

ユーザインタフェースを重視したデータベース管理システム C O O D の利用・評価・発展について

田中信行^{*,**}, 松田孝子^{*}, 吉川悦子^{**}

(^{*}東北大学大型計算機センター, ^{**}東北大学理学部)

1. はじめに

最近, ソフトウェアの評価尺度として, 機能, 性能, 安定/信頼性に加えて「ユーザ親和性」が議論される。T S S によるオンライン利用やパーソナルコンピュータの普及によって, 従来高速大量の計算に用いられてきた計算機が, 日常的な情報処理用のツールとして位置づけられ, コンピュータ利用層が一段と拡大したことからその要因であろう。

学術研究情報処理用として設置されオープン利用方式をとる大型計算機センターにおいては, ユーザ親和性の問題は一層重要である。利用分野が拡がり利用数が増し, コンピュータの性能が向上し多様性が増す状況の中で, ユーザが各自の処理を適確・容易に遂行する方法を見つけ出すことは難しくなっている。これに伴い, センターの業務も従来のオペレーション・パンケという機械対応の仕事から, 利用教育・相談, ソフトウェアの運用・管理など人間対応の仕事に多くの労力と能力を費すようになってきた。

我々は, このような実情を緩和するには, ソフトウェアの設計・開発をユーザ対応の「気配り」を持って行うことが重要であると考え, その実現をデータベース管理システム C O O D¹⁾ (Con conversationally Organized Database Management System) において試みた。C O O D の一般公開は 1979 年 10 月から行っているが, ここではこの満 3 年を経過した運用経験から特徴的な利用例に基づいて C O O D を評価し, その発展について報告する。

2. C O O D の設計と開発経過

C O O D は, エンドユーザを対象としたオンラインデータ蓄積と利用のためのツールであり, 主としてパーソナルデータベースのための管理システム²⁾である。ソフトウェアの使い易さは, 学習を含めた全ての操作・行動を容易に達成できることで評価されるので, 目的を単一化したり機能を限定したりして使い易いソフトウェアを開発することがよく行われる。この方法は終端的なソフトウェアの開発には効果的である。しかし, 多様なアプリケーションシステムを実現するために用いられる基礎的なソフトウェアの開発には採用できない。データベース管理システム (DBMS, Database Management System) は後者のソフトウェアの一つである。そこで, C O O D の設計・開発に当たってはまずユーザ要求・ユーザ行動を分析し, それをユーザインタフェースに反映させることを重視することにした。

C O O D の目標を「研究者がデータベースを手軽に作りそして使えること」に置き, つぎの方針で設計した³⁾。

- ・ユーザ指向-----オンライン指向の利用形態, コンピュータスペシャリストでないユーザ, F O R T R A N によるソフトウェアの蓄積があることを考慮する。
- ・習得容易性-----容易にシステムの習得ができるようにする。そのためには, データベースに関連する諸操作をユーザインタフェースであるデータベース用言語にできるだけ反映させ, その仕様を簡潔に説明書が薄くなるようにする。
- ・容易なデータ定義-----データモデルは表形式の単純なものとし説明付加による

定義ができるようにする。取扱うデータは数値データ（整数、実数など）と文字列データとし、各データタイプに対する空値の扱い、有効桁の保存、連続量の表現を可能にする。

- 容易なデータ操作………データの物理配置からユーザを解放するとともに、簡単な操作でデータの蓄積・追加・変更・移送および検索ができるようにする。また、FORTRAN プログラムからデータベースへアクセスできるようにする。
- 容易なデータベース操作………データベースの生成・追加・変更・削除・分割・統合などの操作が、オペレーティングシステムの諸機能についての知識がなくても容易に行えるようにする。

データベース用語をつぎのように分類、設定し、このほかに COOD システムを対話式に操作するコマンドを設けた。

- (1) DDL (Data Description Language) …… データモデルを記述する。
- (2) FDL (File Description Language) …… ファイル量、機密保護を記述する。
- (3) CML (Conversational Data Manipulation Language) …… 自立型のデータ操作言語で、会話形式に全てのデータ操作を行うことができる。
- (4) DML (Data Manipulation Language) …… FORTRAN を親言語とするデータ操作副言語で、データの蓄積、更新、検索を行うことができる。
- (5) CRL (Conversational Data Retrieval Language) …… データ検索専用の自立型データ操作言語で、会話形式にデータの検索・表示を行うことができる。
- (6) SVR (Database Operation Service Routine) コマンド …… データベース操作を交換するサービスルーチンを伴わせるコマンドで、会話式に操作できる。

COOD のシステム開発は、少し変則的な方法で行った。設計は田中、松田の2人で行ったが、システムインポリメントは松田1人で行い、平行して田中がデータベース構築への適用について検討した。COOD の基本モジュールが動作した段階で、吉川らによる COOD データベース構築を開始した。特に計算機についての知識を全く持たない実験補助者に会話式操作を依頼し、これらの過程を観察してシステムの開発およびユーザ用説明書の執筆にフィードバックした。

ユーザ用説明書^{4,5)}は、2人の設計者がそれぞれの立場で分担執筆したもので、「概要」、「文法」、「利用」および「実習例題」の4つの部分で構成した。ユーザは、概要で COOD とデータベースについての簡単な知識を入手し、例題で COOD に触れてみる事ができる。文法は規則をまとめたもので、利用は適用例を種々の角度から述べているものである。

3. COOD の利用例

研究データの収集・整理・引用は研究活動の中で必須の過程である。これらの研究データ管理に COOD を適用した。図1に研究データの管理と共用の関連を示す。研究グループは、代表者 H とメンバー a, b, c, d からなるものとし、各研究者が何々に "REFx" (Researcher's File x) という名前のパーソナルデータベースを持ち、その中に各自の文献、実験データ、収集データを収録するテーブルを必要に応じて随時作成・追加する。各データベースあるいはテーブルには、随意他の研究者からアクセスできるような許可を付けたり、また取りはずしたりする。また、これらのデータベースの中にはグループ外の研究者からのアクセス要求に応じるために一般公開したデータベース CORMEC や ELDOL が

含まれている。

これらのデータベース群の作成・管理・利用に伴う諸操作はつぎのものである。

- (1) データベースの定義と生成
- (2) データの蓄積
(格納, 追加, 変更)
- (3) データベースの定義の変更
- (4) データベースの検索
- (5) プログラムとデータベースの結合
- (6) データベース共用のための操作
- (7) データベースの分割・併合

図2にDDLおよびFDLによる

データベースの定義とその定義に基づくデータベースファイルの生成例を示す。図3はデータ入力の例である。この例は対話式にデータをタイピングしているが、このほかにエディタと組合せて行う方法、FORTRANプログラムと組合せて行う方法があり、適宜便利な方法を用いて行うことができる。

一旦定義したデ

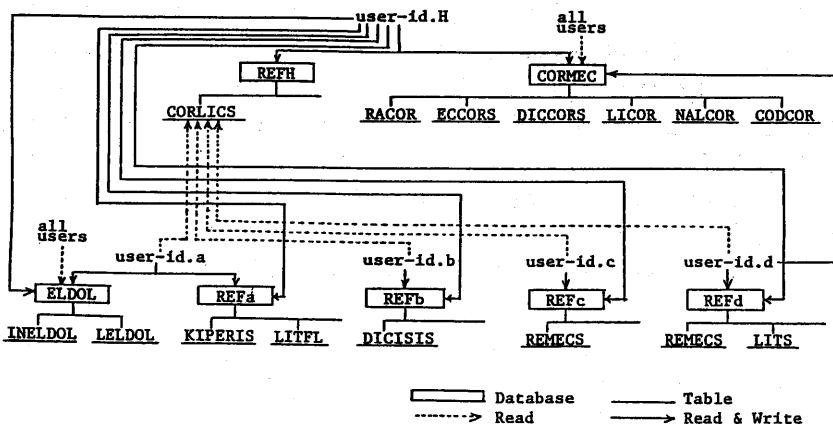


図1. 研究データの管理と共用

```
DDL;
DATABASE CORMEC : COMPLEXATION REACTIONS OF METAL COMPLEXES;
TABLE RACOR : RATES OF COMPLEXATION REACTIONS;
NO (I4) UNIQUE : Serial number;
METAL (A70) : Metal ion;
LIGAND (A70) : Ligand;
RATELAW (A70) : Rate law;
RKVALUE (J10) : Rate constant(value);
RKUNIT (A20) : Rate constant(unit);
HACTIV (J5) : Activation enthalpy;

REMARKS (3) (A70) : Remarks;
COMPILER (A70) : Compiler(s);
TABLE ECCORS : EQUILIBRIUM CONSTANTS OF COMPLEXATION REACTIONS(SELECTED VALUE);
NO (I4) UNIQUE : Sereal number;
METAL (A70) : Metal ion;

END-DDL;
```

図2. データベースの定義と生成

(データ定義 DDL)

```
COOD-PROCESS ... ?CML
?USE CORMEC/RACOR;
EXPLAIN ITEMS OF RACOR, YES OR NO ?
?STORE NEW RACOR;
DISPLAY, NAME(N) OR EXPLANATION(E) ?
NO
=278
METAL
=Ni(2+)
LIGAND
=EDTA
RATELAW
=k[Ni(2+)] [HL(3-)]
RKVALUE
=3.0E+5
RKUNIT
='1 mol(-1)s(-1)'
HACTIV
=-
COMPILER
='Tanaka M., Yamada S., Kikkawa E.'
NO
=-
*** 1 DATA STORED.
?
COOD-PROCESS ... ?
```

図3 対話式データ格納

```
FDL;
DATABASE CORMEC;
PERMISSION WRITE/user-id.d/;
TABLE RACOR;
MAX 750;
TABLE ECCORS;
MAX 600;

END-FDL;
```

(ファイル定義 FDL)

SYSTEM ?COOD

COOD-PROCESS ... ?DDL
SOURCE FILE ?CORMECDDL

COOD-PROCESS ... ?FDL
SOURCE FILE ?CORMECFDL

COOD-PROCESS ... ?DFC
DATABASE NAME ?CORMEC
DATABASE FILE CREATED.

COOD-PROCESS ... ?
SYSTEM ?

(データベースの生成)

下線部分がユーザー入力
→は読取のみを示す。

データベースの定義を変更したり、テーブルを追加したり、削除したりする場合がある。パーソナルデータベースではこのような操作はとりわけ多い。図4は名称変更の例である。図2, 図3および図4の組合せでデータベースの再構成を達成することができるといえる。

データベースの検索機能には、検索条件式による検索、集合演算、ソートなどがあるほか、図5に示すような表形式の出力が可能である。図6はFORTRANプログラムからデータベースをアクセスしている例である。ポリオドではじまるステートメントがDMLである。このプログラムは、錯形成反応データベースCORMECを利用するアプリケーションプログラムAPCORの一部である。APCORはFORTRANを中心にCODDデータベース、数値計算および作図用ライブラリを結合して、錯形成反応に係る各種定数の計算やグラフの作成を行っている。

データベースの共用範囲を設定変更するには、図4のようにSVRを用いて、たとえば、つぎのような会話形式で行う。

```
?Update permission CORMEC;
DATABASE PERMISSION ?READ
TABLE ?CODCOR
TABLE PERMISSION ?WRITE/user-id.H,
user-id.d/
```

また、データベースの中の一部を分割し他のデータベースの中に統合するには、図3のようにCMLを用いて、たとえば、つぎのような会話形式で行う。

```
?Use CORMEC/RACOR=MYRACOR;
?Use #DATABASE/CORMEC/RACOR;
?Select*all MYRACOR to RACOR when(NO<=200);
```

以上の例に示すごとく、CODDデータベースに関連する全操作を研究室のTSS端末から行うことができ、研究データを管理するツールとしてCODDは十分に機能している。

4. CODDの評価

CODDのユーザ適合性をつぎの観点から評価してみる。

```
COOD-PROCESS ... ?SVR
?UPDATE NAME REFB:
NEW DATABASE NAME ?Z
NEW DATABASE EXPLANATION
?Z
TABLE ?LIT
NEW TABLE NAME ?Z
NEW TABLE EXPLANATION
?Z
ITEM ?AUTHOR
NEW ITEM NAME ?A
NEW ITEM EXPLANATION
?Author(s)
ITEM ?TITLE,T
NEW ITEM EXPLANATION
?Title
```

図4. データ定義の変更

	METAL	LIGAND	RATELAW	RKVALUE	LITNO
1)	Cd(2+)	pada	k[Cd(2+)] [L]	1E+7	LICOR-16
2)	Cd(2+)	phen	k[Cd(2+)] [L]	1E+7	LICOR-26
3)	Cd(2+)	terpy	k[Cd(2+)] [L]	3.2E+6	LICOR-26
4)	Cd(2+)	EGTA	k[Cd(2+)] [H2L(2-)]	1.5E+6	LICOR-79
5)	Co(2+)	picolinic acid	k[Co(2+)] [L(1-)]	1E+7	LICOR-52

図5 検索結果の作表表示

```
SUBROUTINE RETC(METL,LIGND)
C Data Retrieval from Database CORMEC
COMMON M,N,M1,N1, PK1,PK,ECPK, KH1,KH,ECKH,
& KOH1,KOH,ECKOH, KNL1,KML,ECKML, B1,B,ECB
& USE CORMEC/DICCOR(S(LIGAND=LIGAND1,ECONST=ECONST1,PKVALUE)
& ,ECCORS;

.ACCEPT NULL(DOUBLE PRECISION) TO DNULL;
.OPEN DICCOR(S;
10 .FIND DICCOR(S;
.IF END,GOTO 19;
.GET LIGAND1,ECONST1,PKVALUE;
.IF(PKVALUE.EQ.DNULL) GOTO 10
.IF(CHASK(LIGAND1,70).NE.CHASK(LIGND,70)) GOTO 10
.DECODE(ECONST1,100) I
N=MAX(I,N)
PK1(I)=PKVALUE
PK(I)=10.0**(-PKVALUE)
ECPK(I)=ECONST1
GOTO 10
19 .CLOSE DICCOR(S;
```

図6. FORTRANプログラムからの利用

(1) 習得容易性----- C O O D 説明書⁸⁾は 116 ページで、その中に前述 (2.) の 4 つの部分が入っている。講習会などにはまだ聞いたことはないが、ユーザは独習で利用している。C O O D のほかにエディタを併用することが多い。

(2) 利用範囲----- C O O D の譲渡先は大阪大学、京都大学など現在 8 機関である。研究分野で見ると理学、医学、工学、薬学、経済学、文学などの範囲に亘る。

(3) 利用数----- 東北大学大型計算機センターで C O O D をアクセスした利用者数 (課題番号) は現在 75 である。使用件数を昨年 11 月と 12 月の調査から拾ったものを表 1 に示す。公開しているソフトウェアの中から F O R T R A N のような基本ソフトウェア、S S L や作図ルーチンのような使用頻度の高いサブルーティンライブラリ、T S S コマンドに類するものを除いたもので、L I S P, S A L S, G P S S, U N I Q-1 などを含むアプリケーションライブラリのうち上位 7 つを示した。統計用の S T A T P A C, S P S S, 結晶構造解析用の U N I C S に並んで C O O D の使用頻度は高い。大阪大学においても同様の傾向を示すことが報告されている。

表 1. アプリケーションライブラリ使用回数

順位	Nov. 1982	Dec. 1982
1	STATPAC 7261	STATPAC —
2	SPSS 604	SPSS 2047
3	UNICS 350	IGL 1235
4	COORD 346	COORD 552
5	REDUCE 278	UNICS/UNICS3 501
6	GPSS 220	ISAP 488
7	TABAI 171	CLUSTER 408

(4) 利用目的----- 数値・文字データを扱い F O R T R A N とも結合できるので、前述 (3.) のような使用例が多く報告されている^{8,9)}。そのほか個人文献ファイルなどの作成に使用されているものと推測される。

(5) 満足度----- C O O D を 3 版にバージョンアップする時にそれまでのユーザの要望事項をほぼ組入れたので、残る問題点としては検索スピードと D M L の機能向上があげられている。検索スピードについては次章で述べる対策をとった。D M L の機能については、F O R T R A N と様々の関係と合せて今後の問題として残っている。しかし、現在の D M L でも F O R T R A N の入出力と比較すれば格段に易しい。C O O D における空値、有効桁、連続量の扱い、会話式操作、プログラムとの結合などユーザのアプリケーションに馴れみ良いことが好評を得ている。

5. C O O D の発展

C O O D で作成されたデータベースを共用の目的で一般公開する場合に、検索スピードが問題になる。C R L で検索する場合の実測では、所要 CPU 時間は約 2 秒/1000 件で直線的増加となるが、経過時間は同時アクティブ端末数、同時実行プログラムの性質などの影響を強く受け、分の単位で急速に増加する。これは C O O D のプログラム構成によるものではなく、高速の CPU と低速のファイル間の転送機構ないしは構成によるもので、同様の現象は F O R T R A N の入出力においても観測される。

C O O D ではこのような状況の下で検索スピード向上に対処するために以下の 2 つの方策をとることにした。

(1) より大規模な D B M S への自動変換を行う。

(2) 他の多くの D B M S と同様に C O O D にも転置ファイル作成機能を設ける。自動変換例として C O O D - I N Q を行った¹⁰⁾。

* 東北大学大型計算機センター、一関京子さんの提供による。12月のSTATPACについては誤操作でデータ消失。

C O O D はパーソナルデータベースのための DBMS を本来の目的とするので、データベース管理者は各研究分野の研究者である。このような研究者にデータベース公開のための過大な労力を課することはできない。また、データベース中のデータは数値ないしキーワードに切り出しにくいものも多く、使い方は非定型のものとなろう。このようなデータベースの転送ファイル設置の方法として、図 7 に示すようにデータベース利用者 (user-id. x) が各自の条件で索引ファイル (Ifile) を作成する方法をとった。索引ファイルには引用したいデータベース中のデータへのポインタを格納し、また作成時のタイムスタンプ (D, T_x) を押して保存する。索引ファイルの利用時にこのタイムスタンプと各データベース中に更新時に押しておくタイムスタンプ (D, T_a など) とを照合して利用可否を決めている。

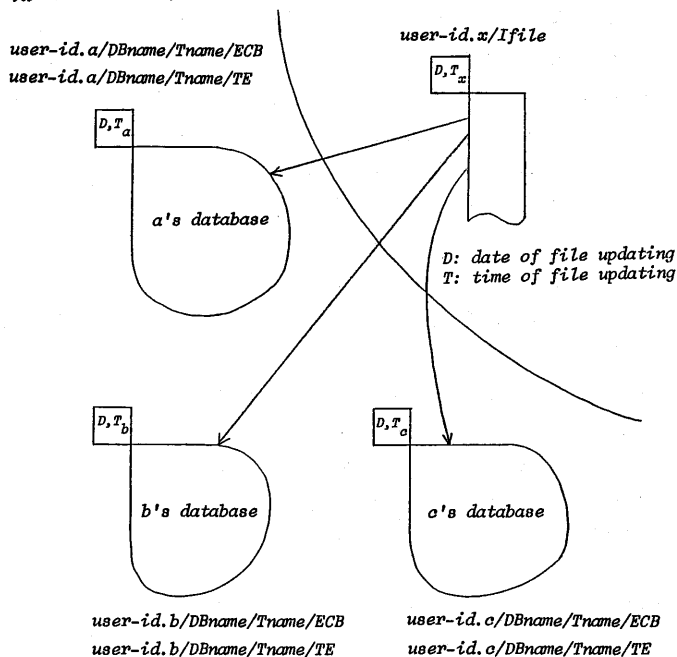


図 7. CRL における索引ファイル

6. おわりに

C O O D は DBMS として独立に使われるほかに、FORTRAN のより簡便なファイル入出力機能としても役立ち、プログラミング言語のとかく複雑になりつつある入出力部分の一つの方向を示すものである。

使い易いソフトウェアの開発には、C O O D の開発経過は大いに参考になるものと思う。これまでのソフトウェアの開発においては、先達のシステム、有名なシステムを見習うことに費す労力が多く、ユーザに目を向けることが少なかったのではないだろうか。その結果として巨大な使い難い“もの”が氾濫することになったのではないだろうか。“ソフトウェアの開発者はユーザフロアに下立って人の世話をしてみる”ことを勧めたい。

参考文献

- 1) 松田, 田中: ユーザー指向のデータベース管理システム C O O D — 設計とデータベース用言語, 情報処理学会論文誌, Vol. 21, No. 5 (1980).
- 2) 松田: 研究活動におけるデータベース技術の応用と展望, 情報処理, Vol. 23, No. 10 (1982).
- 3) 田中, 松田: 会談型データベース管理システム C O O D (第 3 版), TC 研究集録, TC 19 II-15 (1982).
- 4) 田中, 松田: C O O D 文法説明書 (第 2 版), C O O D 使用説明書 (第 2 版) (1979).
- 5) 松田, 田中: 会談型データベース管理システム C O O D 説明書 (第 3 版) (1982).
- 6) 田中, 吉川, 松田, 山田: 分析化学におけるコンピュータデータベースとその利用 I. 錯形成反応データベース, 分析化学, Vol. 30, No. 9 (1981).
- 7) 田中, 吉川, 松田: CORMEC・APCOR 説明書 (1982).
- 8) 松田, 田中: パーソナルデータベースの構築と統合による情報システムの一設計, 情報処理学会アドバンス・データベース・システムシンポジウム論文集 (1982).
- 9) 赤石: 大型コンピュータによる毒物制御例のデータベース化と検索システムの開発, 東北大 (1982).
- 10) Isomoto, Matsuda, Tanaka: Guidance system for structuring or restructuring of a database in multiple database management systems, J. Inf. Process. Vol. 5, No. 3 (1982).