

垂直分散型オフィス情報システム向きの データベース簡易言語†

進藤重平^{††} 天水昇^{†††} 坂本恵市^{†††}
川手寛^{†††} 黒川裕彦^{†††}

パソコン (PC) とホスト計算機とを接続して構成されるオフィス情報システム (OIS) に関する技術的問題を論じる。具体的には、①PC とホスト間の処理およびデータベース (DB) の分散、②複数社のソフト・ハード製品と接続可能とする拡張性、③分散処理環境でソフト生産性向上を狙う高水準言語、の3点を主に論じる。「分散」の問題については、PC とホストとに分散配置された DB の管理、特に更新方法が重要である。DB の更新主体および更新契機が特定される場合は、両 DB を非同期に更新する方式が、PC 側の負荷を増加させずに DB 間の矛盾を回避でき、有利である。「拡張性」の問題については、PC 側プログラムを取り巻く環境 (機種、データベース管理システム (DBMS)、通信回線等) に依存する部分を局所化することにより、環境に依存して改造の必要な部分の全プログラム量に対する割合を、各々数%以下とした。「高水準言語」の問題については、PC 側、ホスト側の両方に簡易言語を用意した。この結果 DB 検索、帳票出力、グラフ表示等を含む経営数値情報システムに適用した例では、従来の COBOL 等を使用する場合に比べ、非専門家が業務構築したにもかかわらず、1/5~1/6 の工数で完成させることができた。

1. はじめに

オフィス業務の効率化を狙いとして OIS が導入されてきた。

オフィスで使用される最重要メディアである「文書」の取扱い (流通・保管) を機械化することを狙ってワープロ (WP)、FAX、電子メール、文書管理等のシステムが開発されてきた。

近年、オンライン技術、DB 技術の進展を受けて、OIS は文書内容の作成あるいは意志決定を支援する DB 中心型のシステムへと進展してきた。これらのシステム構築手段として、大規模なホスト計算機上の DB を利用者が直接アクセスする、SQL、QBE 等の DB アクセス言語が開発され、インプリメントされてきた¹⁾。

しかし、OIS を構成する技術・製品には以下の動向がみられ、システム構成・機能に関し、新たな視点が必要になった。

(1) PC の低価格化、高機能、高性能化に伴って、マン・マシンインタフェース (MMI) の高度化、応答時間の短縮等を狙った垂直分散 (MML) 処理技術が進

展してきた。また、安価な大容量ファイル媒体が PC 側でも使用可能となり、処理の分散のみならず、DB の分散も可能となってきた²⁾。

(2) 大きな OA 市場を目指して PC、LAN、PC 流通ソフト (AP、DBMS) 等のハード・ソフト製品が多数開発・販売されている。これら複数社のコンポーネント製品を自由に組合せて安価で使い勝手の良いシステムの構築が可能となった。

(3) ソフト生産性の向上、ソフトウェアバックログの解消等を狙いとして、第4世代言語が開発・販売されている³⁾。これらの第4世代言語をみると、PC スタンドアローンのものあるいはホスト集中型のものが主流であり、MML 環境に適合する製品は少数である。

このような動向を踏まえ、垂直分散型オフィス情報システム向きのデータベース簡易言語 (VGUIDE*) を開発した⁴⁾。本論文では以下の3点の技術的問題について、設計方針、実現方式、評価等を論じる。

- ① OA システムにおける処理および DB の分散
- ② 複数社の製品 (ハード、ソフト) を組み合わせる拡張性
- ③ 分散処理環境でソフト生産性向上を狙う AP 構築高水準言語

2. VGUIDE の概要

VGUIDE システムは、図1に示すように、ホスト

† Data Base Query Language for Micro-Mainframe Link Type Office Information System by JYUHEI SHINDOH (NTT Network Systems Development Center), NOBORU TENMIZU, KEIICHI SAKAMOTO, HIROSHI KAWATE and HIROHIKO KUROKAWA (NTT Communications and Information Processing Laboratories, Nippon Telegraph and Telephone Corporation).

†† 日本電信電話(株)ネットワークシステム開発センター
††† 日本電信電話(株)NTT 情報通信処理研究所

* Visual & General User Interface for relational Database Environment.

系と PC 系から構成され、それぞれが DBMS を持つ垂直分散型の構成となっている。ホスト系は、DIPS リレーショナル DBMS⁵⁾ (RDBMS) 配下で VGUIDE プログラムが動作する。一方、PC 系は、16 ビット CPU を持つ市販 PC 上に OS として MS-DOS、RDBMS として informix/SQL がのり、その配下で VGUIDE の PC 系プログラムが動作する構成である。

さらに、PC 系プログラムは、OIS を構成する上級管理層、中間管理層、実務担当層、システム構築/保守・運用層の四つの階層に対して、表 1 に示すように、その利用目的、利用形態に応じたメニュー形式、

表形式、カード形式、コマンド形式の各 MMI を設けている。また、MMI 部と共通処理部を分けた構成とし、共通処理部に分散制御機能を持たせることにより、DB アクセス命令の相違を吸収し、ホスト/PC 両 DB に対して同一の MMI でアクセス可能としている。

3. 分散処理

処理の分散と DB の分散とを論じる。

3.1 処理の分散

3.1.1 設計方針と実現方式

16 ビット PC が普及し、市販ソフトの流通が盛ん

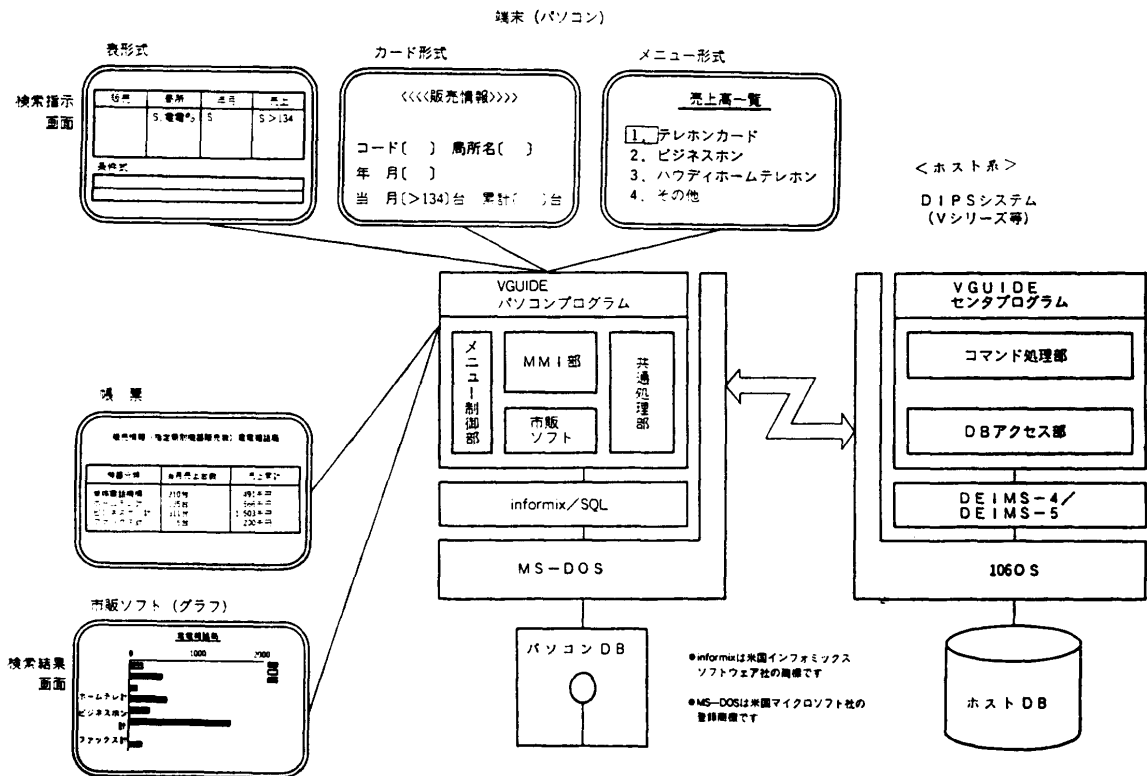


表 1 OA のユーザ層と MMI
Table 1 MMIs for layers of OA users.

| 項番 | ユーザ層 | 利用目的 | 利用形態 | MMI |
|----|------------------|-------------|----------------------------|--------|
| 1 | 上級管理層 (経営幹部) | 経営実態把握、意志決定 | カーソルとリターンキーのみの検索、グラフ表示等 | メニュー形式 |
| 2 | 中級管理層 (部門管理者) | 実績数値の各種分析 | 各種データを組み合わせた簡単なコマンドによる条件検索 | 表形式 |
| 3 | 実務担当層 | データの投入・管理 | 伝票イメージ等によるデータの入力・検索 | カード形式 |
| 4 | システム構築、保守・運用層 | 業務の追加、変更等 | 業務 AP 構築、変更、アドホックな検索、更新 | コマンド形式 |

になってきた。さらに 32 ビット PC の登場により、従来ホスト集中型で発展してきた OIS が分散処理に進む方向にある。PC に分散される処理としては、まず MMI、図形表示、次に表やグラフの作成がある。ホストに残る機能としては、共通あるいは大容量 DB の保管あるいはフルテキストサーチ処理、特殊機器の共用等がある。

DB を検索し結果をグラフ表示するモデル (図 2) では、転送データ量、PC 側 DB とホスト DB との整合、PC 側での市販ソフトを使った分析等を行うため、PC からホストへは SQL 命令、ホストから PC へは表形式データのタプル単位の転送、とした。

3.1.2 評 価

VGUIDE で採用した上記の方式は、SQL を検索言語として使用する限りは妥当なインタフェースである。なぜならば、SQL 言語を介する限り言語のインタプリケーションが必要であり、これは PC 側で実行するよりもホストの高速なマシンで実行した方が応答時間の面で有利と考えられるからである。SQL 言語を介さず MMI から要求された検索条件を直接検索マクロに変換する方式が実現できれば、さらに処理の分散を進めホストの負荷を軽減することができる。

3.2 データベースの分散

3.2.1 設計方針

DB をホストと PC とで分散して持つ場合、PC 側ユーザへの見せ方として次の 2 通りがある。

①ホストと PC の DB とを合わせて 1 DB と見せる。

②ホストと PC の DB を別々の 2 DB として見せる。

さらに方式②には次の 2 方式がある。

②-1 ホストのデータは PC のデータの写しとする。

②-2 PC のデータはホストのデータの写しとする。

①はホストと PC 間でいわゆる「分散 DB」を構成する考え方である。すなわち DB の物理的な分散を PC 側のユーザに見せない方式である。

この方式は、

(i) つねにホストと PC とがオンライン接続されていることが前提であり、PC スタンドアロンの使い方ができない。

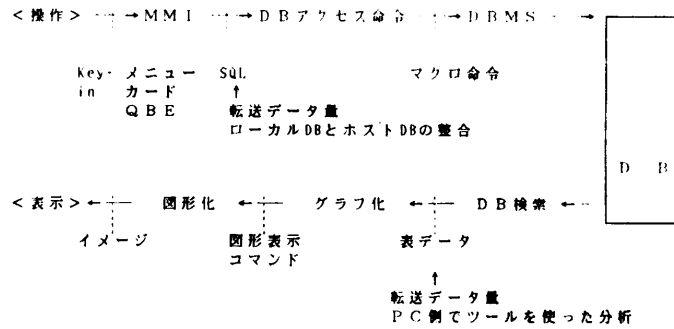


図 2 DB 検索～グラフ表示のモデル
Fig. 2 DB query and display model.

(ii) 現状の 16 ビット PC では分散 DB を実現するためのオーバーヘッドに耐えられない。という判断から②を採用する。

また、一元化されたデータの (共通データ) 管理の面から②-2 の方式を採用する。

②-2 方式で分散された垂直分散 DB を更新する方式として以下の二つがある。

① PC 側データを更新すれば、自動的に対応するホストのデータを更新する (同期更新)。

② PC 側のデータを更新した後、PC 側からの操作を契機として、ホスト DB を更新する (非同期更新)。

ホストと接続せず PC 側スタンドアロンでの更新があり得ることから、方式② (非同期更新) を採用する。

3.2.2 実現方式

ホストデータの参照のためのダウンロード機能およびホストデータの更新のためのアップロード機能を設ける。

アップロード/ダウンロード機能は二つの DB (転送先の DB と転送される DB) とを合併して一つの DB とする操作である。したがって、アップロード、ダウンロード機能では、次の 3 モードを設ける。

① INSERT モード: 転送先 DB のレコードに転送レコードを単純に追加する。

② UPDATE モード: 転送先 DB のレコードの中で指定されたレコードだけを転送レコードで置き換える。

③ MERGE モード: ①② の処理を併せて実行する。

3.2.3 評価—同期更新の必要性について

主として PC 側処理の簡単化の理由で非同期更新方式を採用した。この方式は、① PC 側、ホスト側合わ

表 2 垂直分散 DB における更新
Table 2 Classification of updating distributed DBs.

| 更新主体 | 更新契機 | |
|-----------|------------------------|----------------------|
| | 一 括 | 随 時 |
| 特 定 者 | 文 献 検 索 | 株 価 情 報 |
| 不 特 定 多 数 | 経 営 管 理 (売 上 げ 情 報) | バ ン キ ン グ シ ス テ ム |

せた DB 全体の更新契機が一括でその時刻が決まっていること、および、②レコードに対する更新主体が一つであること、の2条件が成り立つ場合にのみ、DB 全体の無矛盾性が保持される (表 2)。

4. 拡張性

PC を巡る各種ハードウェア、ソフトウェアの進歩は激しく、新製品が次々と発表されている。これに伴い、PC を用いた OA システムは、多くのベンダの多数のハードウェア、ソフトウェアが混在したシステムとして構築されるようになってきた。このような状況に対応するため、VGUIDE は、多様なベンダの製品と連動ができる拡張性の高い構成の製品とした。

4.1 VGUIDE における拡張性向上策

4.1.1 拡張依存項目の考察

VGUIDE プログラムはホスト部分と PC 部分に分かれるが (図 2)、進展の激しいのは特に PC 部分であり、拡張性に対する配慮は以下に示す 5 要因について検討が必要である。

(1) PC 機種に関する拡張依存項目

VGUIDE は、MS-DOS 上での走行を前提とし、ここで共通的に提供されている MS-C を記述言語としている。このため、言語仕様、システムコールについては、拡張に伴う問題はない。しかし、テキスト VRAM の扱いやキーボード等の入出力制御 (BIOS) インタフェースに関しては、各機種により仕様が異なるため、拡張依存項目としてプログラムの局所化を図っている。

(2) DBMS に関する拡張依存項目

SQL 標準化の進展により RDBMS については、DB 命令レベルでの製品間の相違は少なくなっている。しかし、DB 検索の結果データの形式、SQL コード (リターンコード) の体系、NULL 値の表現法等については、製品間に相違があるため、これらの相違をカレントリストと呼ぶ中間ファイルにより共通処理部の中で局所的に吸収する構成とした⁶⁾。カレントリストは、物理構造の異なるホストの DB と PC 上の DB を仮想化し、共通のデータ構造として見せるためのも

のである。VGUIDE は MMI 部と共通処理部との分離により、他の DBMS のサポートに対して拡張性の高い構成となっている。

(3) ホスト接続に関する拡張依存項目

ホストとの通信の確立・開放等の通信手順、終了処理 (LOG-ON, LOG-OFF)、ホストのコード体系は、接続するホストと通信環境により異なるため、この処理は共通処理部の中で局所化している。特に、コード体系については、PC 側はシフト JIS コードにより事実上の産業標準が形成されているが、メインフレーム系については各機種ごとに異なっているのが現状であり、現在の VGUIDE のバージョンでは、JIS コードでホストとのデータの授受を行い、シフト JIS への変換を PC 側で行う方式を採用している。このコード変換処理も共通処理部の中のコード変換部に局所化しており、今後の動向を見て各種の拡張が可能な構成としている。

(4) MMI に関する拡張依存項目

VGUIDE では (2) のカレントリストの採用等により、新たなプログラムの追加だけで新 MMI の追加が可能となる構成となっている。

(5) 市販ソフトウェアとの統合利用

MS-DOS 管理下の実行形式プログラムファイルをメニュー (5 章) から起動出来る機構を設けている⁷⁾。この機構により新たな市販ソフトウェアとのインタフェースを採る必要が出た場合でも、対応可能である。また、市販ソフトウェアとのデータの引渡しについては、CSV 形式⁸⁾のファイルをベースに、ファイル変換機能により、テキスト形式、SYLK 形式⁹⁾による入出力を可能としている。このファイル変換機能は局所化されており、今後、上記の 3 形式以外のファイル形式が必要となった場合も拡張容易な構成となっている。

4.1.2 局所化の方法

VGUIDE では、前項に述べた各拡張依存項目に対して、手入れが必要となる箇所を局所化し、新たに拡張が必要となった場合はこの箇所のみ追加作成すればよいような構成としている。一般に、このような局所化の方法としては、以下の手法が採用される。

① 拡張依存部分を別ファイルとして分離しておきリンク時に結合する方法 (方法 1)

② システム記述言語によるコンパイル制御行を利用して拡張依存部分を分離する方法 (方法 2)

VGUIDE では、プログラムのモジュール化にあたり、以下のような 3 層のレベルを定め、上記の手法を

組み合わせて拡張性向上のための局所化を実現している。

- (i) レベル 1: 拡張に依存する部分がないモジュール
- (ii) レベル 2: 拡張に依存する部分のモジュール内に散在するもの
- (iii) レベル 3: 拡張に依存する部分がモジュール内の大部分に渡るもの

拡張依存項目のインプリメントにあたっては、まずレベル 1 のモジュールを切り出し、さらに残りをレベル 2 のモジュールとレベル 3 に分割している。各モジュールは別ファイルとして切り出し (方法 1)、レベル 2 のモジュールに関してはコンパイル制御行 (方法 2) を利用し、局所化を図っている。

以下に画面表示機能を取り上げ、上記の考え方にもとづく局所化の具体例を述べる。

図 3 は拡張に依存しない関数を示している。関数 `disp()` はスクリーンの左上端から文字を表示するものである。画面出力のための 1 次バッファとして機能する仮想画面に書き込まれたデータは、仮想 VRAM を経由して実画面に書き込まれるが、機種により異なる実画面の仕様差はこの仮想 VRAM で吸収している。つまり、関数 `awrefresh()` が拡張依存部を吸収している。

図 4 は拡張依存部分を含む関数 `awrefresh()` を示している。この中で関数 `pmove()` が実際に実画面に書き込む処理を行っている。この関数 `pmove()` の処理は完全に機種ごとに異なるレベル 3 の関数であり、別ファイルとして分離されている。また、関数 `awrefresh()` は機種ごとに異なる関数 `pmove()`

```
disp()
{
    WINDOW *w, *newwin();
    char *string = "VGUIDE";
    w = newwin(0, 0, 80, 25, 0);
    wclear(w);
    wcsrnmv(w, 0, 0);
    waddstr(w, string);
    wrefresh(w);
}

(1) 関数 disp():

wrefresh(w)
WINDOW *w;
{
    mvtoVir(w);
    awrefresh();
}

(2) 関数 wrefresh():
```

- *1 仮想画面: 画面出力のための横幅と高さを持った一時バッファ。
- *2 仮想カーソル: 仮想画面座標系を移動する論理的なカーソル。
- *3 仮想 vram: 機種による実画面の仕様差 (vram の構造、画面の大きさ) を吸収するための仮想的な実画面

図 3 拡張に依存しない関数

Fig. 3 Functions independent of expansions.

表 3 局所化効果の例
Table 3 Examples of localize effects.

| 拡張依存項目 | | 局所化効果率 |
|--------|------|--------|
| パソコン機種 | 画面表示 | 0.018 |
| | キー入力 | 0.004 |
| DBMS | | 0.072 |
| センタ接続 | | 0.059 |

を含む部分と、機種に依存しない部分とからなるレベル 2 の関数である。

4.2 拡張性の評価

(1) 拡張依存項目の局所化比率

プログラムの局所化効果率は以下のように示される。

$$\text{局所化効果率} = a/(a+A)$$

a : 拡張非依存部分 (レベル 1) 規模

A : 拡張依存部分 (レベル 2, 3) 規模

VGUIDE の各拡張依存項目に関する局所化効果の例は表 3 のとおりであり各拡張依存項目に対して数%の改造で拡張に対応できる構成となっている。

(2) 市販流通ソフトウェアとの統合利用

VGUIDE では、ファイル変換機能により、数値・文字情報を扱う主要なビジネスパッケージのほとんど (売上ランキングベスト 10 の約 9 割) が統合可能となっている。

5. AP 構築高水準言語

5.1 設計方針

MML 環境における AP 構築のための言語は、PC 側とホスト側にそれぞれ必要である。

各々の設計方針を以下のとおりとした。

```

awrefresh()
{
    int n;
    int y = 0;
    register char *chgl = _chgl;
    register char *chgr = _chgr;
    int x, y;
    n = LINES; /* lines to refresh */

    #ifdef BS21
        getxy( &x, &y );
    #endif
    while( n-- ) { /* lines to loop */
        if ( *chgl < *chgr ) {
            #ifdef PC98
                pmove(_y[y] + *chgl, _attr[y] + *chgl,
                    *chgr - *chgl, *chgl + y * COLS);
            #endif
            #ifdef BS21
                pmove(_y[y] + *chgl, _attr[y] + *chgl,
                    *chgr - *chgl, *chgl, y);
            #endif
        }
        chgl++;
        chgr++;
        y++;
    }
    #ifdef BS21
        _locate( x, y );
    #endif
}

```

注：PC9800とBS21の相違を#ifdefで局所化

図 4 拡張に依存する関数

Fig. 4 Functions depend on expansions.

①PC 側にサービスと処理の分離によるメニュー機構を中心とした非専門家向きプログラムレスシステム構築機構を設ける。

②ホスト側にも GOTO レスの構造化簡易言語を用意して、ホスト大規模共用データの業務処理に対処できるようにする。

③PC 側とホスト側の言語仕様を一致させることにより、PC でホスト処理のデバグとプロトタイピングを可能とする。

5.2 実現方式

(1) プログラムレスの業務システム構築

OA における定型業務処理をサービスと処理に分離して考える。

サービスは、メニュー項目として画面エディタで作成し、メニュー項目に対応させてその処理を構成する機能要素（コマンド群および、機能プログラム群）を記述することにより、従来の意味でのプログラミングの不要な業務システムの構築を可能とする。

このようなメニュー制御機構は図 5 に示すように、メニュー定義機構とメニュー実行制御機構および、処理機能要素から構成される。

機能要素としては、通信、DB アクセス、各種関数、実行制御（手続き）コマンド、帳票出力、市販 AP（グラフ出力等）等を用意した。

メニュー制御機構の動作は以下のとおりである。

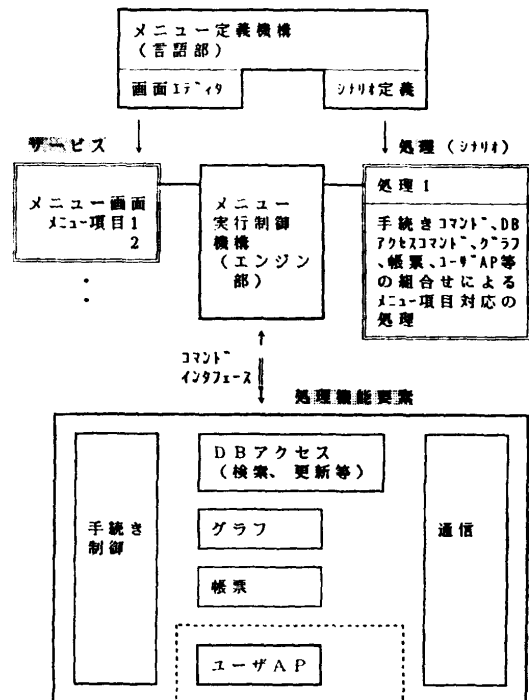


図 5 メニュー制御機構の構成

Fig. 5 Organization of menu mechanism.

まず、メニュー定義機構が提供する画面エディタにより、メニュー画面上にサービスを定義する。次に、シナリオ定義機能により、各サービスの処理部を処理機能要素の中の各コマンドおよび、プログラムを組み

合わせてシナリオとして定義する。

②実行時は、メニュー実行制御機構がメニュー上で選択されたサービスをシナリオに従って機能要素を呼び出しながら実行していく。

(2) 構造化言語

ホストに DB アクセス、実行制御等のコマンドを組み合わせて業務を記述できる簡易言語を用意する。この簡易言語は、表 4 に示すようなコマンドから構成される構造化言語である。

簡易言語の実現形態として、インタプリティブ方式とコンパイル方式がある。ここでは、AP 開発の容易性に着目してインタプリティブ方式を採用した。インタプリティブ方式の難点は、実行性能の悪さにある。そこで、言語のインプリメントにあたっては、SQL 文を除く実行命令はインタプリティブに実行するが、性能への影響の大きい SQL 文についてはコンパイル方式を採用し、DB アクセスサブルーチンを登録することにより、処理の性能を向上させる方式を採った。これにより、DB アクセス部のトータルの処理性能 (DS: ダイナミック ステップ) を 2~3 割改善することができた。

(3) プロトタイピングの容易さ

OIS の構築においては、PC 上での MMI が重要である。そこで、はじめから MML 形態でホスト上に大規模システムを構築するのではなく、まず、スタンドアロン状態で PC 上に小規模システムを構築し、十分ユーザの意見を吸収したのちに MML 形態に移行し、ホスト上に大規模システムを構築するといった方法が考えられる。移行の形態は、データと同時に処理も移

行する場合と、データのみ移行する場合の 2 通りがある。

処理の移行を可能にするためには、PC 側のシナリオ部分とホスト側簡易言語の言語仕様の統一が必要である。VGUIDE の場合、ホスト、PC とも DB アクセスは国際標準化の進められている SQL 文を採用し、実行制御文、関数、変数には統一した言語仕様を採用することにより、ほぼ 9 割以上の仕様を統一することができた。

また、データの移行の場合、一部のデータ (NULL 値、漢字) の扱い、一部の演算子 (不等号) の違いなどホスト、PC 間の SQL 仕様の若干の違いを解決する必要がある。これは仕様を統一して内部で変換することも可能であるが、将来、国際標準 SQL に統一されるであろうことおよび、その違いは小さいことを考慮し、現状では、違いをそのままユーザに意識させることとした。移行を容易にする手段として、SQL 文の先頭に # (ホスト) と * (PC) を付与することにより、対ホスト、対 PC を区別し、PC 側プログラムで処理をふりわける工夫をしている。

5.3 評 価

(1) 言語の生産性

帳票出力、グラフ出力を含む約 200 種類のエンドユーザ向けの検索、更新業務からなる社内のある局所の経営数値情報システムを VGUIDE を用いて構築した。このシステムの構築は、プログラミングに不慣れた非専門家が行ったにもかかわらず、従来の COBOL 等の第 3 世代言語を使って専門家が作成する場合に比べ、完成までの期間、工数を 5~6 分の 1 にすることができた⁹⁾。

(2) MMI の妥当性

「操作のわかりやすさ」に関するオピニオン試験の

表 4 簡易言語の主なコマンド
Table 4 Main commands of VGUIDE.

| 分類 | 主なコマンド | 機能概要 |
|-------|-----------|------------|
| SQL 文 | SELECT | 検索 |
| | UPDATE | 更新 |
| | INSERT | 追加 |
| | DELETE | 削除 |
| 実行制御文 | IF | 条件分岐 |
| | WHILE | |
| | CASE | |
| 関数 | SGETCOL | カラムデータ値取得 |
| | REPCOL | カラムデータ値書込み |
| | ERROR | エラー情報 |
| 変数 | INTEGER | 整数 |
| | DECIMAL | 小数 |
| | CHARACTER | 文字 |
| 代入文 | | 四則演算式 |

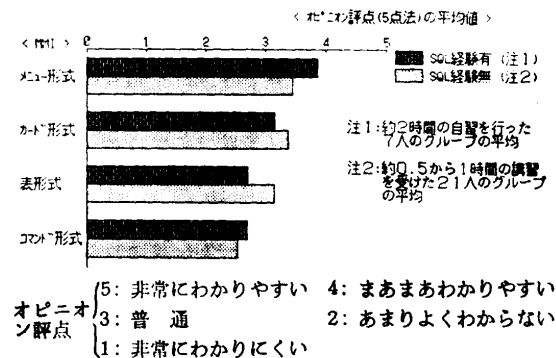


図 6 各 MMI の操作についてのオピニオン試験結果
Fig. 6 Results of opinion test for MMI's manipulation.

結果をまとめると、図6のとおりである。

これによると、操作のわかりやすさは、SQLの知識のあるなしにかかわらず、メニュー形式、カード形式、表形式、コマンド形式の順であり、利用者の階層構成に応じたMMIが実現できているといえる。

6. おわりに

本論文ではPCを用いた垂直分散型オフィス情報システム向けのデータベース簡易言語VGUIDEを構成する技術について述べた。VGUIDEは現在社内のOAシステム等で利用されている。今後はこれらのユーザの要求を吸収していくとともにマルチメディアデータの操作機能の追加等を進める予定である。

謝辞 本研究を進めるにあたってご指導頂いたNTT情報通信処理研究所石野所長、山川情報通信処理方式研究部長他関係各位に深謝いたします。

参考文献

- 1) Jarke, M. and Vassiliou, Y.: A Framework for Choosing a Database Query Language, *Comput. Surv.*, Vol. 17, No. 3, pp. 313-340 (1985).
- 2) 高木, 川合: 高機能インターフェイスの現状と展望, 情報処理学会, マルチメディア通信と分散処理研究会, 32-1 (1987).
- 3) Martin, J.: *Fourth Generation Languages*, Vol. I, Prentice-Hall, Inc., Englewood Cliffs, NJ (1985).
- 4) Shindo, J., Tenmizu, N., Sakamoto, K. and Kawate, H.: Database End-User Facilities for Office Automation, IEEE, Computer Society Office Automation Symposium, April (1987).
- 5) 田中 豪, 戸田 博, 池田哲夫: 実時間型リレーショナルデータベース管理システム, 通研実報, Vol. 35, No. 7, pp. 669-676 (1986).
- 6) 黒川裕彦: MML指向の分散DBシステムの実現方式, 第35回情報処理学会全国大会論文集, 6 Cc-2, pp. 445-446 (1987).
- 7) 川手 寛, 黒川裕彦: 定型的DBアクセス業務向けメニューシステムの実現について, 第33回情報処理学会全国大会論文集, 3 H-5, pp. 833-834 (1986).
- 8) 古川真二: 異種アプリケーション間のデータ交換を可能にする共通フォーマットCDF, *Nikkei Byte*, July 1985, pp. 101-107 (1985).
- 9) 進藤: 垂直分散型オフィス情報システムの構成法, 情報処理学会マルチメディア通信と分散処理研究会, 38-3 (1988).

(昭和63年4月15日受付)

(平成元年1月17日採録)

進藤 重平 (正会員)



1947年生。1971年早稲田大学電気通信学科卒業。同年日本電信電話公社入社。NTT情報通信処理研究所を経て、現在NTTネットワークシステム開発センタ主幹技師。オフィス情報システムの研究実用化に従事。電子情報通信学会、IEEEコンピュータソサエティ各会員。

天水 昇 (正会員)



1942年生。1967年鹿児島大学電気工学科卒業。同年日本電信電話公社入社。現在NTT情報通信処理研究所主任研究員。主にコンピュータシミュレーションプログラム、データベースエンドユーザ言語、オフィス情報システムの研究実用化に従事。電子情報通信学会会員。

坂本 恵市 (正会員)



1951年生。1974年早稲田大学電子通信学科卒業。1976年同修士課程修了。同年日本電信電話公社入社。主に、データベース管理システム、データベースエンドユーザ機能、マルチベンダ・インテグレーション・アーキテクチャの研究実用化に従事。現在、NTT情報通信処理研究所主幹研究員。電子情報通信学会会員。

川手 寛 (正会員)



昭和26年生。昭和49年愛媛大学工学部電気工学科卒業。同年日本電信電話公社入社。現在NTT情報通信処理研究所主任研究員。主にデータベース管理システムと、その言語処理系、およびOAシステム構築支援パッケージの実用化に従事。

黒川 裕彦 (正会員)



昭和33年生。昭和56年福井大学工学部電子工学科卒業。同年日本電信電話公社入社。現在NTT情報通信処理研究所研究主任。主にリレーショナルデータベース管理システムおよび、データベース応用パッケージの研究開発に従事。