

垂直分散型オフィス情報システムの構成法

進藤 重平

N T T 情報通信処理研究所

ワープロ、パソコン、FAXなどの導入から始まったオフィス業務の機械化は、経営管理支援や意志決定支援を狙う、データベース中心のオフィス情報システムへと進展してきた。

一方、最近のマイクロプロセッサ技術及び記憶媒体技術の進歩により、高機能かつ高性能なパソコン及びそれに接続できる大容量記憶装置が安価に入手可能となってきた。

この様な背景の下で、オフィス情報システムはホスト計算機に多数のパソコンが有機的に接続された、いわゆる垂直分散型を取るものが主流になりつつある。

本論文は、この様な垂直分散型オフィス情報システムを実現する上で解決すべき次の2つの技術的問題について論じている。

- ① ユーザ毎に作成が必要な応用プログラムの構築を支援する簡易言語を、垂直分散環境の中でどう実現するか。
- ② ホストとパソコン間で垂直分散されたデータベースをどう取り扱うか。

第一の問題については、オフィス情報システムを迅速かつ経済的に構築するための簡易言語の具備すべき条件として、①端末とホスト間でのAP共用(流通性)、②端末とホスト間での情報流通のための機能、③端末においては既成プログラムとの連動を実現するメニュー機構が重要であることを述べた。

さらにこれらの考え方に基づく簡易言語を構築し経営数値情報に関するオフィス情報システムに適用してみた結果、既存言語のみで構築するのに比べ、プログラム経験の無い要員でも可能でありかつ、4~6倍程度生産性向上がはかれることが判った。

また、第二の問題については垂直分散データベースの使用形態を整理すると共に、部門別計算機導入の広がりに伴い、垂直分散データベースの管理方法について現実的な解を求める必要性を述べた。

Technical Problems concerning Micro-Mainframe-link type
Office Information Systems

Jyuuehi Shindoh

NTT Communications and Information Processing Laboratories

1-2356, Take, Yokosuka 238-03 JAPAN

Technical problems concerning Office Information Systems(OIS), which consists of personal computers(PCs) and a host computer. The problems are,

- how to construct higher level languages in micro mainframe link environments
- how to distribute databases between PCs and a host

The higher level languages for OIS are provided both in PCs and in a host. By using these languages, it takes only 1/4-1/6 efforts of programming in COBOL or C to construct an OIS including DB queries and report generation.

Among the second problems, how to update distributed DBs between of PCs and of a host is the most interesting. Specially, the introduction of departmental computers will make these problems more urgent.

1. はじめに

ワープロ、パソコン、FAXなどの導入から始まったオフィス業務の機械化は、経営管理支援や意志決定支援を狙う、データベース中心のオフィス情報システムへと進展してきた。

一方、最近のマイクロプロセッサ技術及び記憶媒体技術の進歩により、高機能かつ高性能なパソコン及びそれに接続できる大容量記憶装置が安価に入手可能となってきた。

この様な背景の下で、オフィス情報システムはホスト計算機に多数のパソコンが有機的に接続された、いわゆる垂直分散型を取るものが主流になりつつある。

本論文は、この様な垂直分散型オフィス情報システムを実現する上で解決すべき次の2つの技術的問題について論じる。

- ① ユーザ毎に作成が必要な応用プログラムの構築を支援する簡易言語を、垂直分散環境の中でどう実現するか。
- ② ホストとパソコン間で垂直分散されたデータベースをどう取り扱うか。

第一の問題については、簡易言語の持つべき機能と試行システムの適用結果を報告する。第二の問題については、垂直分散データベースの使用形態を整理し、今後解決すべき問題を提起する。

2. 垂直分散型オフィス情報システムの例

垂直分散型オフィス情報システムの一例として、簡易データベース検索システム (VGUIDE) を取り上げる⁽¹⁾⁽²⁾⁽³⁾。

VGUIDEは図1に示すように、ホスト系とパソコン(PC)系から構成され、それぞれがデータベース管理システム(DBMS)を持つ垂直分散型の構成となっている。

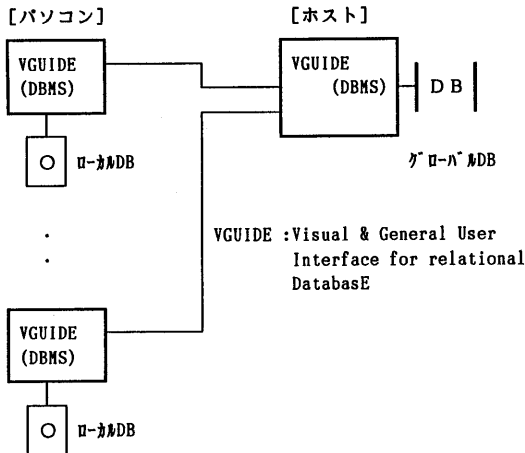


図1 垂直分散型オフィス情報システムの構成例

3. 問題は何か

3.1 応用プログラム (AP) 構築支援簡易言語

オフィス情報システムに限らず、開発すべきソフト量に比べて開発可能なソフト量が追いつかず、バックログを生じていると云う、いわゆる「ソフトウェア危機」に直面している。この現象は開発量の大きい応用プログラム (AP) に於て特に顕著である。

この様な状況を打開するため、「第4世代言語 (4GL)」と呼ばれる新しいシステムが80年代に入って提案され、商品化も進められてきた⁽⁴⁾。4GLはソフト生産性を上げると共に、ソフトウェアの読解性を高めてプログラム作成者の裾

野を広げることにより、バックログの解消を狙うものである。

従来の4GLは、バッチシステムやホスト中心型のオンラインシステム—即ち、インテリジェンスの低い端末がホストに接続されるシステム—には適用できる。しかしパソコンのようにインテリジェンスの高い端末がホストに接続される、垂直分散型システムの環境では、その効果を発揮しない。

そこで、垂直分散型システムの環境 (MML環境) でのAP構築支援用の簡易言語をいかに構築するかが第1の問題である。

3.2 垂直分散データベースの管理

地理的に分散したデータベースをネットワークで接続し、ユーザからはあたかも1つのデータベースであるかのようにみせる、分散データベースについてはすでに種々論じられ、製品化が始まっている⁽⁵⁾⁽⁶⁾。しかしMML環境においては、分担する役割の異なるホストとパソコンの間でのデータベース分散を実現する必要がある。この問題は、従来から研究されてきた分散データベースのそれとは以下の点で異なる。

- ① パソコンのデータベースとホストのデータベースとの間でデータの重複がある場合が多い。
- ② 両データベースが非同期に更新されることがある。
- ③ ホストとパソコンではアーキテクチャも処理能力も異なる。

この様な特徴を持つ、いわば垂直分散データベースをいかに管理・操作するかが第2の問題である。

4. AP構築支援簡易言語

本章では、オフィス情報システムに現れるMML環境を分類し、各々において簡易言語の具備すべき機能を考察する。その結果として、ホストとパソコンとが協同して動作する業務を簡易に構築できる、簡易言語を提案する。

さらにこの様なコンセプトに基づいて実際にインプリメントしたシステム例を紹介し、具体的なオフィス情報システムの構築に於てどのくらい効果が在ったかを示す。

4.1 垂直分散の形態と必要な簡易言語機能

(1) 最も単純な形態——ダム端末エミュレータ

MML環境に於て、ホストとパソコンとのもっとも単純な結合形態は、エミュレータ方式である (図2)。

これはパソコンの上にホストのダム端末を模擬するプログラム (端末エミュレータ) を作り込むもので、ホストからはパソコンが他のダム端末とまったく同等に見える。

この様な形態のMML環境に於て、パソコンとホストとが協同するために必要なメカニズムは、以下の通りである。

- ① 回線接続・切断機能、通信制御機能
- ② パソコンからホストプログラム (ホストの簡易言語で書かれたプログラム: コマンドプロセッサを含む) の起動機能

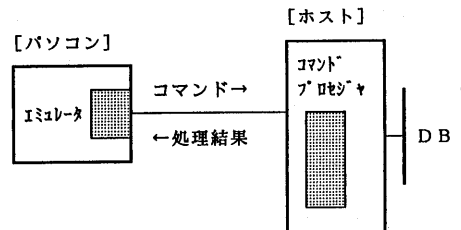


図2 端末エミュレータ

(2) ファイル転送型——ダウンロード/アップロード

次に一段複雑なMMLシステムはファイル転送型MMLである(図3-1、図3-2)。この形態は、端末のパソコンとホストとにファイル転送(データベース転送)機能を作り込むものである。ホストコンピュータのファイル(DB)を端末のパソコンに転送してそこで表示、計算、シミュレーション等の加工を施す。逆に、端末の表計算プログラムを使ってエントリしたデータをまとめてファイル(あるいはDB)単位でホストに転送し保管する。この形態では、ファイル(DB)転送時のみホストと端末とが接続され、ホストから端末に転送されたファイルの内容を端末側で加工したり、端末でデータをエントリしてファイル(DB)にため込んでいる最中は、端末とホストは切り離されている。

この様な形態のMMLシステムにおいて、端末とホストとが協同するために必要なメカニズムは以下の通りである。

- ①ファイル転送機能
- ②DBダウンロード、アップロード機能

①と②の相異は、①が単なるデータファイルの転送であるのに対し、②はデータファイルの他、DBのスキーマ情報ファイル等1つのDB(テーブル)の転送に対して複数のファイルを一括して転送する点異なる。DBのダウンロード、アップロード時にはソースDBからのデータ選択、デスティネーションDBへのデータの追加・更新方法等の機能が、ファイル転送に比べると、必要となる⁽¹⁾⁽³⁾。

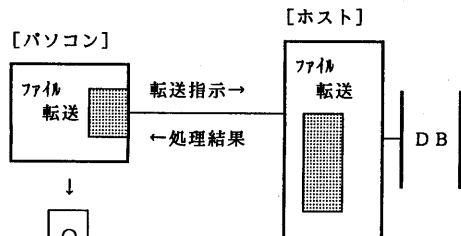


図3-1 ファイル転送型MML (データの加工・分析)

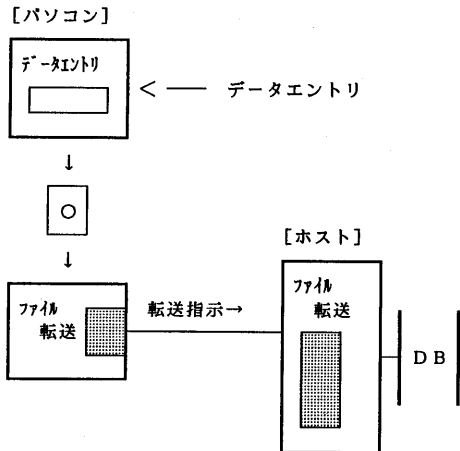


図3-2 ファイル転送型MML (データエントリ)

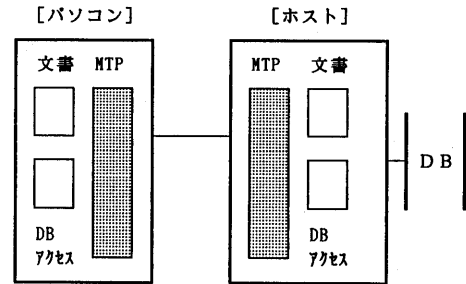
(3) 統合型——メッセージ/データ転送

最も複雑で、端末とホストとが強く結合された形態が統合型のMMLシステムである(図4)。この形態は端末~ホスト間のメッセージ(MSG)転送機能をベースに、メール、文書、データベースアクセス等の機能を端末とホストに作り込むのである。

この形態のMMLシステムでは、端末の利用者は自分の要求した処理が端末とホストのいずれで実行されるかについて意識しないケースもあり得る。

この様な形態のMMLシステムにおいて、端末とホストとが協同するために必要なメカニズムは以下の通り。

- ①MSG転送機能
- ②MSG転送をベースとしたメール、DBアクセス、文書通信等の高位プロトコル処理機能



MTP:メッセージ転送

図4 統合型MML

4.2 簡易言語構成法の考え方

(1) 利用者は誰か

オフィス情報システムを構築する際考えるべき関係者としては次の3種類ある。

- ①システム構成要素の開発者とこれらの統合を担当する者(製品開発者とインテグレータ)
- ②システムを利用して具体的なオフィス情報システムを構築するサービス提供者
多くの場合システムの運用・保守をも担当する。
- ③オフィス情報システムを日々の業務で利用するエンドユーザ(端末利用者)

サービス提供者向きには専門家向き、手続きの、従来言語に比べて高生産性等の特徴を持つ簡易言語が求められよう。一方エンドユーザ向きには、素人向き、非手続きの、視覚的等の特徴を持つ簡易言語が求められる。

(2) AP流通の観点から——アーキテクチャの統一

簡易言語で記述されたAPの流通、共用性の観点から、サービス提供者向き言語は、端末の簡易言語とホストのそれとはできるだけ仕様が似かよっていることが望ましい。その理由は「AP開発の容易さ」である。

従来は、マイクロプロセッサ上のソフト開発を効率化するため、大型機ホストに対象マイクロプロセッサのクロスコンパイラを開発し、これを用いてデバッグした後、最終確認の為にターゲットマシンに出来上がったプログラムをダウンロードし、実行させていた。しかし現在では(ア)統一な環境(例えばMS-DOS)をもったパソコンが広く普及したこと、(イ)パソコン上の言語コンパイラやデバッグツールが整備されてきたこと等の理由から、パソコン上のプログラムはほとんどの場合ターゲットマシン上で開発される様になってきた。

さらに今後の傾向としては、上の(ア)、(イ)がますます顕著になると予想されるため、大型ホストのプログラムも

逆にパソコン上で開発し、デバッグ終了後ターゲットマシンであるホストにアップロードして実行させる様になって来る。

この様な傾向を踏えると、簡易言語によるプログラミングであっても、端末側とホスト側の言語仕様をできるだけ一致させておくことにより端末側で開発・デバッグし、最後にホストへアップロードするという開発方式がとれる様にしておくことが望ましい。

この考え方は最近各メインフレームから発表されている「アプリケーション・アーキテクチャ」の基本の1つである。

(3) 業務拡張の容易さ——パソコンスタンドアロンから MML へ

(2) で述べた A P 開発手順はそのまま O A におけるシステム拡張の手順でもある。即ち、まずパソコンスタンドアロンで個々の業務の O A 化を図り、ある程度軌道に乗り、効果が出はじめた段階でホストコンピュータを導入して、データの共用や処理能力のアップを目指すと言うものである。

この際初期に作成した A P やデータが、拡張された段階でもそのまま使用できると言うことが、スムーズな拡張に有効である。

4. 3 端末側の簡易言語

前2節で述べて来た様に端末における簡易言語には、①エンドユーザが直接利用する、②サービス提供者が利用し、各エンドユーザに提供する という2面あることが判る。さらに③既存の、特に市販流通するソフトパッケージを活用したい要求も強い。

端末においてこれら3種類の A P を統合し所要の業務を遂行する機構が必要となる。即ち、

①エンドユーザ用言語の具体例としては、データベースに対する表検索、カード検索、端末へのレポートジェネレータ等視覚的な言語が有力である。

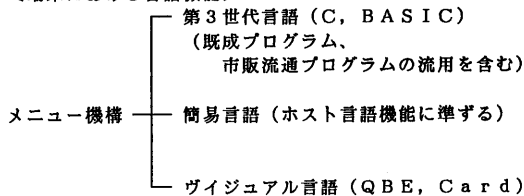
②サービス提供者が提供する簡易言語の具体例としては、次節で述べるホスト用簡易言語の仕様に準じた言語がある。

これは世の中の 4 G L と類似のものである。

③ C や B A S I C 等で記述された、あるいはオブジェクトのみの既存ソフトとの連動については、データの引渡しポイントである。これは最近かなり自由に行えるようになってきた。

これら3種類の A P を、統一した M M I の中で起動し得る機構として、エンドユーザのメニュー選択によってこれらを使い分けるメニュー機構がある⁽⁷⁾。

<端末における言語機能>



4. 4 ホストにおける簡易言語

ホストにおいて業務を記述するための言語としては、通常の 4 G L の仕様と類似したものが必要である。即ち、

- ①外部ファイル (DB) へのアクセス、ファイルの管理
 - ②プログラムの流れの制御 (WHILE, IF THEN, ELSE etc.)
 - ③資源の管理 (キュー制御、エリア制御等)
- 以上に加え MML の形態をとるシステムでは、
- ④通信制御 (回線の起呼・切断、プログラムの起動)

⑤ファイル、DBの転送

⑥メッセージの転送およびメール、DBアクセス、文書通信コマンドの転送
に関するステートメントが必要となる。

4. 5 適用の効果例

パソコンとホストコンピュータから構成される MML システムにおいて両者の協同を実現する簡易言語を開発した。この言語は主としてデータベースを中心とするオフィス情報システムに適用するためのものである。(表1、付表)

ホストのデータベース中の経営数値情報を検索し、その結果を予め定義した帳票の形で端末に出力するとともに、グラフ表示する業務を、この簡易言語で表現(記述)することができる。

この様な業務 200 余りからなるサービスを、10~20人が3か月程度で構築し稼働させた例もある。同様のことを既存言語のみで実現しようとするとも1年~1年半はかかる。

表1 簡易言語の適用結果例

事例	A (人年)	B (人年)	C (人年)
既存言語 (X)	20	31	34
簡易言語 (Y)	3	6	8
生産性比 (X/Y)	6	5	4

具体的な適用事例を付表に示す。付表のデータ及び適用した経過から次の事が判った。

- ①同等の A P を C O B O L または C で作成する場合に比べ、1/4~1/6の稼働で作成できる。
- ②ほとんどプログラミング経験の無い要員が、1~2週間の訓練でシステム構築に参加できる。
- ③事例 A はホスト、パソコン両方、事例 B はホスト中心、事例 C はパソコン中心の簡易言語をそれぞれ使用した。3事例間で生産性の向上率には大きな差は出なかった。しかし、ホスト簡易言語を使ったプログラミングには、ホスト計算機に関する知識や操作が必要なため、この面でスキルを持った要員を集めた。一方、パソコンの簡易言語では特別な知識が不要なため、全く計算機の経験の無い要員にもプログラミングが可能であった。
- ④事例 C ではまずパソコンだけでカード・帳票・メニューを作成して試行運用し、その後ホストの共用データベースを組み入れる方式を取った。このため、スムーズなアップグレードが可能であった。

5. 垂直分散データベースの管理

垂直分散データベースの適用例を分析し、各々データベース管理上の問題を述べる。

5. 1 垂直分散データベースの操作例と問題

(1) データエントリ

ホストデータベースへのデータエントリ的手段として、まずオフラインでパソコンのデータベースを更新し、次にホストと接続して、アップロード機能によりホストデータベースを更新するやり方である。この場合、①パソコンデータベースの中でアップロードすべきデータをいかに識別するか、②特に、削除されたデータをどう管理しておくかが問題となる。

(2) サブセッティング

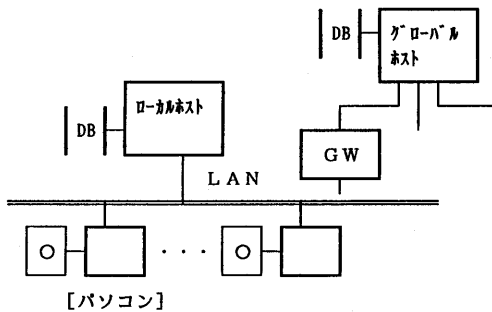
ホストDBの一部をパソコンにダウンロードし、パソコン上でグラフ表示したり、シミュレーションにより分析する。このケースではDBに対して参照のみであるので両DB間の矛盾は発生しない。

(3) アップデート

ホストDBの一部をパソコンにダウンロードし、パソコン上で更新した後ホストへアップロードする。このケースでは、パソコンでの更新中はホストDBの更新を抑制する必要がある、更新が長期にわたる場合（月日のオーダとなる場合がある）、ホストDBに特別な排他制御機構が必要となる。

5. 2 部門別計算機導入時の問題

ホストとパソコンとが2階層で接続された構成では、パソコン側のDBがホストDBの「写し」であると割り切ることにより、両DB間の矛盾は無視することが可能である。しかし、部門別計算機の導入された構成では（図5）、グローバルホスト、ローカルホスト、パソコンの3階層でDBを持つことになる。この構成に於いては、グローバルホストとローカルホストの両DB間の無矛盾性が問題となる。即ち、大部分の処理をローカルホストで済ませ、共通データのみグローバルホストへ転送する。この場合分散DBとして管理すべきか、別の管理方法を取るべきかが問題となる。今後部門別計算機導入の広がりに伴い解決すべき課題である。



GW：ゲートウェイ

図5 部門別計算機構成例

6. まとめ

パソコンとホストコンピュータとが有機的に結合される、マイクロ・メインフレームリンクの形態が、オフィス情報システムの主流となってきた。

この様な動向において、オフィス情報システムを迅速かつ経済的に構築するための簡易言語の具備すべき条件として、①端末とホスト間のAP共用（流通性）、②端末とホスト間での情報流通のための機能、③端末においては既存プログラムとの連動を実現するメニュー機構が重要であることを述べた。

さらにこれらの考え方に基づく簡易言語を構築し経営数値情報に関するオフィス情報システムに適用してみた結果、既存言語のみで構築するのに比べ、プログラム経験の無い要員でも可能でありかつ、4～6倍程度生産性向上がはかれることが判った。

また、部門別計算機導入の広がりに伴い、垂直分散データベースの管理方法について現実的な解を求める必要性を述べた。

参考文献

- (1) J. Shindoh, N. Tenmizu, K. Sakamoto, H. Kawate: Database End-User Facilities for Office Automation, IEEE Computer Society Office Automation Symposium, April 1987.
- (2) 天水昇、坂本恵市、川手寛、黒川裕彦：データベース簡易検索パッケージ、通研実報、VOL.36, No.11, PP.1493-1501, 1987.
- (3) 進藤重平、天水昇、坂本恵市、川手寛、黒川裕彦：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、情報処理論文誌（投稿中）。
- (4) James Martin: Fourth Generation Languages, Volume I, Englewood Cliffs, NJ, Prentice-Hall, Inc., 1985.
- (5) 増永良文：分散型データベースシステム、情報処理、VOL.28, No.4, PP.446-455, 1987.
- (6) 市場形成に向けて製品化始まる分散データベース、日経コンピュータ、1988.5.23, PP.101-116.
- (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講演論文集 3H-5 (1986.3)

付表 簡易言語の適用事例

項目	事例A	事例B	事例C	記事	
業務	<ul style="list-style-type: none"> ・経営情報管理 ・販売状況 ・物件費管理 ・不良施設管理 ・社員録 ・オレンジ情報 ・紙使用量 	<ul style="list-style-type: none"> ・大規模エーサ管理 ・経営管理 ・保全業務 ・投資評価 ・NCC情報 	<ul style="list-style-type: none"> ・顧客管理 ・ビル管理 ・イベント情報 ・NCC情報 ・経営情報 ・オレンジ情報 ・社員住所録 		
期間	61.12~62.3末 (実際の物作り 2~3ヶ月)	同左 同左	62.11~63.3末 同左		
要員	約10名	約20名	約20名		
カード画面(枚)	0	0	約80	パソコン	
帳票画面(枚)	41	0	約40		
メニュー画面(枚)	18	約20	約40		
プログラム	数	約200本	約400本	---	ホスト
	規模	約4KS	約400KS*1	---	
ハード	V30	V30	V30S		
端末	PC98:186台	PC98 :約250台 DT9658:約250台 BS21 :約500台 (未使用)	PC98XL ² :20台		
ローカルDB	無し	無し	有り		

*1:1ヶ月分のコマンドプロセッサを12ヶ月分コピーしているので、
実質1/10の40KSが実製造規模。

【生産性の評価】

- (1) VGカードプログラム規模(15KS)×1/10×
カード枚数
- (2) VG帳票プログラム規模(40KS)×1/10×
帳票枚数
- (3) VGメニュープログラム規模(15KS)×1/10
×メニュー画面枚数

【事例A】

・帳票 : 4KS×41枚 = 164KS
 ・メニュー : 1.5KS×18枚 = 27KS
 ・コマンドプログラム : = 12KS

 計 203KS

COBOL生産性10KS/人年とすると 20人年

【事例B】

・メニュー : 1.5KS×20枚 = 30KS
 ・コマンドプログラム : = 120KS
 ・COBOL-AP : = 160KS

 計 310KS

COBOL生産性10KS/人年とすると 31人年

【事例C】

・カード : 1.5KS×80枚 = 120KS
 ・帳票 : 4KS×40枚 = 160KS
 ・メニュー : 1.5KS×40枚 = 60KS

 計 340KS

C言語の生産性10KS/人年とすると 34人年