

データベースとOA：統合化ソフトウェアの1つの試み

渡辺正親

株式会社 アプリケーションウェア研究所

事務処理のパターンを特にデータベースとの関連において検討するとともに、現在のOAソフトウェアとしての市場性を考慮したコンパクトなデータベース管理システムと事務処理用リポータライタ/ファイルユーティリティを備えた統合ソフトウェアH&L&Nの紹介

A DATABASE IN OFFICE AUTOMATION ENVIRONMENTS

A HOLONIC SOFTWARE HOLON

Masasi Watanabe

Application Ware Inst.

[abstract] Today personal computers are widely populated in almost all types of business offices. However, most softwaers were tailored for certain specific applications, and hence these can deal only with relatively simple classes of business applications. For this reason, a unified system with a database is required. we have developed a holonic system HOLON to accord with such a requirement. It consists of severl end-user languages developed on a compact relational database system. In particular, an end-user language OA-org can deal with almost all patterns of business data processing efficiently. The system specification and its theoretical background are presented briefly.

1 はじめに

パーソナル・コンピュータ i f 8 0 0 の上位機種と、i f 1 0 0 0 のためのソフトウェアとして、データベースを中心に事務処理全般を支援する統合ソフトウェアの開発を行った。

すでにながりの数の O A 用ソフトウェアが商品化されてはいたが、その多くは簡単ないわゆるスプレッドシート型パッケージか、あるいは個別の業務別に開発された特定のソフトウェア群であった。

一方、このクラスの計算機のために開発されたデータベース管理システムも、すでにくつか登場していたが、これらは簡単な顧客管理など検索中心の小規模なアプリケーションに耐え得ても、事務処理全般を支援するにはなお多くの問題を残していた。

事務処理のパターンを、特にデータベースとの関連において深く検討するとともに、現在の O A ソフトウェアとしての市場性を考慮して、最終的にコンパクトなデータベース管理システムと、5 つの異なるタイプのアプリケーション向きのエンドユーザ言語を備えた、統合ソフトウェア H O L O N を開発することとした。

開発は 1 9 8 4 年春から手がけ、1 9 8 6 年春には商品として出荷準備がととのった。

このクラスの計算機のためのデータベース管理システムとして、関係型の採用が妥当であろうと考えたが、既存の関係型データベース管理システムの設計を、そのまま踏襲することが適当であるとは思えなかった。

実際、従来の関係型データベース管理システムでは

- (1) 順序処理の効率が低い
- (2) 事務処理の基本パターンである外結合を十分に支援していない
- (3) 集計演算に制約がある
- (4) 伝票や帳票の編集機能が不十分である

などの致命的欠陥が見られた。このような機能を十分に管理するためには、事務処理用レポートライタ／ファイル・ユーティリティとして、O A . O R G を用意することとしたが、データベース管理システム側でもこの点を十分考慮して、コンパクトかつ効率よい関係型システムの設計を目指す所となった。

データベース管理システムの中核となるデータベース・アクセス・ルーチンは、プログラム言語 C の関数としてつくられた。

データベースの検索に関しては、タプル論理と順序を指定する式を引き数として与え、多様なアクセス方式を許している。タプル論理は 1 つの関係の上で定義されるものに限ったが、親言語インターフェースとしてはこれで十分使用に耐え得る。ロック、リカバリなどの付帯機能は、計算機の規模を考えて必要最小限に留め、オーバーヘッドを軽減した。

エンドユーザ言語としては上述の O A . O R G のほか、スプレッドシート型言語、チャート／グラフの生成機能、図面作成用の簡易 2 次元 C A D 、および日本語ワードプロセッサが用意された。そのほかに検索言語として

SQLとQBEの開発が予定されている。このうち、特に設計に苦心が払われたものにはOA・ORGがあり、これがまた最も特色のあるパッケージとなっている。

OA・ORGは、事務処理用のレポートライタ／ファイル・ユーティリティであるが、論理的には射影、選択および結合演算をおこなうプログラム群となっている。ここで結合は、外結合さらにそれを2つ以上の関係に対するものに拡張した形のもであり、また集約演算が組み入れられている。事務処理のパターンに応じて、リアクティブ処理のための2つと、バッチ処理のため2つの、4個のプログラムから構成されている。

リアクティブ処理用のものは、手軽に台帳を修正するためのプログラムと、伝票を発行しつつトランザクション・ファイルを作ったり、台帳を直接修正したりできるプログラムとなっている。

前者は選択演算、後者はネスト型結合演算が基本となっている。バッチ処理用のものは、伝票あるいは帳票の発行のためのレポートライタと、トランザクション・ファイルを使って台帳を更新するためのプログラムで、前者はネスト型結合が後者はマージ型結合が前提となっている。

特に伝票・帳票の発行にあたっては、多段の集計機能とフレキシブルな編集機能があり、日次作業や会計諸票の作成など効率よく行うことができる。もちろん計算機に習熟していないユーザのために、非手続き型の簡易言語が用意され、伝票・台帳の編集方式やその作成・更新プログラムの指定を容易に行うことができる。

HOLONは、当初MS-DOS/C、次にUNIX/Cの環境下でインプリメントされた。プログラム言語Cが使用できる計算機であれば、HOLONの移植は容易である。

現在までの実験結果では、様々な事務分野について高い効率が得られているが、今後色々な既存のデータベース管理システムを使って、処理効率を比較してみたいと思っている。

以降HOLONのデータベース管理システム、およびOA・ORGについてやや詳細に解説する。

2 データベース管理システム

2.1 データベース定義機能

データベースの定義は、下記の項目について行うこととした。

- (1) ファイル名
- (2) 最大レコード数
- (3) 見出し項目名と形式及び2次索引の有無
- (4) 一般項目名と形式及び2次索引の有無
- (5) 一次キーの重複を許すかどうかの指定

形式としては、整数、実数（浮動小数点形式）の単精度、倍精度、10進浮動小数点数、文字列を許している。

2.2 データベース編成

HOLONのデータベースは、DB1とDB2の2つの部分から構成され

ている。DB1は、データベース・アクセス関数の引き数として与えられた様々なタプル論理式に対して、効率よい検索を保証している部分であり、DB2は、特定のブロック転送によるデータのアクセスを前提とする部分である。

DB2は、例えば伝票・帳票の定義情報など、各パッケージで使われる特定のテーブルや、パッケージ間のデータ転送のために用いられる。いわゆるデータベース・ファイルは、DB1に収容される。各関係のタプルを収容した主ファイルは、1次キーによって区分順編成を行った。

各区分の先頭のタプルに対するB木編成による索引ファイルを用意したほか、関係中の全タプルに対して、1次キーによる直接編成の索引を設けることとした。後者には線形ハッシュ法を採用した。このほかユーザが望めば、指定した2次キーに対してB木編成の索引が作られる。

直接編成の1次キー索引を経由して、主ファイルがアクセスされる。主ファイルの区分、索引ファイルのバケットの大きさなどは、全体の効率に大きく影響する。これを最終的に決定するまでには、何回もの試行錯誤を繰り返した。

2.3 データベース・アクセス関数

C言語のデータベースアクセス関数として以下のものが用意されている。

(1) dbschema

該当ファイルの項目情報を読み込む。

```
p = dbschema( fname , n1 , gname , n2 )
```

p : 該当ファイルの項目情報を読み込んだ領域のポインタ

fname : ファイル名

n1 : 読み込んだ項目数を格納する領域へのポインタ

gname : 将来の可変長項目の取扱の可能性を考慮して設けているが現

n2 : 在未使用である

(2) dbselect

検索の実行。

```
rn = dbselect( name , v_name , inf , seg , t_name , no )
```

rn : 検索条件にマッチしたレコード数

name : ファイル名

v_name : 変数名 (複数ファイル上の検索のために設けているが現在未使用である。)

inf : 検索条件

seg : 読み込み順序条件

t_name : 項目情報

no : 呼び出し番号

検索条件は、

項目名 関係演算子 定数 または 項目名 関係演算子 項目名

なる形式の式、またはそれらを論理演算子でつないだものを指定する。
読み込み順序条件は、

項目名 `ase` または 項目名 `dis`
なる形式で指定する。

(3) `dbreleas`

`dbselect`等の作業領域を解放する。

`dbreleas(no)`

`no` : `dbselect`等で知らされた呼び出し番号

(4) `dbread`

`dbselect`で検索したレコードを1レコードずつプログラム領域に読み込む。

`cursor = dbread(area , no)`

`cursor` : レコード識別子

`area` : レコードを読み込む領域

`no` : `dbselect`で知らされた呼び出し番号

(5) `dbinsert`

レコードを追加する。

`dbinsert(name , t_name , area)`

`name` : ファイル名

`t_name` : 項目情報

`area` : 追加するレコードが格納されている領域

(6) `dbreplace`

レコードの変更を行う。

`dbreplace(t_name , no , area , cursor)`

`t_name` : 項目情報

`no` : `dbselect`で返された呼び出し番号

`area` : 変更するレコードが格納されている領域

`cursor` : `dbread`でかえされたレコード識別子

(7) `dbdelete`

レコードを削除する。

`dbdelete(no , cursor)`

`no` : `dbselect`で返された呼び出し番号

`cursor` : `dbread`で返されたレコード識別子

(8) `dbsllock`

ファイルを共用モードでロックする。

`dbsllock(name , n , no)`

`name` : ファイル名のリスト

n : ロックするファイルの数
no : 呼び出し番号を格納する領域のポインタ

(9) dbelock

ファイルを占有モードでロックする。

dbelock(name , n, no)

name : ファイル名のリスト

n : ロックするファイル数

no : 呼び出し番号を格納する領域のポインタ

(10) dbunlock

dblock,dbelockで宣言したロックを解除する。

dbunlock(no , n)

no : 呼び出し番号のリスト

n : 呼び出し番号数

(11) dbopen

ファイルをopenする。

dbopen(name)

name : ファイル名

(12) dbclose

ファイルをcloseする。

dbclose(name)

name : ファイル名

2.4 データベース・ユーティリティ

データベース・マネジメント機能のうち必要最小限度の機能をデータベース・ユーティリティとしてまとめた。

(1) スキーマの変更

データベースの定義のうち、ファイル名、項目名を変更するユーティリティである。

(2) ファイルの再構成

ファイルを編集し新しいファイルをつくる。

(3) ファイルの削除

該当ファイルを削除する。

(4) ファイル名一覧表示／印刷

スキーマ定義されているすべてのファイル名を表示／印刷する。

(5) ファイル項目一覧表示／印刷

指定されたファイルの項目名及び形式等を表示／印刷する。

(6) ファイルの外部保存

指定されたデータベースファイルをOSが標準としてサポートしているファイルへ、順編成ファイルとして出力する。

(7) 外部ファイルの導入

OSが標準としているファイルのデータをデータベースファイルへ読み込む。

(8) ファイル内容印刷

データベースファイル内のレコードを1次キー順に印刷する。

(9) ファイル更新情報

ファイルの更新履歴の保存/廃棄のモード設定及び更新履歴ファイルの消去、内容印刷を行う。

3 O A . O R G

基本設計の段階で、まず言語モデルとして α 言語を考えた。

C o d dの α 言語は、関係論理として与えた探索条件と、それによって検索されたタプル、あるいはタプルの(順序づけられた)組から生成すべき新しいタプルを定義する目的リストから構成されている。 α 言語は理論的に優れているものの、事務処理を記述するに当たっては、いくつかの点で欠陥がある。

その第1は外結合ないしはこれを3個以上の関係まで拡張した形の演算が指定できない点であり、第2は、目的リストが単純な射影演算に相当するものであり、事務処理を記述するには、十分強力と言えない点にある。第1のステップは、この2点が補強されたモデルを設計することであった。さて、 α 言語はごく一般的な環境の下で考えられていて、広汎な応用に適用可能であるものの、事務処理という特定の応用分野についての考慮は全くなされず、さらに関係論理は一般ユーザにとって決して理解し記述し易いものとは言えない。

この意味で第2のステップは、事務処理環境において容易に理解し使用し得るような言語と、環境に応じて高い効率を期待し得るプロセッサを開発することにあつた。この段階で、事務処理で瀕出する集約的演算を容易に遂行し得るような機能の組み込みが考えられた。こうして、O A . O R Gは次の4個のプログラムから構成される所となった。

3.2 リアクティブ処理・台帳作成

事務処理におけるマスタ・ファイルの作成、及び更新を行うことを主眼として設計した。

マスタ・ファイルへの入力機能及びすでに入力されているデータの検索、表示、変更を行う。また、検索した結果を印刷する機能及び変更したレコードを印刷する機能(変更履歴の印刷)を備えている。

表示形式は、いわゆる表形式のもの、1レコードを1画面に表示するもの(カード形式と呼んでいる)があり、随時切り替えが可能となっている。表形式の場合には、画面分割の機能があり、レコードの項目数が多い場合に有効に利用ができるようになっている。

検索に対しては、集合演算としての選択、更新に対しては、タプル単位の挿入、削除、変更が支援されていることになる。

3.3 リアクティブ処理・伝票発行及び台帳修正

入力されたデータに基づいて、関連するファイルの参照及び更新を行う。参照に際しては、参照すべきレコードがない場合にキーボードからの入力がたらく機能を持っている。更新に対しては、トランザクション・ファイルの作成を行う「蓄積」と、マスタ・ファイルの更新を行う「突き合わせ」の指定ができる。突き合わせを指定した場合は、画面上のデータと一致する値をもつレコードをさがし、そのレコードを更新する。このとき一致するレコードが見つからない場合にエラーとするか、レコードを追加するかを選択することができる。

さらに、入力されたデータ及び参照したレコードの項目に演算を加え、新たなデータをつくりだす機能が用意されている。演算子としては、四則演算と簡単な初等関数及び合計、平均、標準偏差などの集約関数が用いられる。検索に対しての集合演算としての選択及びネスト型結合、更新に対してのタプル単位の挿入、削除、変更が操作環境に応じて組み合わされていることになる。

3.4 バッチ処理・伝帳票発行

伝帳票を出力するための主となるファイルと、最大3個までの参照ファイルをもとに伝帳票を出力する。たとえば、日々のトランザクション・ファイルを主ファイルとし、トランザクション・ファイル内の項目の値により、得意先マスタファイルを参照するといったような手続きを定義することにより、伝帳票の作成が行われる。

主ファイルに対しては集約機能を用いることができる。出力に対しては、最大5段階までの集計を指定することができる。これは、出力するデータの値の変化により、集計機能が働くものであるが、この他にページ毎の集計をとる機能も備えている。出力すべき各項目に対して演算が加えられることについては、前述のリアクティブ処理・伝票発行及び台帳修正と同様である。区切り機能の挿入、ゼロ消去等の編集も容易である。この処理は、集合演算としての選択または、ネスト型結合であるが、集約などの強力な編集機能をあわせもっている。

3.5 バッチ処理・台帳更新

トランザクション・ファイルを使って台帳を更新することを主目的としたプログラムであり、入力ファイルが最大2つ、出力ファイルが1つから成る最も基本的な形態に処理対象を限定した。複雑な更新も、この形態を組み合わせるにより処理可能と判断した。

入力ファイルを2つ指定した場合には、2つめのファイルに対して参照または突き合わせの指定をすることにより、ネスト型の結合またはマージ型の結合が実行される。入力ファイルに対して集約機能を指定することができる。ネスト型結合の場合には、参照データがない場合の処理も指定できる。マージ型結合の場合には、入力ファイル1のみにレコードが存在する場合、入力ファイル2のみにレコードが存在する場合、入力ファイル1、2ともにレコードが存在する場合の各々について処理が指定できる。

出力については、出力ファイルの各項目毎に生成式を定義することにより

行う。

このユーティリティは、選択または結合を行うものであるが、結合に対しては外結合が支援されている。ネスト型結合については、主ファイルのレコードの存在が仮定されているが、マージ型結合ではこの仮定も置かれていない。

4 おわりに

パーソナル・コンピュータの上位機種を利用して事務処理全般を支援し、いわゆるOAを実現するために、データベースを中心とした統合ソフトウェアの1つの試みを述べた。データベース管理システムとして、関係型を採用したが、従来の関係型データベース管理システムにある様々な欠陥をいかにして解決するか1つの方策の提示ともなっているものと考えられる。システムの設計にあたっては、データベースの論理的枠組みと、その応用場面との双方について深い検討を行い、その調和が図られた。現在までに適用された業務の経験から追加、検索、更新時間は10000件までの範囲であるが、レコード数によらず一定しており高い安定性を示している。システムの開発の全般にわたって、産業能率大学 小林功武教授と菊池光昭助教授の指導を仰いだ。

参考文献

1. Kobayashi An Overview of Data Management Technology, in J. T. Tou, ed, Advances in Information Systems Science, Vol. 9, pp49~219 Plenum Publishing, New York, 1985
2. E. F. Codd A Data base Sublanguage founded on Relational Calculus Proc. ACM SIGFIDET'71, pp. 35~68, 1971
3. HOLON_H.DB、HOLON_OA.ORG、HOLON_S.SS、HOLON_I.WP、HOLON_A.BG、HOLON_DRAW 各レファレンス・マニュアル