

## 解説

## 古河電工におけるデータベース適用事例\*

牛田善和\*\* 桑邑一男\*\*\*

## 1. はじめに

昭和49年1月に、FACOM 230-55を導入することが決定し、各システムのコンバートが検討された。その際、旧機種での処理の問題点が摘出され、限られた時間の中でできるだけの問題点を解決するように検討した。

対象の業務は大別すると、営業関係、人事関係、会計関係、技術計算の4分野に分類できる。特に問題となったのは、1つのオーダがどの状態まで進行したかの進捗状況を把握することで、各ファイルを使用しているバッチ処理ではその実現はむずかしく、データ間の関連づけが問題となった。その問題解決を含めて、大型計算機を導入することにより、

- (1) データベース・マネージメント言語 (DBMS) が使用可能になる。
- (2) 大容量ファイルの使用が可能になる。
- (3) 多重処理を利用した即時処理により、アップ・トゥーデイトなファイル更新が可能になる。
- (4) 伝送回線を利用した照会応答、更新が可能になる。

という背景を基にデータベースの採用を検討した。

当社のデータベースの特徴は、唯一つの全社データベースを作るという考えではなく、各業務分野ごとに必要ならば個別のデータベースを設計することとし、オープン型 DBMS と、クローズ型 DBMS の併用で問題を解決していくことである。

我々が当面考えているデータベースとしては、

## (1) 営業データベース

ランニング中のオーダーステータス把握——オープン型

時系列データ分析——クローズ型

社外データ分析——クローズ型

## (2) 会計データベース

財務データ分析——クローズ型

管理会計用データ分析——クローズ型

## (3) 人事データベース

人事情報管理システム——オープン型

のデータベースである。

機器構成は、昭和50年10月に、FACOM 230-58 にレベルアップし、現在図-1(次頁参照)の構成になっている。

## 2. データベースの開発

## 2.1 使用 DBMS

FACOM 230-58 におけるオープン型 DBMS は、既存の RAPID\*\*\*\* とオンライン・ソフトウェア SOM\*\*\*\*\* とを結合し、オンライン・データベース INIS\*\*\*\*\* として拡張されている。さらに、データベースの検索、レポート作成等のノンプログラマでも容易に扱うことのできる簡単な独自の言語をもつクローズ型 DBMS として、STAFF\*\*\*\*\* がある。

処理手順、処理内容が決まっている定型処理では、前者の INIS を使用して開発、運用を行い、定型レポート以外の情報を必要とする管理者、分析者、担当者からの突発的要求に対しては、後者の STAFF を使用し対処している。ここでは、前者の INIS を使用したデータベースに関して説明する。

## 2.2 データベース開発手順

INIS (DB) の設計は大きく分けて、

- (1) データベースに収容する項目をまとめ、および項目間のつながりを定める論理構造の設計、
- (2) 処理効率、スペース効率を落とさないように項目を物理的にランダム・アクセス・ファイル

\* The Application of Data Base in Furukawa Electric Co., Ltd. by Yoshikazu USHIDA (Systems Engineering Division, The Furukawa Electric Co. Ltd.) and Kazuo KUWAMURA (Systems Engineering Division, Computer Systems, The Fujitsu Ltd.)

\*\* 古河電工(株)計数部

\*\*\* 富士通(株)第2システム部第3システム課

\*\*\*\* RAPID (Retrieval And Production for Integrated Data)

\*\*\*\*\* SOM (Standard Online Module)

\*\*\*\*\* INIS (INtegrated Information System)

\*\*\*\*\* STAFF (STAF Ffor file management system)

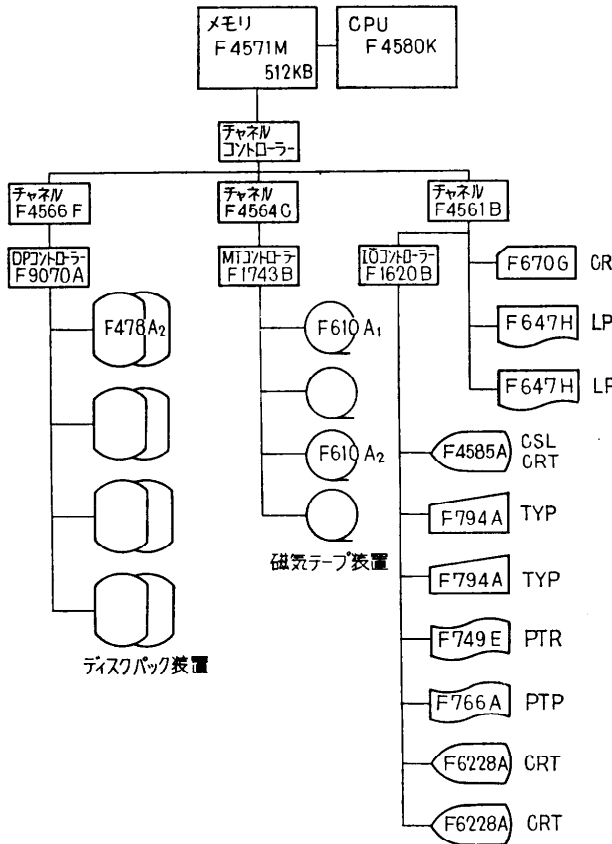


図-1 F230-58 構成図

上にどのように收容するかを決定する物理構造の設計、  
の2つに分類することができる。

従来のシステム設計と異なり、データの共有化を本格的に目指すためには、伝統的なファイルや、データに関する概念を変える必要がある。従って、はじめて

データベースを採用する者にとって最適なデータベースを構築することは非常に骨のおれる仕事となる。また、上記(1),(2)の設計過程で、試行してみないと決定できない点も多く、一般に次のような手順をふみ、データベースの開発を行っている。

- (1) データベースの理解、
- (2) 適用業務の分析 (データベースの適用範囲の明確化)。
- (3) データの内容分析, データ量分析。
- (4) 論理構造の設計。
- (5) 物理構造の設計。
- (6) PILOT MODEL のための抽象化したメイン業務の仕様作成。
- (7) PILOT MODEL による RUN。
- (8) PILOT MODEL の評価 (処理効率, スペース効率)。
- (9) 論理構造の再設計。
- (10) 物理構造の再設計。
- (11) 本番用プログラムの仕様作成。
- (12) データベースの登録。
- (13) データベースの SET UP。
- (14) プログラムのデバック。
- (15) 論理構造の微調整。
- (16) プログラムの TEST RUN。
- (17) 総合評価。

以上を図示すると図-2 のようになる。

### 2.3 論理構造設計への過程

データベースでは、前もって考えられるデータ間のつながりを登録しておく必要があり、さらに従来のファイルに相当するデータのグループを作ってやらなければならない。

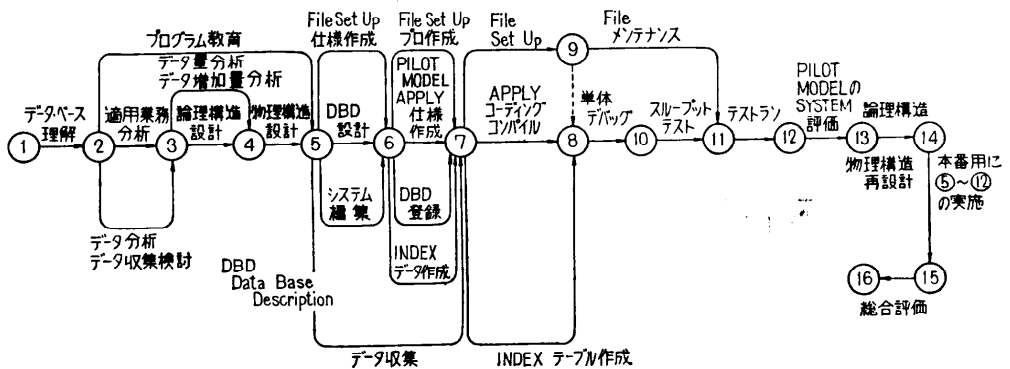


図-2 データベースの設計手順

INIS ではグループのことを SEGMENT と呼び、特にデータベースの入口に当る SEGMENT を ENTRY SEGMENT という。また、SEGMENT 間のつながりを SET と呼ぶ。INIS の基本データ構造は RING 構造をとっており、単に POINTER の修正だけで効率的なメンテナンスが行えるようにしてある。

検索手段も図-3 に示すように

- (1)  $A_1 \rightarrow B_1 \rightarrow B_2 \rightarrow B_3$  検索順序を NEXT
- (2)  $A_1 \rightarrow B_3 \rightarrow B_2 \rightarrow B_1$  検索順序を PRIOR
- (3)  $B_1 \rightarrow A_1, B_2 \rightarrow A_1, B_3 \rightarrow A_1$  検索を MASTER

の3つの RING TYPE がある。

データベースの構造は、SEGMENT を単位としてネットワーク構造まで柔軟なデータ構造がとれることが必要であり、INIS は図-4 のように RING 構造をもとに簡単な階層構造から複雑なネットワーク構造までのデータベース構築が可能である。

以上のことから、

- (1) データを分析し、SEGMENT を構成する項目を決定する。
- (2) 定型処理を中心にプログラムの分析を通して SET 関係を決定する。
- (3) プログラムの分析、今後の検索条件などの推測からデータベースの入口である ENTRY SEGMENT を決定する。

などがデータベース論理設計の作業である。

### 2.4 物理設計への過程

データベース設計者は、論理設計で決定した SEGMENT を実際にランダム・アクセス・ファイル上どのように配置していったらよいのか検討しなければならない。

INIS では、狭義な意味で図-5 に示すような関係でデータベースを構成している。

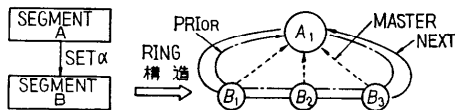


図-3 RING TYPE

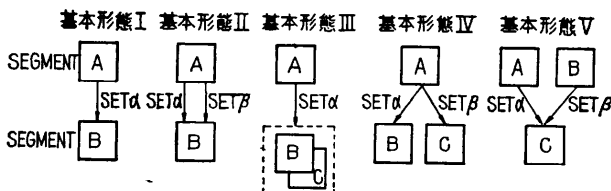


図-4 階層関係の基本形態

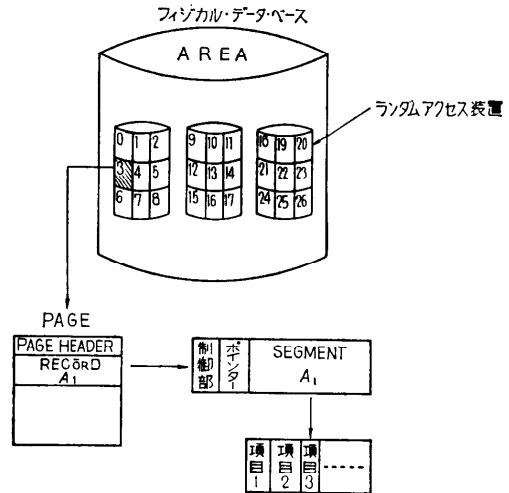


図-5 狭義 DATA BASE の構成

PAGE は、物理的な FILE のブロックであり、アクセスの単位でもある。

物理設計の作業としては、

- (1) プログラム単位による SEGMENT 使用頻度の検討。
- (2) SEGMENT SIZE の決定。
- (3) SEGMENT 別データ量検討。
- (4) RECORD SIZE の決定。
- (5) PAGE への各 SEGMENT 格納方法検討。
- (6) PAGE SIZE の決定。
- (7) 空領域管理方法の検討。
- (8) 各 SEGMENT ごとに必要な PAGE 数算出。
- (9) オーバフローの設定検討。
- (10) ランダムアクセス装置の使用台数算出。

を順次決定していくこととなる。

### 3. 当社における適用例

現在、INIS を使用して運用中、開発中のデータベースとして、

- (1) 人事情報管理システム
- (2) 営業オーダーエントリ・システム
- (3) 電算室総合管理システム

の3つのデータベースがある。

現在、(1)、(3)の処理はデータベースを採用して実行中で、(2)のデータベースは鋭意開発中である。人事情報管理システムは、1年間の開発期間で完成したデータベースで、49年1月より運用を開始し、現在に至っているもので

ある。人事処理は、約 15,000 人のデータに限定できる処理がすでに安定している。マスター類のデータの整備が行き届いている、コード化が進んでいる、システムのユーザが人事にしばられているなどの要因を考へて、最初にデータベースを採用してシステム設計をしたものである。

3.1 人事情報管理システム

人事情報のファイルを一元化してデータベースとし給与計算はもとより、スキルズインベントリー、経歴管理といったマネジアルな面までを吸収するシステムを作り運営しているデータベースである。

3.1.1 データベースの目標

従来の処理は、基本データ・ファイル、社会保険データ・ファイル、控除関係データ・ファイルを個別に順編成ファイルとして持ち、給与計算全般、社会保険処理、控除処理、融資残高管理などの例月処理、昇給賞与、査定考課、年末調整計算などの期間処理、自己申告制度による個人情報管理に加えて、退職金引当などの会計処理までも実施していたが、次のような問題点が指摘されていた。

- (1) 個人に関する情報が処理ごとに複数個のファイルに分割され保管されているので、処理要求ごとにマッチングする必要があり、処理が複雑である。
- (2) 個々のファイルが別々になっており、各ファ

イルを更新するためのインプットが事務担当者の段階から別になっていて、各ファイル間のアンマッチ、情報の重複が問題である。

- (3) ファイルが各々別々なので、処理の段階で異なったファイルの管理が問題である。
- (4) ファイルは 1レコード固定長の設計であり、項目の数、項目の桁の変更が不便である。

これらの問題点を解決し、発展させることをデータベース構築の目的として、人事データベースの目標を次のように設定した。

- (1) 個人単位に情報を一元化し、各種のキー指定に対して、自由に関連したデータが取り出せること。
- (2) 一度の更新処理で関連あるデータは、すべて更新すること。
- (3) ファイルのデザインと処理を分離し、デザイン変更に対処できること。
- (4) 保存ファイルは、インプットデータとデータベースのみとし保管を容易にすること。

3.1.2 論理構造と物理構造

このデータベースの論理構造、物理構造は、各々図-6、図-7(次頁参照)の如くである。

論理構造を設計し、物理構造を決定していくのは、データベース設計の中心部分であり、特に設計過程で我々が工夫した点を以下に述べる。

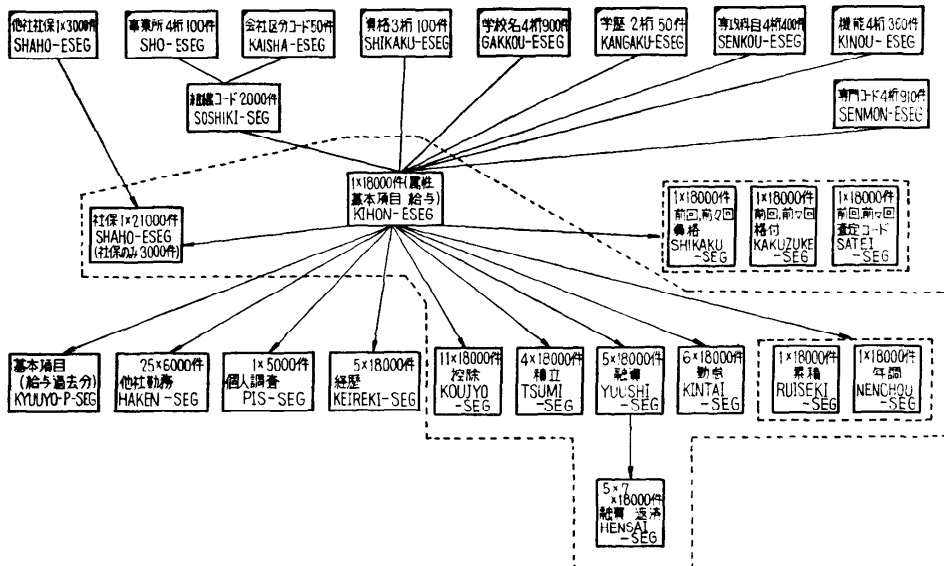


図-6 論理構造

SEGMENT 名	SIZE	データ 件数	所要ペ ージ数	START -PAGE	END -PAGE
PAGE-SSEG		1	1	0	0
SHAHO-ESEG	42	4,000	30	1	30
SHO-ESEG	20	100	1	31	31
KAISHA-ESEG	18	50	1	32	32
SOSHIKI-SEG	36	2,000	10	33	42
SHIKAKU-ESEG	20	100	1	43	43
GAKKOU-ESEG	16	900	2	44	45
KANGAKU-ESEG	14	50	1	46	46
SENKOU-ESEG	16	400	1	47	47
KINOU-ESEG	20	360	1	48	48
SEMON-ESEG	24	910	3	49	51
KIHON-ESEG	370	24,000	6,000	52	6,051
SHAHO-SEG	102	24,000			
KOUJYU-SEG	16	264,000			
YUUSHI-SEG	42	120,000			
HENSAI-SEG	18	340,000			
TSUMI-SEG	28	96,000			
RUISEKI-SEG	82	24,000			
NENCHOU-SEG	38	24,000			
KINTAI-SEG	16	144,000			
SHIKAKU-SEG	30	24,000			
KAKUZUKE-SEG	30	24,000	300	6,052	6,351
SATEI-SEG	30	24,000			
*SHAHO-SEG	102	4,000			
KYUUYO-P-SEG	58	24,000	194	6,409	6,602
HAKEN-SEG	24	18,000	60	6,603	6,662
PIS-SEG	508	5,000	358	6,663	7,020
*KOUJYO-SEG	16	12,000	27	7,021	7,047
*TSUMI-SEG	24	18,000	60	7,048	7,107
*YUUSHI-SEG	42	9,000	53	7,108	7,160
*HENSAI-SEG	18	63,000	158	7,161	7,318
*KINTAI-SEG	16	6,000	14	7,319	7,332
*NENCHOU-SEG	38	1,000	6	7,333	7,338
KEIREKI-SEG	24	120,000	400	7,339	7,738
オーバーフロー領域				7,339	7,800

\* SHAHO-SEG, KOUJYO-SEG, TSUMI-SEG, YUUSHI-SEG, HENSAI-SEG, KINTAI-SEG, HENCHOU-SEG は、1人当たり平均件数を越えたものを収容するページである。

図-7 物理構造

実際にデータベースを収容する場合、そのページの大きさと同一ページ内にどの SEGMENT を何件ずつ一緒に収容していくかが問題になる。

同一ページには、同一処理で取り扱う関連ある SEGMENT は同時に収容しておいた方が処理スピードの点からいって当然よいことになるが、どの処理でも取り扱う SEGMENT が同一とはならず、どの処理を優先させるかによって収容方法を決定していかないとまずいわけである。また、図-6 の如く、1つの基本項目に複数の SEGMENT が結合される可能性がある場合、いくつまで基本項目と同一のページに収容した方が良いかが問題となる。その場合、平均何個という分析と同時に、最大何個という分析も必要となり、ある定数個まで同一ページに収容しオーバーフロー分は別ページに収容するように設計していかなければ、同一

ページ内の空領域が多くなって無駄になり、問題である。これは、従来のシステム開発のファイル設計と、プログラムの仕様作成を同時に進めていく感じになる。

ページの大きさは、以上の同一ページへの格納方法の考慮と同時にコア上のバッファ領域の大きさとも考え合わせていかねばならない。ランダム処理を考えれば、1ページの大きさを小さくして、バッファ内に多数ページ収容した方が効率がよく、シーケンシャル処理を考えれば、1ページの大きさを大きくして、バッファ内のページ数を少なくしても効率は変わらないことになるので何に重点を置いて設計するかにより決定されることになる。

3.1.3 処理内容

現在、人事データベースを使って次のような業務を行っている。

- (1) 昇給計算。
- (2) 月例給与計算。
- (3) 臨時給与計算 (賞与など)。
- (4) 年末調整計算。
- (5) 社会保険資料。
- (6) 労政資料。
- (7) 会計資料 (退職金、決算用半月計算など)。
- (8) 人事情報管理 (個人調査票、経歴情報など)。

この中で、月例給与計算システムについて概略記述する。

月例給与計算は、

- (1) 基本項目の更新
- (2) 社会保険の更新
- (3) 控除項目の更新
- (4) 支給額計算
- (5) 計算結果諸資料のアウトプット

の5段階で行っている。

- (1) 基本項目の更新

図-6 の基本項目の内容を更新するのが目的で、所

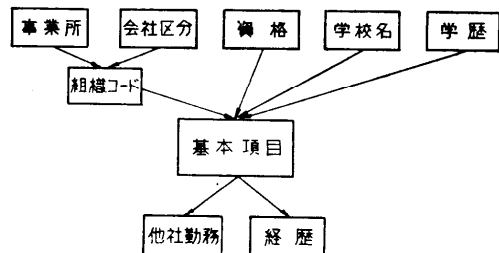


図-8 基本項目更新処理に関する論理構造

属の変更、家族構成の変更、退職、採用、資格の変更などを行うと同時に、組織コード、資格、学校名、学歴の各 SEGMENT との SET を更新する。

### (2) 社会保険の更新

健康保険、厚生年金、雇用保険のマスターの内容を更新する目的で、登録、脱退、等級変更などを行い、同時に、健康保険の各種給付計算も行っている。

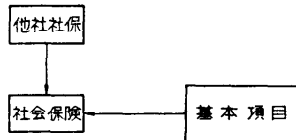


図-9 社会保険更新処理に関する論理構造

### (3) 控除項目の更新

固定的な控除項目の内容更新、積立金額の内容更新融資金の残高管理の内容更新を行っている。その月のみの変動的な控除はここで扱っていない。

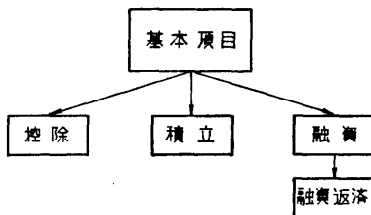


図-10 控除項目更新処理に関する論理構造

### (4) 支給額計算

ここでは、例月変動する項目（出勤日数、残業時間、休暇など）をインプットに基本項目を索引しながら、支給額計算、税金計算、手取額計算を行っている。

(5) 計算結果ファイルを利用して諸資料をアウトプットしているが、そのファイルはクローズ型 DBMS の STAFF のマスターにも登録しており、大量データを扱った突発要求には、STAFF を使用して対処している。

現在の運用状態は次のようである。

- |             |                |              |
|-------------|----------------|--------------|
| (1) 基本項目の更新 | データ量 200 件     | ランニング時間 8 分  |
| (2) 社会保険の更新 | データ量 366 件     | ランニング時間 10 分 |
| (3) 控除項目の更新 | データ量 1,600 件   | ランニング時間 15 分 |
| (4) 支給額計算   | データ量 11,500 人分 | ランニング時間 20 分 |
| (5) 諸資料の作成  | 種類数 34 種類      | ランニング時間 7 時間 |

#### 3.1.4 評価

人事データベースを運営してきて、次のような利点

があげられる。

- (1) ファイルが一元化されたので、ファイルの管理が容易になり、オペレーションも楽になった。
- (2) 関連したデータの索引が楽になり、処理ステップが減った（例えば、従来の例月給与計算の処理ステップは 77 ステップであったものが、新システムでは 51 ステップとなっている）。
- (3) 更新の回数が減り、1つのインプットで関連データの更新ができる。
- (4) プログラムの仕様を考えると、1つのファイルから出発できるのでまとめやすくなった。欠点もないわけではない。

- (1) DBMS の理解が不十分で、データの組み合わせによってバグがでる。
- (2) データベースを更新中にダウンすると途中ページまでの更新がページ単位に終わっているので復元が面倒である。
- (3) 更新時に、関連あるデータ全部のつながりも更新していくので、単一目的からいうと時間が余計にかかる。

我々も十分にデータベースを利用しきっていないが、特にマネジアルな面の利用を現在開発中であり、その時点でさらに効果を発揮するものと期待している。

## 4. おわりに

電算機の主目的は、営業支援、決算業務、管理会計、人事業務など各種考えられが、営業支援が重点目標であり、営業オーダエントリー・システムの完成と、その展開が当面の課題である。

データベースを設計し、運用してみて、我々がデータベース・マネージメント言語に今後期待する機能は、

- (1) 物理構造の最適設計手法
- (2) パイロット・データベースのシミュレーション手段
- (3) デバッグ用テスト・データ・ジェネレータ
- (4) スペースの有効活用

などである。

今後、データベースが我々 EDP 部門のシステムの中核となっていくことは自然の流れであり、秀れた汎用化されたデータベース・マネージメント・システムが開発されることを望んでいる。

(昭和 51 年 6 月 8 日受付)

(昭和 51 年 7 月 21 日再受付)