

コンピュータ
サイエンス
シリーズ

データベース・システム

データ・ベース・システム

上條史彦著

産業図書

<著者略歴>

かみ じょう み ひと
上 條 史 彦

昭和32年 東京大学工学部卒
昭和42年 米国マサチューセッツ工科大学修士課程卒
昭和47年 日本アイ・ビー・エム(株)
昭和48年 現在情報処理振興事業協会開発振興部長

データ・ベース・システム

定価 1600円

昭和47年7月10日 初版
昭和54年9月25日 第9刷

著者 上 條 史 彦

発行者 森 田 勝 久

発行所 産業図書株式会社

東京都千代田区外神田 1-4-21
郵便番号 101-91
電話 東京(253) 7821 (代)
振替口座 東京2-27724番

は し が き

データ・ベースという言葉が使われるようになってから久しい。言葉の意味は必ずしも1つではなく、時代と環境によって多少のニュアンスの違いがある。最近では差異もかなり小さくなり、前後関係から意味を汲み取れば議論をするうえでは困らなくなった。厳格に考えれば、まだまだデータ・ベースに関する書物を公けにするには時期がはやすぎるかもしれない。しかしながら、こと計算機利用の分野では、呼び名はともかく、本書で述べるようなソフトウェアが1960年代初期から開発され、実用化されてきているのである。ここにも理論より実用化が先行している、システム・エンジニアリングの分野ではよく見られる現象が起きている。

データ処理を行なうためにはハードウェア、ソフトウェアとともにデータが必要である。技術という面から眺めると、1960年代にハードウェアとソフトウェアについては主要な部分の開発は終わった感がある。しかしデータの管理、ファイルの体系化の分野はやっとじまったばかりということができよう。現在データ・ベースの管理技術として提唱されているものにはデータ・ファイルの検索を行なうためのシステムと、ファイルの総合的体系付けを行なうものがある。両方とも10年近く、あるいはそれ以上の歴史をもちながら、特殊な分野で実現されていたにとどまり、最近まで世の注目を浴びることはなかった。

データ・ベース・システムはともすれば、データの万能自動倉庫と目される。これは危険なことである。機能や開発目的な異なるだろうが、いずれのシステムも与えられたデータの管理が目的で、内容は問題にしていない。単なる管理技術にすぎない。単純作業の自動化に較べると技術的には格段にむずかしいが、決してデータの内容や質を左右する性質のものではない。利用者にとってみれば、大切なのはデータの内容と量とであろう。データ・ベース・システムの利用価値は、管理されているデータの質と量によって定まるべきものである。システム設計者にデータ・ベースの利用者が最終的に求めるのは、価値のあるデータをはやく、経済的に供給できるようなシステムをつくりあげることだ。データ・ベース・システムはそのようなシステムを実現するための手段である。

データは形式、表現方法、量、さらに利用目的などが異なることによって、さまざまな形態をとる。多様性の中に共通性を見出しながらまとめられたものが現在あるデータ・ベース・ソフトウェアである。共通性をもたせるためには、それが利用者にとってどんなに重要な条件であっても、特殊な要求は退けられる。これは多様なデータを種々な利用形態を可能にしながら、総合的管理を行なうためには必要なことである。欲をいえば販売管理には販売の特殊性を、在庫管理にはその特性を考慮したデータ・ベース・システムが欲しくなるが、現在のハードウェアやソフトウェアの技術、経済性を前提として、個性的なデータ管理システムを毎回つくることは至難の技である。システム設計者に望まれるのは、共通部分だけを実用化したデータ・ベース・システムを利用して、応用分野の条件を満たすような設計を達成することであろう。

データ・ベース・システムは上手に利用すれば、十分報いられるアイデアであると信ずる。なにぶんにも一夜にしてなるといったものではなく、初期投資と、データの収集・整理の期間とを要する。また小人数の技術者に任せればよいものではなく、データを管理している利用部門も含めた広範囲の人々の理解と援助が、完成への大きな前提となる。計算機利用の分野でだされた単なるアイデアと異なり、多くの成功例があるだけに、データ・ベース・システムは将来の利用技術に強い影響を与えるであろう。

昭和 47 年 3 月

上 條 史 彦

目 次

1. システム設計とファイル	1
1.1 データ・ベース・システムの定義	2
1.2 業務別システムと総合システム	3
1.2.1 業務別システムの利点	5
1.2.2 業務別体系の欠点	6
1.3 総合システムの特徴	15
1.3.1 データ構造とその表現	17
1.3.2 データ独立性	18
1.3.3 データ検索言語	20
1.3.4 データの安全	21
1.3.5 オンライン・システムの応用	22
1.4 データ・ベース・システムの歴史	23
2. データ・ベース・システムの設計	27
2.1 データ・ベース・システムの適用例	27
2.1.1 取り扱う製品の種別と点数	28
2.1.2 入出力形式	29
2.1.3 端末操作	31
2.1.4 在庫高の問合せ	34
2.2 ファイルの設計	35
2.2.1 索引付きのファイル	36
2.2.2 索引の構造	38
2.3 プログラムの設計	40
2.4 システムの拡張	42
2.4.1 プログラムの追加	42
2.4.2 ファイルの拡張	48
2.5 検索コマンド言語	55
3. データ・ベースとオペレーティング・システム	57
3.1 オペレーティング・システムの機能	57

3.1.1	ジョブ・スケジューリング	57
3.1.2	資源配分	59
3.1.3	マン・マシン・コミュニケーション	63
3.1.4	データ管理	65
3.2	レコードとフィールド	66
3.3	記憶媒体としてのファイル装置	67
3.3.1	順次アクセス記憶装置	68
3.3.2	磁気テープの記憶容量	75
3.3.3	磁気ディスクの記憶容量	76
3.4	順次編成ファイル	78
3.4.1	ラベルなしの磁気テープ・データ・セット	79
3.4.2	ラベル付き順次編成データ・セット	81
3.5	索引順次編成	87
3.5.1	索引順次データ・セットの構造	87
3.5.2	索引の構造	89
3.5.3	レコードの追加	90
3.6	区分編成データ・セット	92
3.7	ランダム編成	94
3.7.1	除算法	98
3.7.2	ディジット分析	98
3.7.3	平方数の中央部	99
3.7.4	重合せ	99
3.7.5	基数変換	100
3.7.6	代数的な方法	101
4.	データ・ベース・システム	103
4.1	概 説	103
4.2	データの構造	108
4.2.1	データの論理構造	108
4.2.2	順次構造	110
4.2.3	木構造	112
4.2.4	木構造とポインタ	115
4.2.5	ネットワーク構造	117
4.3	記憶構造	119
4.3.1	階層順次編成	119
4.3.2	階層索引順次編成	120

4.3.3 階層索引直接編成	122
4.3.4 階層直接編成	125
4.4 データの検索	127
4.4.1 索 引	127
4.4.2 索引の構造	132
4.4.3 索引構造の例	133
4.5 データ独立性	134
4.6 データ記述言語とデータ操作言語	137
4.6.1 データ記述言語	137
4.6.2 データ操作言語	146
4.7 データの保護	153
4.7.1 機密保護	153
4.7.2 データの保全	154
5. システム設計とデータ・ベース	155
5.1 システム・イメージ	156
5.1.1 データ検索システム	156
5.1.2 定型処理システム	161
5.1.3 スーパーバイザリ・システム	163
5.2 システム設計への影響	167
5.2.1 データ・ベースの設計	167
5.2.2 システム設計	168
5.2.3 コマンド言語	170
5.2.4 データ・ベースの処理形式	173
参考文献	175
索 引	177

1. システム設計とファイル

計算機を利用しようとするとき、自動車や工作機械と異なり、それが設計されたときの目標にはこだわらない方がよい。計算機を製造する側でも、あまり製品の用途に神経を使っている形跡がない。機械というものが目的に合わせて設計され製作されるという常識からみると、汎用計算機は大変ユニークな存在である。

汎用性とは換言すればそのままでは役に立たない素材であることを意味する。素材を生かし、目的とする機能をもたせるためにはシステム設計と呼ばれる作業が必要である。ほかの機械の場合には設計者が十分配慮して機能と使用目的が一致しやすいようにつくりあげてくれている。計算機は利用者がシステムを設計することによってはじめて目標が定まる。素材を生かすのも殺すのも利用者次第なのである。

システム設計を行なうにあたって、対象業務の内容に関する知識がまず必要となろう。単に仕事の手順をよく知っているというだけでなく、計算機導入後は現行の手順がどう変わるべきかというポイントを見通せる能力が欲しい。計算機を軸とするシステムの未来像を描けることが、新しいシステムを成功に導く。システムの設計作業は計算機側からみると伝票、報告書などの入出力形式と、帳簿すなわちファイルの設計が2つの柱である。処理の流れはどのような作業でも大差ない。入力データを読み、プログラムで処理したうえでデータをファイルに記録し、結果をまとめて編集して報告書として印刷する。計算過程で具体的な問題ともくされるのはプログラムとデータである。

プログラムはテクニックとしての性格が強いので、多くの入門書があり、研究者も多い。一方のデータやファイルについては業務内容と密着していることもあって、統一された記述がなされていない。最近になってデータをソフトウェアの方向から眺め、その扱いを総合化しようとする試みが行なわれている。ファイルの扱い方が統一化手法によって可能となれば、プログラミングやオペレーティング・システムの進歩とあいまって、システム設計に対する新しいアプローチが切り開かれることとなる。ファイル設計とプログラ

ム設計が業務内容とは独立の手法を用いて処理できるのである。

本書は新しいファイルの取扱い手法をシステム設計者の立場から述べたものである。データ・ベース・システムと呼ばれる新手法はプログラム言語、オペレーティング・システム、ハードウェアなどにより種々な制約を受ける。三者が急速な進歩の渦中にある現在、データ・ベース・システムはいまだ流動的である。本書では時代の底流をつかまえて基本的な事項に探りを入れたり、実在しないシステムを設計したりするが、特定のソフトウェアやハードウェアに偏することはその意味から避けた。

1.1 データ・ベース・システムの定義

データ・ベース (data base) という言葉は広範な意味をもっているが、データの貯蔵、取扱い、検索を簡便に、しかも経済的に行なえるような情報管理技術ということができ。描かれた理想像を現在のハードウェア、ソフトウェアによって実現することが技術的にも経済的にも容易でないため、現実利用されているシステムはいずれもその条件の一部を満たしているにすぎない。

どの部分に重きをおいたシステムを前提とするかによって情報管理機能が異なってくるため、詳細を論ずるにあたっては現存するシステムの特色を明らかにしておく必要がある。第4章でデータ・ベース・システムの区分を述べる際いろいろなシステムの特性を明らかにするつもりである。

データ・ベースはまた、データ・バンク (data bank) と同じような意味で使われることもある。データ・バンクでは情報管理ではなく、貯蔵されている情報の内容が問題となる。情報の内容にまで立ち入った議論は本書の目的ではないので、はっきり区別するため、情報管理の技術について述べる際には今後データ・ベース・システム (data base system) と呼ぶことにしよう。

データの内容にまで立ち入る必要がある場合には、一般に図書館学などをひもとかれるがよい。十進分類法その他の有益なシステムがすでに実用化されている。ここでは議論の対象を計算機可読性 (machine readable) をもつデータにしぼる。

1.2 業務別システムと総合システム

現存する事務用の計算機の利用形態を調べてみると、適用業務ごとに独立したシステム設計が行なわれていることが明らかである。業務全体を1つのシステムで処理しきれない場合には、サブシステム (subsystem) がいくつも集まってシステムを構成する。業務ごと、あるいはサブシステムごとに独立している状態を業務別システムと呼ぼう。

システムが独立しているか否かはデータ、特にマスタ・ファイルの共有関係を調べればわかる。同一のマスタ・ファイルがいくつもの業務にまたがって使用されているときに、これをデータの共有状態という。システムまたはサブシステムがデータを共有していないとき、互いに独立していると定義

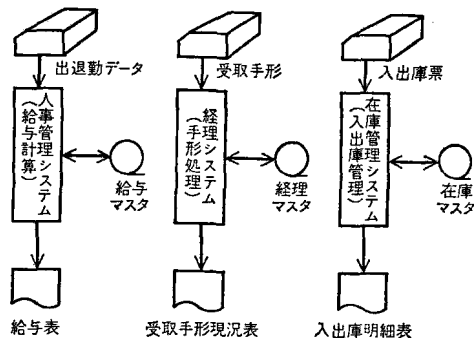


図 1.1 相互に独立したシステム

しよう。現存する適用業務システムに独立形態をとるものが多いというのは、マスタ・ファイルに共通性がないことを意味している。

システムはいろいろあるが、人事管理、経理事務処理、在庫管理などを例としてあげることができる。在庫管理のシステムはそれ自身、経営情報システム (management information system) といったより大きなシステムの一部にすぎないが、また自分の中に多数の独立しうる小さなシステム——サブシステム——をもっている。

受 注 管 理
出 庫 処 理
在 庫 量 管 理
発 注 管 理
入 庫 処 理

などは在庫管理システムの要素であって、それぞれ独立しうるものである。

在庫の管理システムの例では、顧客からの注文が出されてから、出荷に至るまでの道筋をたどることにより、容易にサブシステムの内容をつきとめることができる。図 1.2 はサ

ブシステムの構成を示す例である。

モデルでは営業部門から受注伝票が送り出されると「受注管理」のサブシステムがこれをファイルに記録し、受注報告を出荷部門に伝えている。受注報告を受け取ったのち、出荷部門では適当な倉庫に対して出庫伝票をもって出荷指示する。出荷倉庫には注文された品物が在庫品としてなくてはならないし、運賃などの

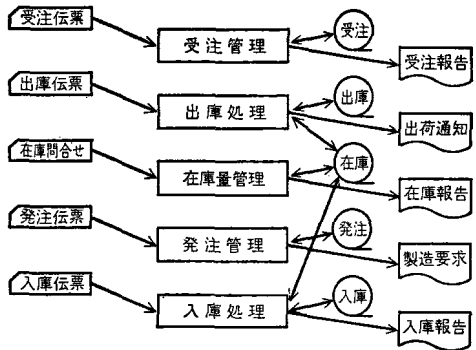


図 1.2 在庫管理のためのサブシステム

条件から、最も引き渡し費用が少ない倉庫を選ばなければならぬ。適当な倉庫を探すのには在庫問合せの作業が必要となることもある。

倉庫では受け持っている地域の需要に最も即した組合せで在庫品目を選ぶ必要がある。在庫費用を小さくするとともに、在庫切れによる余分な納期がかからないように、在庫水準の制御を行なうことも重要だ。「発注管理」では上述のような機能を盛り込み、在庫水準をモニタして自動的に工場へ発注したり、不活発な品目を選び出して報告書を作成する。発注品が倉庫にはいつてくるのを管理するのが「入庫処理」サブシステムの役割である。

例にあげた5つのサブシステムは相互にかなり独立している。マスタ・ファイルに相当するものが図に示されているが、なかには共通に使えるものがある。品目別（倉庫別）在庫水準はその好例であろう。「出庫処理」、「在庫管理」、「入庫処理」の3つのサブシステムは在庫マスタ・ファイルを共通にすると独立ではなくなる。

在庫を管理するためには以上5つのサブシステムを、必要に応じて独立に働かせればよい。一応の目的は達せられる。管理手法はこのままでは単に入庫事務を計算機にさせるというだけの域を脱していない。在庫の問題をさらにきめ細かく管理するには、在庫切れ、倉庫間での融通性、工場発注量のロット・サイズ、納期など2次的な変量を既知の管理データと組み合わせる必要がある。高度な管理データはいくつものサブシステムにまたがって処理されるものが多いので、ここにサブシステムの総合化の問題が発生する。

総合化による利点は観念的にははっきりしている。なぜいままでのシステムはファイルを統合しなかったのだろうか。

1.2 業務別システムと総合システム

問題を取り上げる前提として、従来の独立性の高い業務別システム方式の特色を考えてみよう。

1.2.1 業務別システムの利点

計算機が経営情報処理のために導入されるようになったのは近々10年の間のできごとである。第1章の冒頭にも述べたように、計算機は利用分野の要求を十分考慮して製作されるものではなく、どんな情報処理も行なえるように汎用設計がされている。経験不足と、使用上の自由度のために、いままでシステム設計者は暗闇を手探りで歩くことを強制されてきた。

設計の範囲を限定し、できるだけほかの利用分野と切り離して機械化を進めていくという考え方は、未知の分野に分け入る最良の策であったのである。当初は情報処理の機械化部分が少ないために、少数の機械化担当者にあれもこれもたくさんの適用業務が機械化対象として立候補する事態がもちあがった。システムを独立した部分に分け、システムの開発を段階的に行なう個別独立方式は、開発に必要な労力や費用を期間的に平均化するうえで大きな役割を果たした。計算機の利用システムを開発するためには、業務担当者ばかりでなく、計算機に精通した技術者を多数必要とする。システム・アナリスト、システム・エンジニア、プログラマなどからなる機械化の専門家の数は、どここの企業体でも少数であり、完成した一人前の技術者を新規に雇用することはたいへんむずかしい。業務別システムの方式は、限られた技術的資源を用いて機械化を進めていくためには、是非とも必要ないき方だったということができよう。

開発作業が終了すると、プログラムとデータ・ファイルが結果として残される。計算機の場合、ビルの建設や機械の製造と異なって、努力の結晶は具体的に目に見えるものではない。のみならずプログラムもファイルも比較的簡単に変更することができる。問題は不安定性から発生する。適用業務システムが組織や人間活動に密着していればいるほど——それは事務機械化の効果を高める条件なのである——システムの寿命は短い。極論すれば完成のときからただちに陳腐化への一步を踏み出している。システムの寿命を延ばすためには変更を重ねて時代遅れになるのを防ぐしかない。個別に開発され、互いに独立しているシステムは、システム単位、サブシステム単位での変更に応じられる弾力性をもつ。総合化されたシステム、たとえば一体として設計される経営情報システムがもしあるとすれば、変化に対する即応性をもたせにくだけに、すぐ陳腐化してしまうことだろう。

システムが容易に変更できる、すなわち設計に弾力性があるということは、保守作業の面でも他システムとの関連性を考えなくてすむことを意味する。しかしながら弾力性、保守における個別体系の利点の裏には、改造費用、保守費用がたえず必要となる。どのようなシステム設計をしても、主体がソフトウェアであるかぎり改造と保守の費用から逃れることはできないが、業務別システムの場合には、システムまたはサブシステムの数が増すと、それに比例して費用がかさんでいく傾向がある。

業務別システムの立場をとるシステム設計法の利点をまとめれば、つぎのようになりう。

- (1) 段階的機械化の可能性
- (2) システム単位での変更の容易さ
- (3) 保守の容易さ

どのようなシステムでも完成後時日を経て反省すれば欠点がわかってくるものである。業務別体系はどちらかというと、たくさんのシステムができあがる時期まで見通して採用された技術ではなく、自然発生的な面を有する。最近のように適用業務の数が多くなってくると、当然欠点をあらわしてくる。どのような問題を生じたのか、節を改めて考えてみよう。

1.2.2 業務別体系の欠点

計算機の利用法はハードウェア、ソフトウェアの進歩に伴い急速に高度化している。ハードウェアの分野では等速呼出し記憶装置(ランダム・ファイル, random access memory)が容量、速度の両面で進歩し、高価な特殊メモリの印象から計算機の標準装備へと変身している。最近発表された IBM 3330 型磁気ディスク装置を例にとると、1システムで約8億バイト(byte, 8ビットで1字を表現する)の記憶容量を有し、呼出し時間(access time)は平均約30ミリ秒である。ビット当たりのコストはかつて第2世代の計算機に用いられたものと比較すると、数百分の一になっている。ソフトウェアもオペレーティング・システムの進歩に伴ってデータ管理機能が拡充されてきた。IBM の OS/360 に見られるファイル編成、ファイル呼出しの自動化(access method)はその第1歩であり、内外各メーカーの開発しているデータ・ベース・システムは本格的なデータ管理への幕開けを思わせる。

オンライン・システム(on-line system)はファイルの利用法に新しい分野を開拓した。ファイルの処理サイクルによって、月末処理のファイルでは1ヶ月遅れのデータ、期

末処理では6ヶ月遅れの数値しか得られなかった状態から、一足とびに現況を写すファイルが誕生したのである。オンライン・システムはまた、ファイルの集中、統合を条件としている。総合ファイルは経営情報システムの前提条件でもあるので、両者の利害はよく一致する。

業務別体系では適用業務の範囲が広がるほど、システムの保守費用が大きくなる傾向がある。経営情報システムを具体化するためには、主要な業務をもれなく機械化しないと、十分な効果が期待できないといわれている。業務別方式を用いて機械化の範囲を広げるためには、保守費用は大きな障害となる。会社が計算機による情報処理 (E. D. P., electronic data processing) に投ずる費用に限度があるとすれば、保守費用の増加は新規開発費用の減少を意味し、業務別システムの拡張ができなくなっていくことを暗示しているのである。

磁気ディスク装置はマスタ・ファイル用の記憶装置として脚光を浴びており、ビット当たりのコストも下がってきたが、まだ磁気テープ装置には及ばない。にもかかわらず磁気ディスク装置は適用業務間で共有関係にある (図 1.2 の在庫マスタのような) マスタ・ファイルにとっては、最も適した外部記憶装置 (external memory) である。個別体系を経営情報処理に利用するためには、全体を磁気ディスク装置に記憶させる必要があるが、業務別体系の常として、ファイル間にデータが重複しているため、磁気ディスク装置のコストをもってしてはあまりに高価につく。データの重複はオンライン・システムについては致命的な問題を提起する。少なくとも同じデータ・フィールド (data field) が何度もあらわれるようなファイル体系はオンライン・システムにも向いていなければ、共有式のファイルにも不適当である。

オンライン・システムには別の視点からの問題がある。現在のオペレーティング・システムでは、1つのジョブ (job または run) の範囲をこえてデータをやりとりすることはないという前提でつくられている。オンラインの情報システムのように一方ではファイルの更新 (update) を行ないながら、並行して経営情報の検索を行なうシステムにあっては多種多様なオンライン・ジョブが、マスタ・ファイルに対して働きかける必要性がある。利用者の専用区域を割り当てたら、ほかの区域との交信を許さないという現行のオペレーティング・システムでは逃れることができない限界である。オンラインの業務を並行処理でいくつも処理しようすると、上記の理由からデータ通信端末装置は、1つのジョブにしかサービスできない。結果として遠隔のデータ発生点では、何台も同じ端末装置を用意

しなければならなくなる。

業務別システムを基本とするいき方については以上述べたように、システムの規模が大きくなるにつれて不満が発生している。まとめるとつぎの4点に集約できる。

- (1) 総合ファイル・システムの必要性
- (2) システムの保守コストの増加
- (3) データ・レコードの重複
- (4) 機器、特にオンライン・システムに要求される多重投資

これらの問題をさらにはっきりさせるために、1つずつ取り上げて論じてみよう。

a. 総合ファイル・システム

ファイルが分散している業務別方式では、必要なデータがいずれかのファイルに記録されていることがわかっていても、簡単に取り出すことができない。個々のデータ・レコードが記録されている形式を調べて、それに合わせたプログラムを開発しなければならないからである。記録媒体（磁気テープ、磁気ディスクなど）、記録形式、データの形式などが調査しなければならない項目である。プログラムの開発には、たとえ簡単な報告書を作成するだけのものであっても、プログラム仕様書、入出力データの設計書、流れ図、プログラミングなど多くの工程を踏むので、意外な工程数がかかり、短时日ではできない。要求されるデータの一部が機械が読める形になっていない場合には、早急な要求はあきらめた方が賢明である。経営情報システムが理論としてはすぐれていても、実用化の面で問題とされる理由の1つがここにある。

経営情報システムと従来の業務別システムとの関係がどうなっているか考えてみよう。データを総合ファイルの形で管理することは、前述のような報告書を随時作成するための前提条件でもあるからである。

経営情報システムは会社の組織構造と深いかわりあいをもっている。業務別システムに代表されるいままでのデータ処理の体系が、組織の体系に従って分割されてきたからである。図1.3に経営情報と組織の体系を図示してみた。企業の管理組織は日常の業務の管理を行なう第一線管理者からなるフロント・ライン(front line management)と、運営計画の立案、

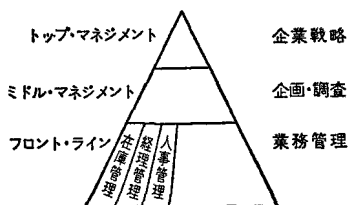


図 1.3 組織と経営情報

企画、調査などを行なう中間的管理者層からなるミドル・マネジメント、企業の将来にかかわるような経営戦略に取り組むトップ・マネジメントの3つの層に分けられる。フロント・ラインに属する管理者の仕事は、自分にまかされた組織を活用して、日常会社を運営していくのに必要な業務を処理することにある。業務ごとに独立した組織となっているので、計算機を利用する場合も業務別システムの形をとる。さきにあげた在庫管理、経理事務、人事管理などはすべてこの範ちゅうに属する。

部門間の予算、人員計画を調整したり、市場予測を行ったり、長期にわたる生産計画や販売計画を立てるといった仕事はミドル・マネジメント (middle management) の受け持ちである。会社の組織が大きくなると、企画・調査的な仕事は独立して1つの部門を形成することがあるが、スタッフ部門ができたところで、各ラインの中間管理者層に対する要求は本質的には変わらないであろう。スタッフ部門が独立すれば、より全社的な見地から仕事をするようになるだろうから、個々の部門独得の企画的業務はその部門に残されてしまうからである。いずれにしてもミドル・マネジメントの要求する情報は、フロント・ラインと異なり業務別システムのわくをこえるもの、時間的に過去の歴史的データにさかのぼるもの、一時的にデータの組合せを変えたり、処理方法を変えたものなどが多い。個別に設計されたシステムから一定の時間間隔 (処理サイクル) に従って出力される定期的報告書では満足してもらえない。

1番上のレベルはトップ・マネジメント (top management) である。企業戦略の決定がここで行なわれる。新しい工場の建設や新製品の発売、会社間の各種契約の締結などのように、企業の存立、将来性を定める事項は最高レベルのマネジメントの決定にゆだねられる。トップ・マネジメントに対しては、会社の運営状況を示す情報を集約したもの、企画された事項の最終草案といった報告書が要求される。特に問題となるのは企業体外の情報、たとえば政府機関の動き、議会における情勢、国際的な経済の動きなど通常の経路からは計算機に達しないものが大きな比重を占める点である。単に収集しにくいばかりでなく数量化しにくいし、定形的なファイルを用意することもできない。図1.4は以上述べた経営情報システムで計算機用のファイルがどんな位置を占めるかを図示したものである。

業務別システムで処理される部分がピラミッドの最下部を占める。在庫管理、経理、人事管理などはさらに細かなサブシステム、受注、在庫、生産指示等々に分かれている。入力データは伝票であったり、オンライン・システムの端末装置であったりする。サブシステムはそれぞれ適当な処理サイクルをもってデータ処理を行ない、結果として定期報告