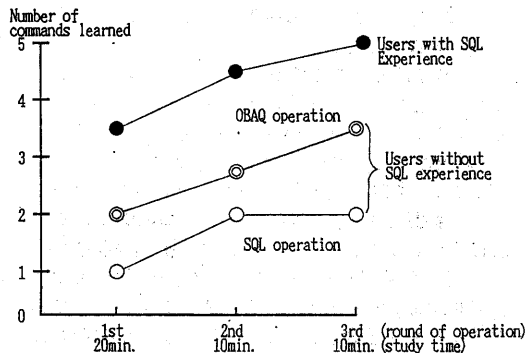


図7 OBAQとQBEの学習曲線

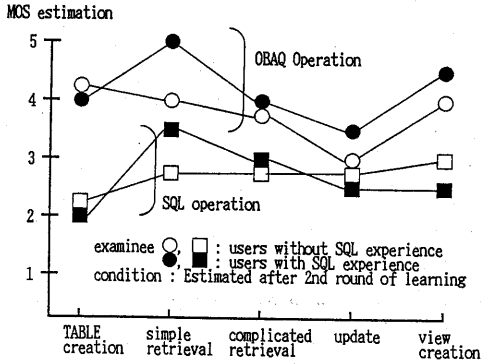


図 8 OBAQとSQL操作のMOS主観評価

またはデータベースの知識を有するものは速やかにOBAQの使用法を記憶することを示している。図8は、学習速度の測定の後には被験者にSQLとOBAQの使いやすさを評価させた結果である。図に示されているように、SQL使用経験の有無に関係なくOB

AQの使用のほうが楽であると答えており、このことから、視覚表現を用いたデータベース操作の優れていることがわかる。

## 5.2 OBAQの処理時間

OBAQを用いることにより生ずるOBAQの内部処理時間を見積もるため直接データベースからデータを関数インタフェースで入手した場合との所要時間の比較を行った。DBMSシステムはoracle[4,5]を使用した。

測定としては、1000件、5000件および10,000件のデータベースを用意し、ユニークな値をもつカラムにインデックスをつけ、その中から1、5、10、15、20件の検索を行いその時に要したCPU時間と描画時間及び検索用に発行した関数の所要時間を測定した。件数をこのように比較的少ないところを設定したのは、人間が会話的に問合せに答える時間を除くためである。すなわち、OBAQは一度に多くの検索結果を不用意に表示することを避けるため一画面に表示し切れない場合にユーザーに対して検索件数を示して画面に表示するかどうか問合せを行っており、一画面に表示できる範囲に測定件数を制限したためである。5000件の場合の測定結果を図9と図10に示す。他の件数でも測定結果は同様であった。

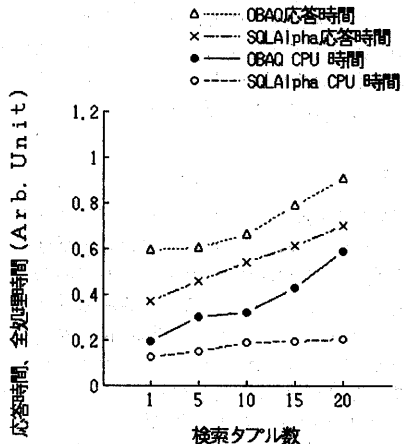
比較用として、SQLコマンドインタプリタ(以下SQLAlphaと呼ぶ)を用いて、データベース検索関数の所要時間を測定した。SQLAlphaは専用アプリケーションの処理時間を見積もるために用意したコマンドインタプリタで、データベースからSQLコマンドでデータを検索するために必要最低限の処理のみ行うアプリケーションである。従って、SQL

Alphaで、要した処理時間との差がOBAQでの内部処理におけるオーバーヘッドとみなすことができる。

図9はCPU時間とその他の処理時間の所要時間をOBAQとSQLAlphaとで比較して示したものである。図からわかるようにCPUの所要時間は両者とも同じであり、OBAQの内部処理のためのオーバーヘッドはほぼ一定であることがわかる。

図10には、OBAQとSQLAlphaそれぞれの描画時間を含めた場合の所要時間の比較である。図に

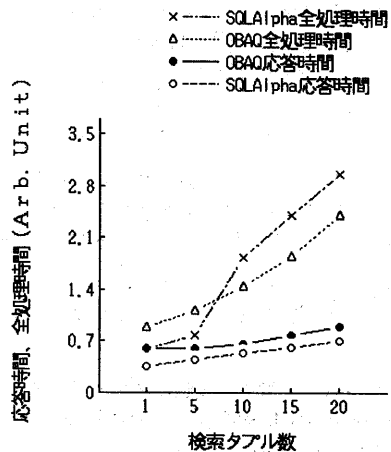
### 応答時間及び処理時間特性



データ件数5000件  
インデックス有

図9

### 描画時間も考慮した場合の処理時間特性



データ件数5000件  
インデックス有

図10

示されているように、描画時間の割合は、件数に比例して増えるのに対し処理時間はあまり増加していない。OBAQ の内部処理の時間が描画時間にかかる時間を越えることは無いことが示されている。また、SQLAlpha でもそれほど検索時間の違いは見られず OBAQ の内部処理によるオーバーヘッドはそれほど見られないことがわかった。

OBAQ の試験以外でも、OBAQ を実際に何人かのデータベース及びコンピュータの初心者を使用してもらい部内の OA 用のデータベース作成を行った。データ投入に関しては漢字変換等の方法を覚えればほぼ問題なく投入が行え、検索に関しても簡単なパターン検索等ならすぐに覚えられた。従って、初期の目的である初心者向けの汎用データベースシステムの実現は十分達成できたと思われる。

## 6 まとめ

操作が簡易な初心者向け汎用データベースシステム OBAQ の機能の説明、および OBAQ の評価を行った。OBAQ は実際の初心者による使用経験およびその後の調査より、操作方法の修得に効果が有ることが明らかになった。また、OBAQ 内部の処理によるオーバーヘッドはデータベースアクセスに使用する時間や描画時間に比べ比較的小さくすんでおり、OBAQ 内部処理時間における問題は特に無いことが明らかになった。現在、著者らの部内で OA 用として OBAQ を実際に資産管理等に用いており、DB システムの使用経験のない人でも十分使用に耐えることが実証されている。

### 謝辞

OBAQ のオーバーヘッド測定に際し、比較用として用いた SQL コマンドインタプリタ SQLAlpha を提供していただいた本橋征行氏及び測定及び作図に協力していただいた大木由美氏に感謝します。

## 参考文献

- [1] 加藤・鈴木・織田・外川「視覚情報を用いたオブジェクト指向による分散 DB 検索法」、情処研報、Vol.87, No.4 87-DB-57, pp.79 - pp.86 1987
- [2] 加藤・鈴木・外川: 「リレーショナルデータベース SQL 言語の視覚表現法」:70 周年記念総全大、6-87, 1986
- [3] 加藤・外川・鈴木・織田「視覚情報を用いたオブジェクト指向 RDB 検索法」、信学論 (D)、Vol.J70-D No.11 November, 1987
- [4] Oracle Corp. "SQL, The quiet revolution", 1986
- [5] Jack L. Hursch and Carolyn J. Hursch, "Working with Oracle", 1987