

50-R005

経営計画情報システムに関する 調査研究

—経営情報調査報告書(Ⅶ)—

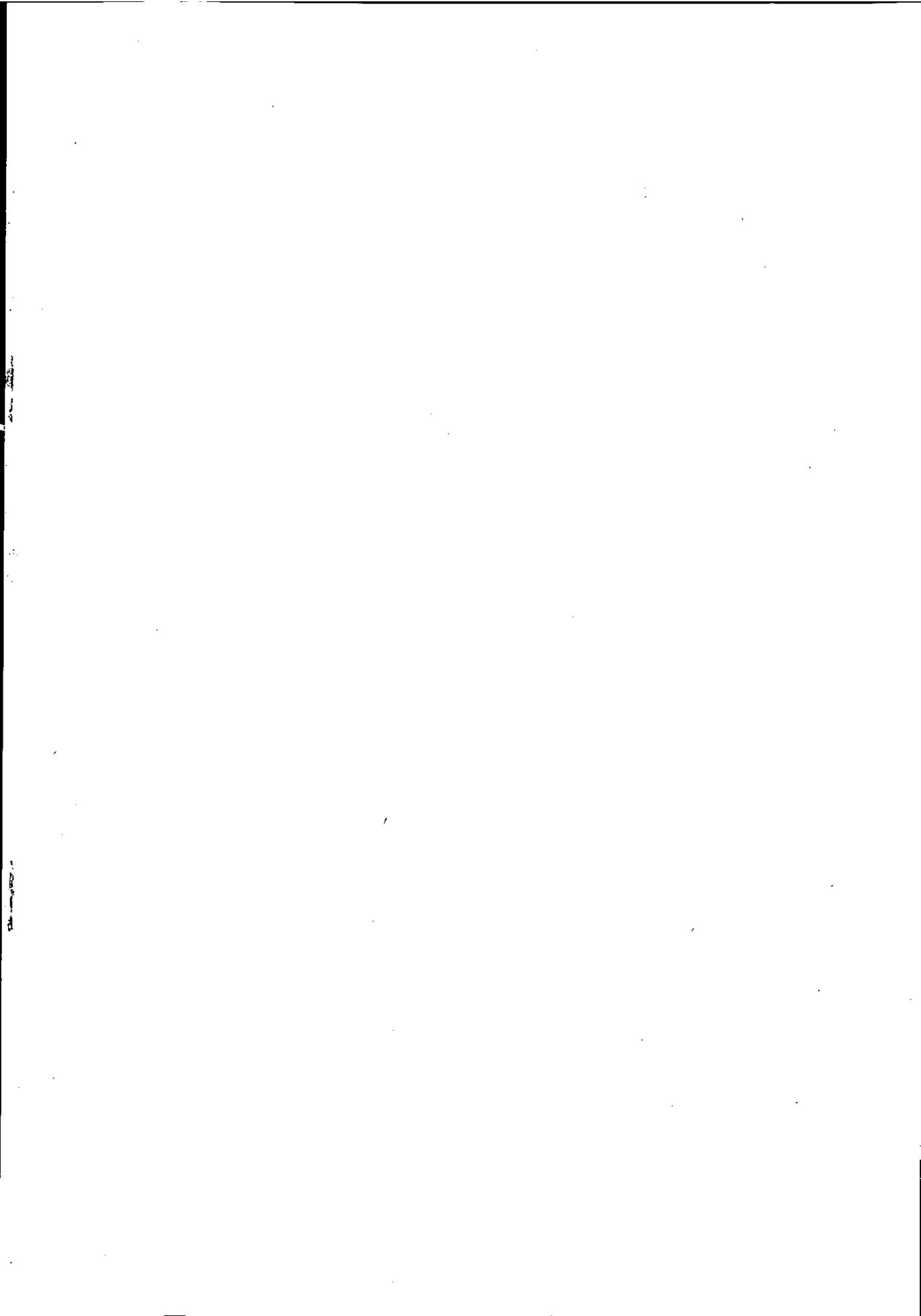
昭和51年3月

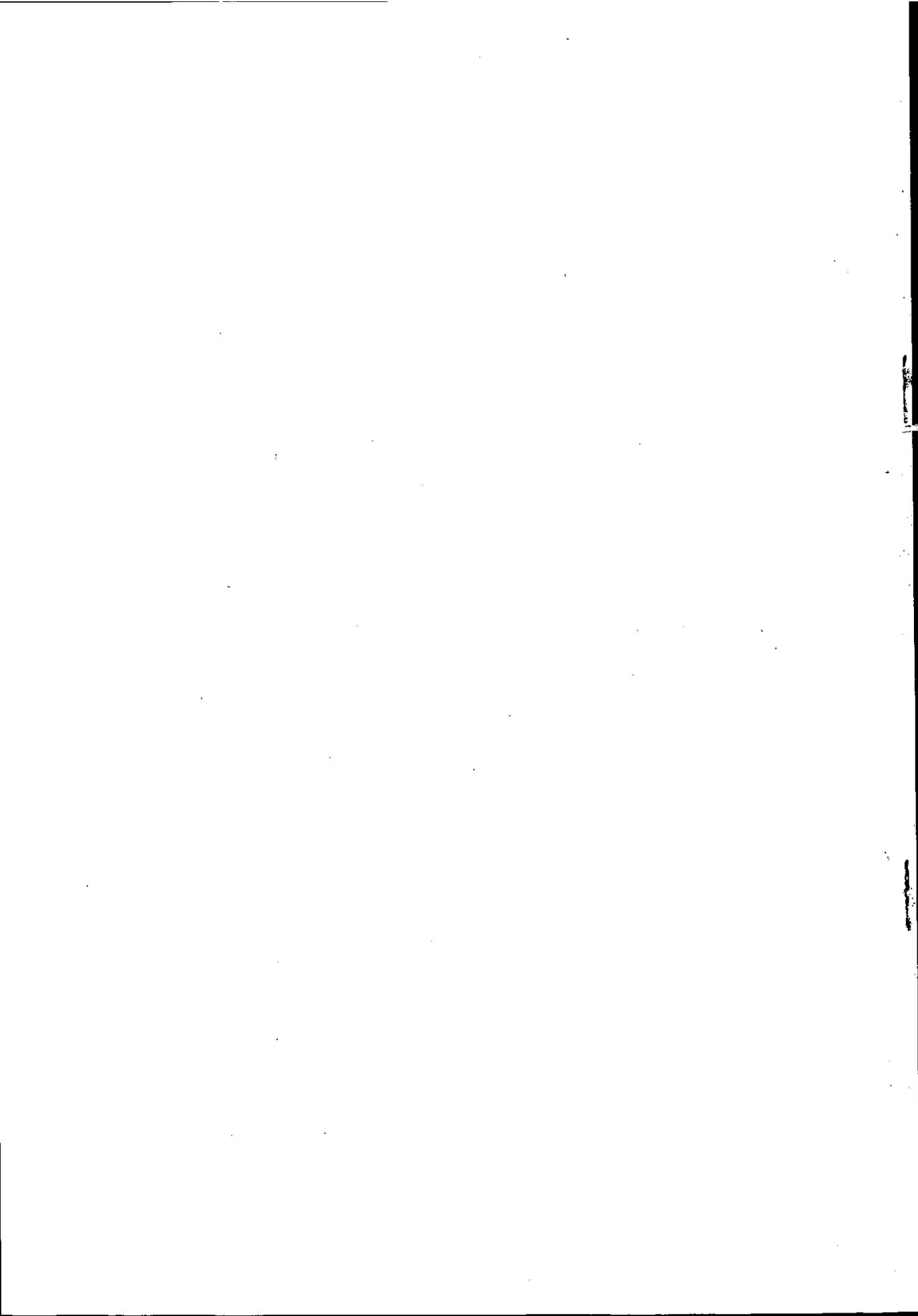
JIPDEC

財団法人 日本情報処理開発協会

JIPDEC
50
R005

この報告書は日本自転車振興会から競輪収益の一部
である機械工業振興資金の補助を受け、昭和50年度
に実施した「経営計画情報システムに関する基本問題
の調査研究」の結果をとりまとめたものであります。





序

わが国の企業が指向する経営情報システムはオペレーショナルレベルのシステム管理および経営計画レベルのシステムへと開発が行われつつありますが、経営計画の策定における経営科学手法を中心とする意志決定支援システムについては、必ずしも有効に利用されているとは言えない状況にあります。

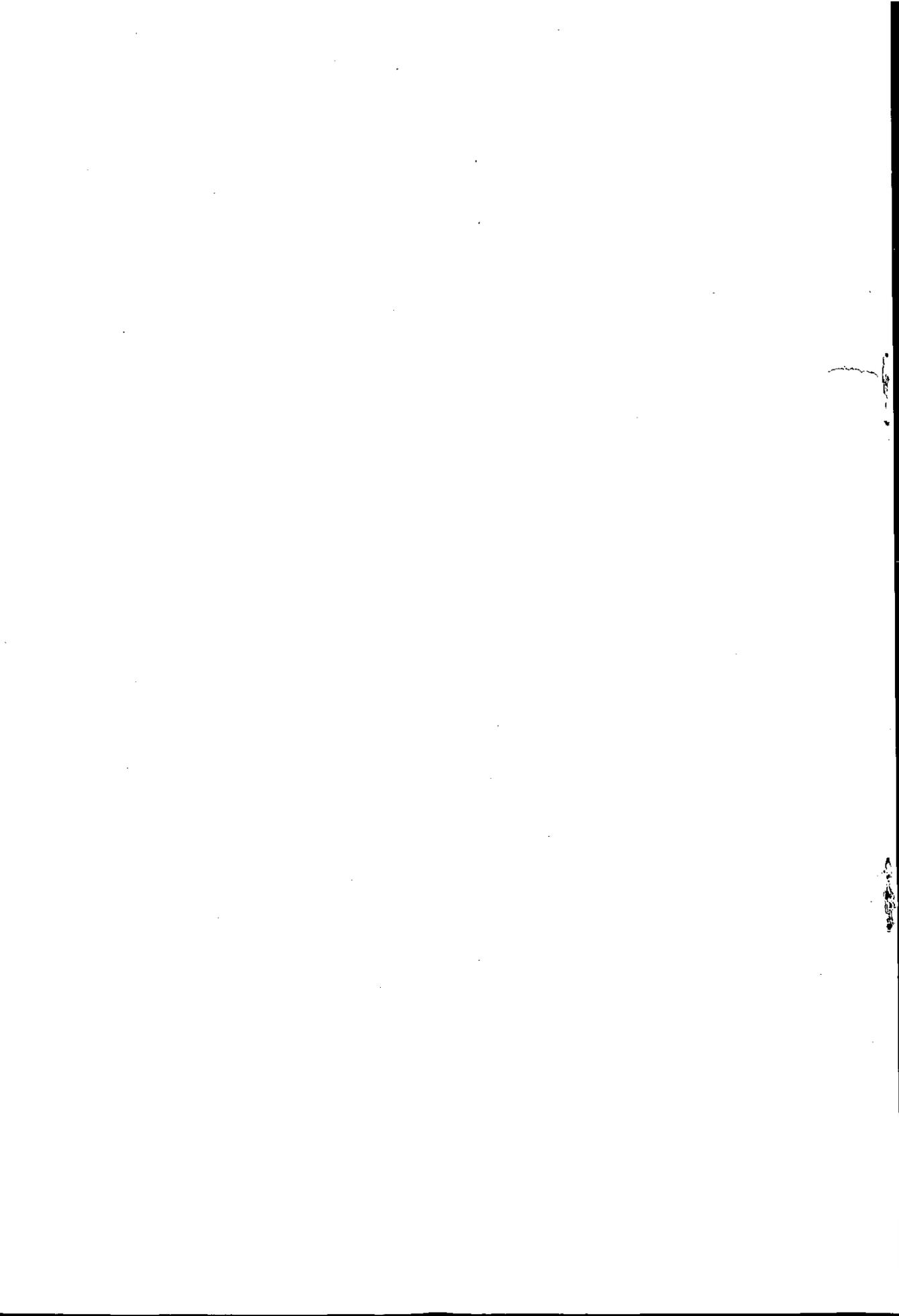
当財団では、本年度は企業の経営計画の意思決定段階におけるコンピュータの利用状況を調査し、情報処理の理論的諸問題について研究するとともに、モデル企業内の経営計画情報システムの開発を実施いたしました。本報告書はこれを取りまとめたものであります。

ここに本調査実施に際しご協力いただきました経営計画情報システム調査委員会委員、ならびにアンケート調査、面接調査にご協力いただきました各位に心より感謝の意を表しますとともに、本報告書が各方面に利用されることを念願する次第であります。

昭和51年3月

財団法人 日本情報処理開発協会

会 長 植 村 甲 午 郎



目 次

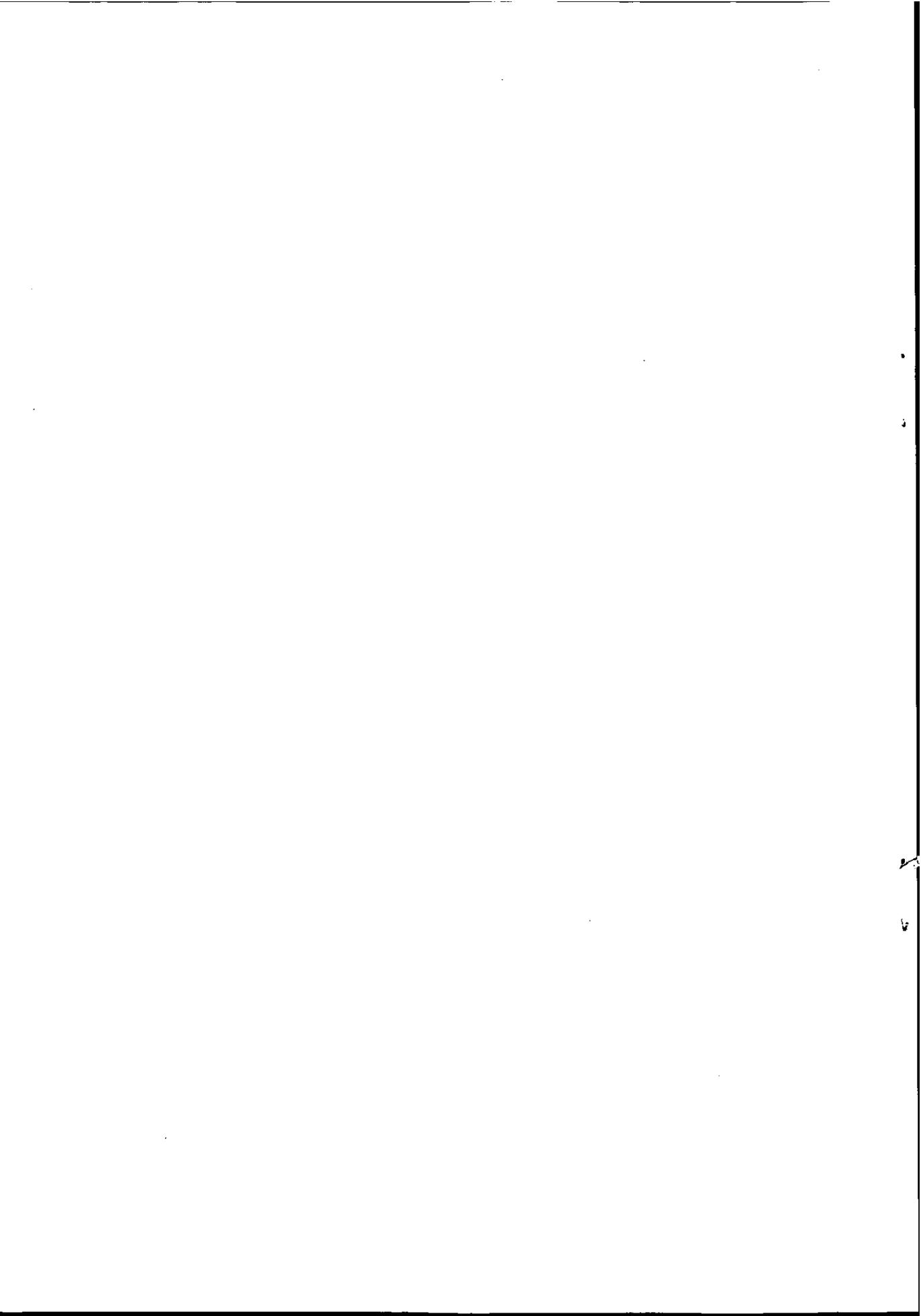
1.	調査研究の概要	1
1.1	経営計画情報システム調査の目的	1
1.2	調査研究の項目	2
1.3	経営計画情報システムの今後の課題	3
1.4	調査委員会	4
2.	コンピュータ・ソフトウェアとしての 経営意思決定サポートシステム	5
	はじめに	5
2.1	コンピュータ利用形態の発展	5
2.1.1	コンピュータ処理の適用対象	5
2.1.2	いくつかの試み	5
2.1.3	データ管理システム	6
2.1.4	構造化プログラミング	6
2.1.5	ディスプレイ装置による入出力	8
2.2	経営意思決定サポートシステム	8
2.2.1	経営計画	9
2.2.2	経営計画過程での情報処理	11
3.	経営計画情報システムの設計	17
3.1	システムの設計基本方針	17
3.1.1	解 説	17
3.1.2	情報処理機能による分割	17
3.1.3	分析手法, 言語とモデルの区別	18
3.1.4	会話的入出力	19
3.1.5	モデルの管理	20

3.2	システム構造	21
3.2.1	システム構造の概要	21
3.2.2	各サブシステムとその機能	21
3.2.3	メイン・ルーチン	23
3.3	データ・ベース管理	24
3.3.1	データ・ベース管理	24
3.3.2	データ・ベース管理システム	27
3.4	データ管理の理論	29
3.4.1	データ構造	29
3.4.2	記録構造	32
3.5	MDSのためのデータ管理の方法	34
3.5.1	MDSで扱うデータ	34
3.5.2	データ構造, その他	38
3.5.3	記録構造	45
3.5.4	データ管理の内容	47
3.6	データの入出力	48
3.6.1	記録媒体	48
3.6.2	データの入出力の単位	49
3.7	レポート作成, 情報検索	50
3.7.1	レポート作成の手順	50
3.7.2	レポートのフォーマット	51
3.8	システムの管理	52
3.8.1	データの機密保護と保水性	52
3.8.2	記録媒体の管理	53
3.9	データの分析・処理	54

3.9.1	データの分析・処理	54
3.9.2	演算・関数	55
3.10	モデルの管理	63
3.10.1	モデルの表現	63
3.10.2	モデルを定義する	66
3.10.3	モデルの管理と実行	67
3.11	使用者とシステムの会話的入出力	68
3.11.1	端末装置の利用	68
3.11.2	会話の理論構造	69
3.12	システムの機能と利用方法	71
3.12.1	システムの提供する処理ツール	71
3.12.2	利用方法	76
3.13	まとめ	77
3.14	カード・ベース・システム	78
3.14.1	カード・ベース・システムの目的	78
3.14.2	カード・ベース・システムの機能	79
3.14.3	設計手順	81
3.14.4	処理ツール	82
3.14.5	M O D E L	88
3.14.6	システム構造	91
3.14.7	記録する情報	94

4.	経営計画情報システムに関するケーススタディ	97
4.1	経営計画情報システム形成に関するアンケート調査	97
4.1.1	経営計画システムの構成	97
4.1.2	個別計画におけるコンピュータ利用	100
4.1.3	計画用データの維持管理	103
付 表	経営計画情報システム形成に関するアンケート調査票...	108
4.2	経営計画プロセスにおけるコンピュータ利用	118
	は じ め に	118
4.2.1	需要予測と販売計画	119
4.2.2	需要と供給能力 — 需給計画	120
4.2.3	容器, 原料, およびその他の材料の受払い計画	120
4.2.4	その他の計画	123
4.2.5	長期計画プロセスにおけるコンピュータ利用	123
	参 考 資 料	126

1. 調査研究の概要



1. 調査研究の概要

1.1 経営計画情報システム調査の目的

1960年代に各方面から主張された「MISへのアプローチ」は、それ以前におこなわれていたコンピュータを各個別処理ごとに利用するというやり方を発展させ、より積極的に全体的情報処理体系として構成しようという試みであった。しかし、その後、当初の目的を完成させるにいたらず、数々の失敗事例が報告され、問題点が指摘されてきた。この原因は、基本的には、コンピュータ・システムの適用対象である一般的な情報処理と処理システムとの相互関係を詳しく分析することをせず、そのため、各々の適用対象における情報処理の特徴を明らかにし、それに基づいて処理システムの設計に適当な工夫をし、またそのために必要な技術的問題を開発するといった作業が伴わなかったことにあるといえよう。

各経営体における経営計画策定過程における情報処理を適用対象とする処理システムを研究対象とするのは次の理由により意義のあることである。すなわち、第一に、経営計画策定過程はそれ自体、各経営体における根本的な行動内容を規定する重要な領域であり、第二に、一般的な情報処理と情報処理システムの相互関係に特に工夫を必要とする典型的な分野だからである。経営体における全ての情報処理を体系化し、その中にコンピュータによる情報処理を位置づけるという意味での経営情報システム(MIS—Management Information System)を完成させるには、計画過程における情報処理だけを対象としていては不十分である。しかし、予測される諸々の問題点を十分考慮した上で、経営計画過程における情報処理が他の分野のシステム、あるいは全体システムを考察する上でもっとも基本になるであろうという展望をもつことはおそらく間違えてはいないと思われる。

経営計画過程における情報処理を対象とする処理システムについて、現在、ある確立した理念があり、我々もその考えを十分一般に妥当するものとして支持できると考える。それは、経営計画における情報処理の基本は、その計画を担当する人間の意思決定であり、情報処理システムはその意思決定を支援（サポート）することが目的である、という考えである。それ故、このようなシステムを、経営意思決定サポートシステム（MDS— Management Decision Support System）と呼ぶのである。我々の研究もこの考えを基礎にして行う。問題は、経営計画過程において、人間の創造的な分析能力、適切な意思決定を効果的にサポートするには、どのような処理システムが望ましいか、ということである。

1.2 調査研究の項目

- (1) 各企業における経営計画の仕様およびコンピュータの利用の仕方については2つの方法により調査した。1つは、特定の企業について、計画の策定方法、使用される情報、コンピュータの利用方法等についての詳細な調査である。これは、麒麟麦酒株式会社の協力を得て行ったものである。もう1つは、多数の企業について、同様の事項についてのアンケート調査である。
- (2) MDSに関係する理論的問題については、すでに開発されている、いわゆるMDSをはじめ、情報管理システム、企業モデル等のシステムの設計仕様、およびそれに関する文献、論文等を検討した。
- (3) MDSについての理論的研究、およびそれに基づくシステム設計については、経営計画における情報処理の特徴、システムに要請される機能、性質を分析し、それに基づいてMDSとして望ましいシステム設計の方法を考察した。
- (4) MDSの理論的研究、設計仕様を前提にして、カードだけを使用者からの入力媒体として利用するシステムについても考察した。これについては、具体的にシステムを作成し、それに必要な外部仕様書、内部仕様書を作成した。

1.3 経営計画情報システムの今後の課題

- (1) 経営計画情報システム形成の諸々の理論的問題を検討した結果、次のような基本的な考えを提案することができると思う。すなわち、経営計画過程での情報処理を適用対象としたコンピュータ・ソフトウェアとしての情報処理システムは、使用者の問題設定、ニーズから独立にしておくことが望ましいということである。経営計画それ自体が高度の情報処理体系として設定されるのだが、それに対処するコンピュータ・ソフトウェアとしての情報処理システムは、その情報処理体系の論理構造と機械的に対応させるのではなく、全く独立に処理システム自体の論理構造をもっているものであるべきである。そして、このためにシステムの情報処理機能による分割と、使用者とは独立な分析手法、言語を基にしたデータの分析、処理ツールをもつことが重要である。
- (2) 一般的な情報処理体系と情報処理システムの論理構造は互いに独立なものとしておく。そして、この二つの体系の接点に特別の工夫をすることが次の提案である。接点の工夫とは、要するに、使用者が意図する処理内容を簡単な入力作業をもって、処理システムにそれを入力することができるようにしておくことである。このために、次の二つのことを提案する。一つは、使用者のおこなう情報入力を、あらかじめシステムの側で解析しておき、質問-応答の列として表現する、そしてそれを端末装置を使用して、逐次のおこなうことである。二つ目の提案は、使用者の記述した処理内容を示す情報を、システムに記録しておくという機能をもつことである。使用者は以後、類似する処理は、その情報を指示することにより、容易に実行させることができるようにしておく。
- (3) このシステムの技術上の決め手は、各々、性格の違い処理のために、統一したデータの表現形式を用いることができるかどうかということである。そして、この表現形式は、使用者がコンピュータとは離れて、日常用いている表現形式とも一致している必要がある。このために、一般のデータ・ベース管理システムとは多少異なるデータ管理の方法を提案することにした。

(4) 結局、MDSのもつ問題は、従来の個別問題に対応したプログラムでは適当でないことにある。使用者の側のニーズが多様で、複雑だからである。一般的な情報処理がいくつかの問題を含んだ体系であることも重要である。こうした事実の上に、コンピュータ・ソフトウェアとしての情報処理システムの設計の方法が問われているのである。そして、経営計画過程での情報処理は、その典型的な分野なのである。我々の研究は、この課題に対する一つの解答であると思う。

(5) 今後の課題は次のとおりである。一つは、業務レベルにおける情報処理を含めて、各企業の全体的情報処理体系を考察し、その中でMDSの占める位置を具体的に明らかにすること、それから経営計画過程における一般的情報処理を考察し、処理システムとの関係を探究し、処理システムの有効性を検討することである。

1.4 調査委員会

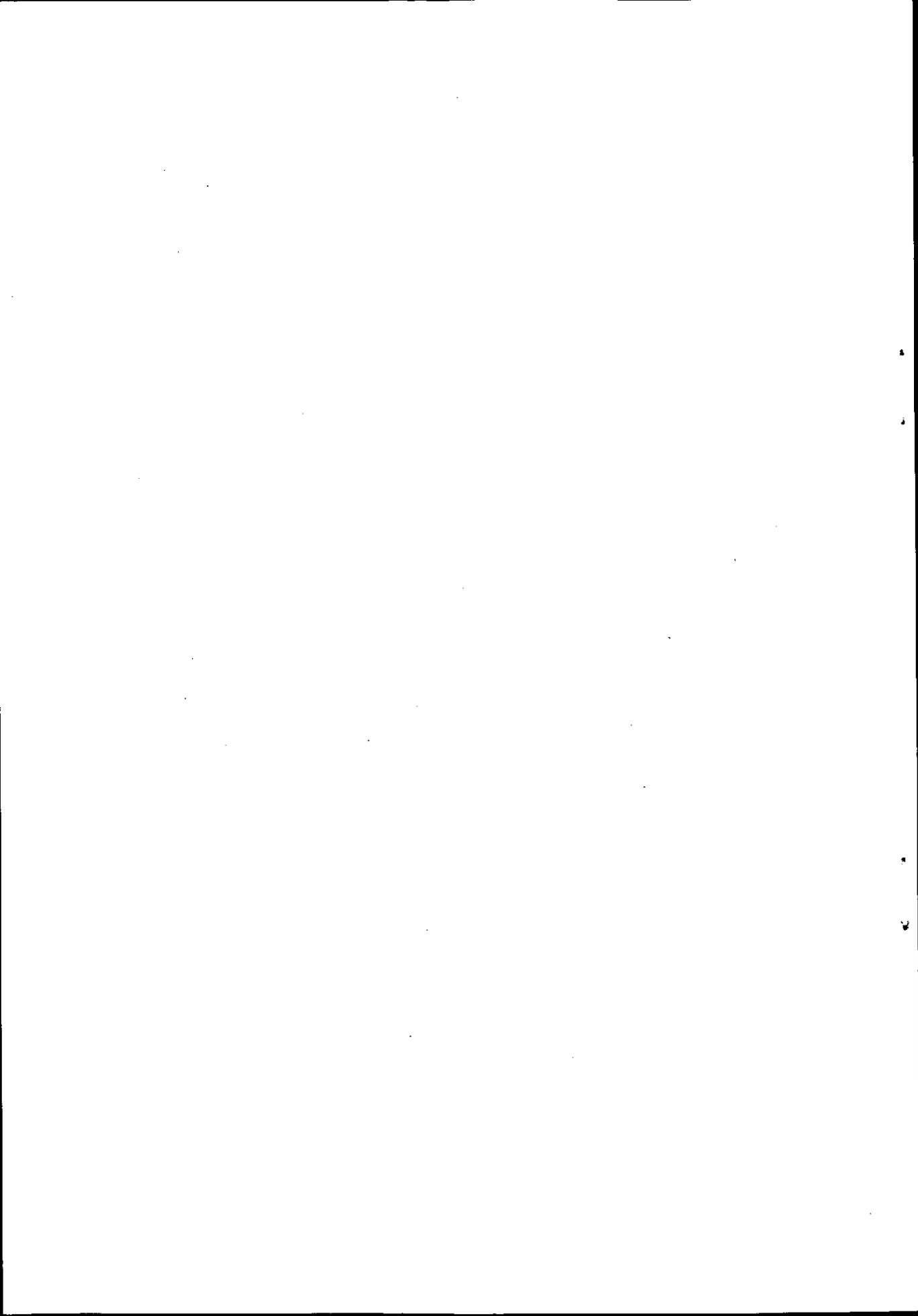
本調査を進めるにあたり、経営計画情報システム調査委員会を設置した。

経営計画情報システム調査委員会構成

(敬称略 順不同)

委員長	宮川公男	一橋大学商学部教授
委員	岸人宏亘	麒麟麦酒株式会社 企画部副部長
"	山田豊彦	同上 経理部管理課長
"	山中有二	同上 事務管理部システム課長
"	高橋三雄	成蹊大学経済学部助教授
"	丹下忠之	一橋大学商学部大学院
"	市川隆	(財)日本情報処理開発協会 技術調査部 調査課長

2. コンピュータ・ソフトウェアとしての 経営意思決定サポートシステム



2. コンピュータ・ソフトウェアとしての 経営意思決定サポートシステム

はじめに

この報告書は、経営意思決定サポートシステムとして、我々が提案するシステムの設計の概要を説明するのが目的である。このために、関連する諸問題を検討しつつ、我々の主張を展開するという方法をとった。それで各部分において、理論上の問題と設計概要とが混在しており、多少整理不足であるかもしれない。厳密な設計仕様書ではなく、設計仕様書の基礎になる考え方と技術上の問題を説明するのが目的であるから、この点御了承いただきたい。

2.1 コンピュータの利用形態の発展

2.1.1 コンピュータ処理の適用対象

一般に、各経営体において、広い意味での情報処理という作業がある。そして、その作業をいろいろな意味で効果的であるように、コンピュータを利用することを考える。そしてそのとき、その対象である情報処理の内容と関連させつつ、処理の一部または全部を担当するコンピュータ・ソフトウェアとしてのシステムを設計しなくてはならない。同時に、適用対象である一般的な情報処理の様式、論理構造と情報処理システムのそれとは、どのように関連し、その接点はどのようにしておくべきかということをあわせて考察することになる。つまり、適用対象によってシステム設計に適切な工夫を必要とする。

2.1.2 いくつかの試み

コンピュータは、実現以来現在までに著しく進歩してきたといわれている。

しかし、それら進歩の内容を見れば、ハードウェアとそれに近いソフトウェア（主としてオペレーティング・システム）に限られていて、使用者の利用方法については、現在まで本質的に変わらないある方法が続いてきたのである。それは、極端に言えば、システム設計者が使用者の要求を厳密に理解し、個別の問題ごとに各々固有のシステムを作成するというやり方である。現在でもこの方法は続いている。しかし、必ずしも経営計画の分野ではないが、使用者の側からみた、異なったコンピュータの利用形態を生み出すであろうと思われるいくつかの試みがある。

2.1.3 データ管理システム

第1は、データの管理を基本機能とする汎用システムを作成、使用し、問題別アプリケーション・プログラムの作成に際しては、この部分を個別に作成するのではなく、データ管理システムの提供する処理機能を利用するというものである。このようなシステムはデータ管理システム、情報管理システム等と呼ばれるものである。データ管理システムはアプリケーション・プログラムの中で利用されるだけでなく、データの記録、検索、報告書の作成等、データの記録装置それ自体としても利用される。データ管理システムについては、その問題点を後で詳しく議論するが、ここで注目しておくべきことは、従来の個別問題に対応したシステムの作成ではなく、情報処理機能に対応してシステムを作成するという基本的考え方である。さらに、コンピュータによる情報処理の流れは、例えば、データの入力、処理、出力というような、固定したものではなく、いくつかの処理機能がランダムに使われ、1回の処理において、単にデータを入力するだけ、あるいは出力のみといったことも頻繁にあり得るということである。

2.1.4 構造化プログラミング

第2は、E. W. ダイクストラによって提案された構造化プログラミング

という考え方である。[※] 構造化プログラミングの提案は、プログラムというものについての厳密な検討を主張したことに意義がある。現在、大量のプログラム(システム)が作成され、利用されているが、作成されたプログラムの処理内容についての正確さを論理的に証明する手段がないのがほとんどである。あるデータを入力し、しかるべき出力であるか否かをチェックすることによって、その判定をしているだけである。そのプログラムが利用されている間に入力される可能性のあるデータをすべて検証することはもちろんできない。構造化プログラミングは、ある情報処理を正確にプログラミングするための方法についての提案である。その具体的な内容については、現在多くの議論があり、今後発展していくであろう。いわゆる GO TO ステートメントの是非などはいろいろな見解がある。しかし、多くのコンピュータ・ソフトウェアの分野の専門家の間で一致しているのは、プログラミングという作業を再度反省し直し、正確な処理を記述するための方法を探究すべきであるということである。従来の、いわばトライ・アンド・エラー (try and error) のプログラムの作成は反省されねばならず、プログラミングあるいはシステム設計を正確さ、安定性、コスト等を十分考慮して、理論的基礎の基におこなうことが大切である。このようなことは特に、大規模なシステムを作成するとき問題になる。大規模システムの作成に際しては、プログラミングそれ自体のコストを考慮して、システムの開発、維持、修正の方法と利用上の方法との関係を吟味する必要がある。

プログラムそれ自体の論理構造は、処理の対象となる問題の論理構造と機械的に対応させる必要はない。使用者は、単なる利用上の問題からのみ、システムを考えるのではなく、システムそれ自体の作成にも留意しなくてはならない。

※ 参考： Dijkstra, E.W., Notes on Structured Programming, "Structured Programming", pp. 1-82, Academic Press, London.

2.1.5 ディスプレイ装置による入出力

第3は、グラフィック・ディスプレイ (graphic display) , キャラクター・ディスプレイ (character display) 等の端末装置を利用した入出力である。端末装置を利用すると、使用者が意図する処理を入力してから、出力結果が手元に届くまでの時間 (turn around time) が非常に減少する、使用者が入力作業を容易にできるようにする等の実際効果がある。それに加えて、プログラム上の処理の論理構造 (実行の流れ) を多様にしておくことができるようになるという効果がある。システムと使用者はダイナミックに対応し、処理内容を逐次的に変更するという使い方ができるようになる。

以上のようないくつかのアプローチは、結局、高度の情報処理体系を目的とする、コンピュータ・ソフトウェアの側の発展であるといえよう。情報処理の内容が複雑になると、それに対処する処理システムも、いろいろの条件を考慮して、組織的な体系として作成することを要請される。この組織的な体系としての処理システムと一般的な情報処理との相互関係およびその接点を詳しく分析することが、今後の処理システムの設計上の課題である。ここで、展開する経営意思決定サポートシステムも、このような試みの一つである。すなわち、システム作成上のコスト、利用上の容易さ、システムの維持、修正等を考慮して、使用者の側からみた適切な工夫をすること、この観点から全体的情報処理体系とコンピュータ・ソフトウェアとしての処理システムの論理構造の相互関係およびその接点を考えることである。

2.2 経営意思決定サポート・システム

経営意思決定サポート・システム (MDS—Management Decision Support System) は、経営計画策定に関する情報処理を適用対象としたコンピュータ・ソフトウェアとしてのシステムである。この節では、前節での「使用者の側からみた適切な工夫」が、経営計画策定過程における情報処理

の場合、どのように具体化されるべきであるか考える。

2.2.1 経営計画

(1) 経営計画は後に詳述するように定型化された処理内容をもつ部分が少ない。もちろん、個別企業の業種、規模、市場環境等によって異なる。ここでは以後の説明のために、経営計画の内容として、概していえることを確認しておく。

経営計画は高度の政治的判断を必要とするものと、いわゆる経済分析、経営分析によるものがある。前者を政治的意思決定、後者を経営的意思決定と呼ぶ。

政治的意思決定は、概して、一回限りの意思決定、非数量的な情報を主として利用する高度の戦略的意思決定といった特徴がある。

経営的意思決定は、数量的情報を基礎に、生産、販売、財務等の具体的な行動内容を明らかにし、経済学、経営学上の分析尺度によって企業活動を追求する。両者はともに関連しあう。MDSは、この中で、経営的意思決定において主として利用する情報の生成を目的とするが、この中の情報は、政治的意思決定においても、参考資料として利用されるものも多くある。

(2) 経営的意思決定の対象になる計画は、個別計画（部門別計画）と全社的一計画が生まれ、相互に関係づけながら政策決定を行う。個別計画は概略次のものから構成される。

- ① 需要予測
- ② 販売計画
- ③ 生産計画
- ④ 在庫計画
- ⑤ 設備投資計画
- ⑥ 要員計画
- ⑦ 研究・開発計画
- ⑧ 財務計画

全社的統一計画は、長期計画、短期計画がある。短期計画は、通常一年を単位とすることが多く、実際は予算の作成が主たる内容である。予算は期間利益の観点から、企業活動を総合的に計画するもので、最終的には見積損益計算書、見積貸借対照表として示されることが多い。短期計画は、長期計画によって与えられた計画を前提とするので、変動的な部分の見積り、調整を特に重視する。そのため、需要の予測（短期的変動）と原価計算に作業の中心がおかれる。長期計画、短期計画は、分析視点、取り扱い問題が異なるのであるが、両方で同じような情報を期間や視点を変えて作ることも多い。これらの情報は、各企業のその時々状況によって、いろいろであるが、一応の参考のために、基本的なものを例示しておこう。

- (a) 見積損益計算書： 利益目標くりかえし、パラメータ感度分析、売上げ、コスト変更の影響の分析
- (b) キャッシュ・フロー（cash flow）： 資金源泉、用途の探究、割引キャッシュ・フロー、投資収益、資本支出、負債調達、税率、支払計画等の設定
- (c) 見積貸借対照表： 財務比率
- (d) 全社的収支計画
- (e) 原価見積
- (f) ロジスティックス計画： 加工、処理工場の場所、生産能力、在庫能力、原材料入手先、得意先分配等。
- (g) 人員計画： 生産計画、人員数、残業量、コスト等。
- (h) 国民経済動向予測： 国民所得、消費支出の予測等。
- (i) 製品の需要予測

以上のような情報の他に、長期計画に個有のものとしては、次のようなものがある。

- (j) 設備投資計画についての利益、コスト等。
- (k) 製品計画に関する技術情報、市場動向等。

このような情報はごく一部であり、またこのような情報を生成する過程でさまざまな分析がおこなわれ、関連する情報が生成される。そして、全体としての企業行動を考察し、政策が策定されることになる。

2.2.2 経営計画過程での情報処理

(1) 意思決定のサポート

我々が経営計画過程における情報処理システムを考えるときに、第一に注意すべきことは、経営計画という広い意味での情報処理の中で、処理システムの担当する役割を正しく位置づけることである。

従来、計画立案そのものを全面的に自動化させるというアプローチがあった。そして、その失敗が各方面から指摘されている。計画過程では、その企業の内部の状態、環境の分析、問題の発見、代替案の作成、評価、選択等々の一連の作業が行われる。この場合、人間の分析能力の果す部分は重要である。企業の政策立案において、事務機械化、経営科学の手法だけに限定できないことは明らかである。それ故、情報処理システムは、経営計画担当者またはそのスタッフの分析、意思決定との関係において、果すべき役割を考えるべきだということになる。

全ての情報処理は他の情報処理と入出力関係をもつ。情報処理の中には、コンピュータによって行われるものもあるが、それが全てではない。とにかく、情報処理システムは他との関連において、考えなくてはならない。この意味で、全ての情報システムは、それだけで成り立つものではない。しかし、計画過程で使用する処理システムの場合、単なる入出力関係の問題ではなく、使用者の意思決定と有効に結びつくためには、計画過程の情報処理個有の特徴がある。これが、特にMDSについて考察する場合の基本的な問題点なのである。以下、この点をみていこう。

(2) 多様な情報処理

経営計画過程における情報処理は、いわゆる非定型的な問題が多く、そのた

め、処理手順をあらかじめ想定しておくことが困難である。計画過程では、問題の発見、予期しなかった環境の変化への対処等を含み、そのためその都度問題を定式化し、望ましい行動を探るという作業が重要だからである。

さらに、ある一つの問題について、例えば需要の予測をする場合等、そのために必要な情報の生成、分析の方法は、そのときの問題意識、環境等によってさまざまに変わるであろう。むしろ、問題の種別、あるいは分野と分析手法とは独立なものであって、これらを固定的に関係づけるべきではない。ある一つの問題について、異なる処理によって生成された情報を比較検討するのも意味のあることである。

従来、経営計画は各問題別に考える傾向があった。多くの場合、経営組織における各部門が対応する問題についての計画を策定して、会議等の何らかの方法で相互調整をおこなうという様式が多かったようである。今日重要な課題は個別計画相互の関係を明示的に定式化し、体系的な計画を策定することである。

全社的な体系的計画は、そこに含まれる変数の数、それらの相互関係の複雑さを考えると、大規模なモデルにならざるを得ない。大規模なモデルほど、定式化することが困難で、一般に妥当する固定的なものを設定しにくく、そのため、常にモデルの検討、改善を必要とする。このため、大規模なモデルの作成は、ある意味では、常に完全なものではないと考える方がよい。完全なモデルが完成するまで待つのではなく、漸次改善を加えてより分析能力のあるものに発展させる方がむしろ一般に妥当する方法である。

経営計画の策定方法、必要な分析は、それ自体その企業の発展、環境の変化によって漸次変化発展するものである。さらに、計画に必要な知識、技術も発展していくものと考えておくべきである。したがって、処理システムはそのような変化に耐え得るものであるべきである。経営計画の改善に際して、処理システムの変更を必要とすると、どうしても妥協的になる。後に詳述するように、システムの設計、プログラミングそれ自体がコストのかかる作業であることを認識しておくべきである。

経営計画策定はそれ自体が企業にとって重要な仕事である。そのため、常にそのやり方を改善することを試みるのが望ましい。この意味で、ある程度の長期間をとって考えると、その処理内容はやはり多様なのである。処理システムは、長期間の使用に耐え得るものでなくてはならないから、計画それ自体の発展を十分考慮して作られる必要がある。

(3) 使用者が容易に利用できるシステム

コンピュータの処理システムは、可能な限り、コンピュータについての専門知識をもたない人によって直接利用できることが望ましい。特に、MDSの対象とする計画策定過程では、処理が定型化されていることが少なく、使用する人の意図が正確に、煩雑さを伴わずに、システムに入力できることが大切である。この意味では、業務上のルーチン化された処理とは異なる。

使用者にとって、コンピュータについての専門知識を必要としないことと合せて、入力作業が繁雑でないことも重要な条件である。

使用者が容易にコンピュータを利用するためには、完全に処理内容が固定している処理システムに対し、データの指定だけを行うという方法もある。しかし、MDSの場合は、このようなわけにはいかない。使用者が容易に利用できるとは、その意図する処理を複雑な作業をせずに、指定することであって、内容的には複雑な作業を実際の操作では容易にできるようになっているということである。

問題の発生の都度、意思決定者またはそのスタッフがプログラムを作成するというわけにはいかない。このようなことは、本来の仕事を軽視することになる。あるいは、システムの専門家に依頼すると、実際問題として、間に合わないことが多い。だから、非定型的なあらかじめ予定できない処理であっても、使用者がすぐさま、容易に利用できるように、あらかじめプログラムとして作成されていなくてはならない。

(4) 処理内容のよくわかるシステム

経営計画において参考となる情報は、何らかの処理結果としてのデータであ

るから、いわゆる客観データは少なく、分析結果としての予測値、推定値、各種統計量等が多い。例えば、需要予測、販売見積りはそうであるし、原価計算も主として標準原価計算が用いられ、このようなデータを前提とする財務諸表は全て見積りの数値である。そのため、使用者は、出力された情報について、その意味、条件、処理手順等を厳密に認識していなくてはならない。情報の意味を知らずに、それを利用することはできないことはもちろんであるが、計画過程で参考にする資料としてのデータは、特に、処理手順を吟味しないで用いることは意味がない。

企業の計画策定の中で経営的意思決定といわれるものであっても、経営科学、統計学、管理会計等の理論は万能ではない。我々はなるべく一般に妥当する経営理論を発展させることを期待してよいし、各企業において、実際に応用し、なるべく合理的、科学的分析を踏まえて、計画を策定することが望ましい。しかし、現時点では理論や分析手法は現実適用性という点で限界があるとみなくてはならない。

経営についての理論や分析手法を現実適用させていく上で、一つの壁は、計画策定者またはそのスタッフが十分その使用した理論や分析手法を理解せずに、評価を下すことがあることである。例えば、回帰分析をすることと予測をすることとは別のことである。統計学上の手法は、関連するさまざまな情報を計測し生成する。これらの情報を相互に関連づけながら、分析してみるとによって、対象としている問題について、役にたつことがわかることもある。あるいは、シミュレーションについても、そのモデルを構成する関係式を厳密に理解しないで、単に入力データやパラメータの値と評価対象になる変数の値とを結びつけて考えるのは危険である。

分析手法や分析上設定するモデルは、具体的に検討することによって、はじめてその効果、限界、生成される情報を使用する場合の注意事項等が明らかになる。そして新しい手法、モデルを開発するための基礎になるといえる。

以上のようなわけで、コンピュータによる情報処理は、経営計画の場合であ

るが、その処理内容、および出力結果である情報については、使用者が十分知らなくては意味がない。ところが、これは意外にめんどろな事なのである。このためには、プログラムの中のロジックが非常にわかりやすくなっていなければならない。そして、このための工夫を必要とする。具体的には、使用者自身によって記述されたもの以外は、厳密な定義がされていて、プログラムの内容に立ち入らなくとも、理解ができるものであるべきである。

(5) トータルシステム

MDSは経営計画策定に関する情報処理を全てカバーするトータル・システムであることが望ましい。経営計画の過程で参照される情報は互いに関連していることが多く、そのため、別々のシステムにしておくと、どうしても不便なことがおこる。

さらに、全体的な計画が設定されたならば、それを容易に実行させ得るようなシステムでなくてはならない。このようなわけで、経営計画全体を一つの適用対象とした、トータル・システムが望ましいことになる。

(6) 容易に維持、修正可能なシステム

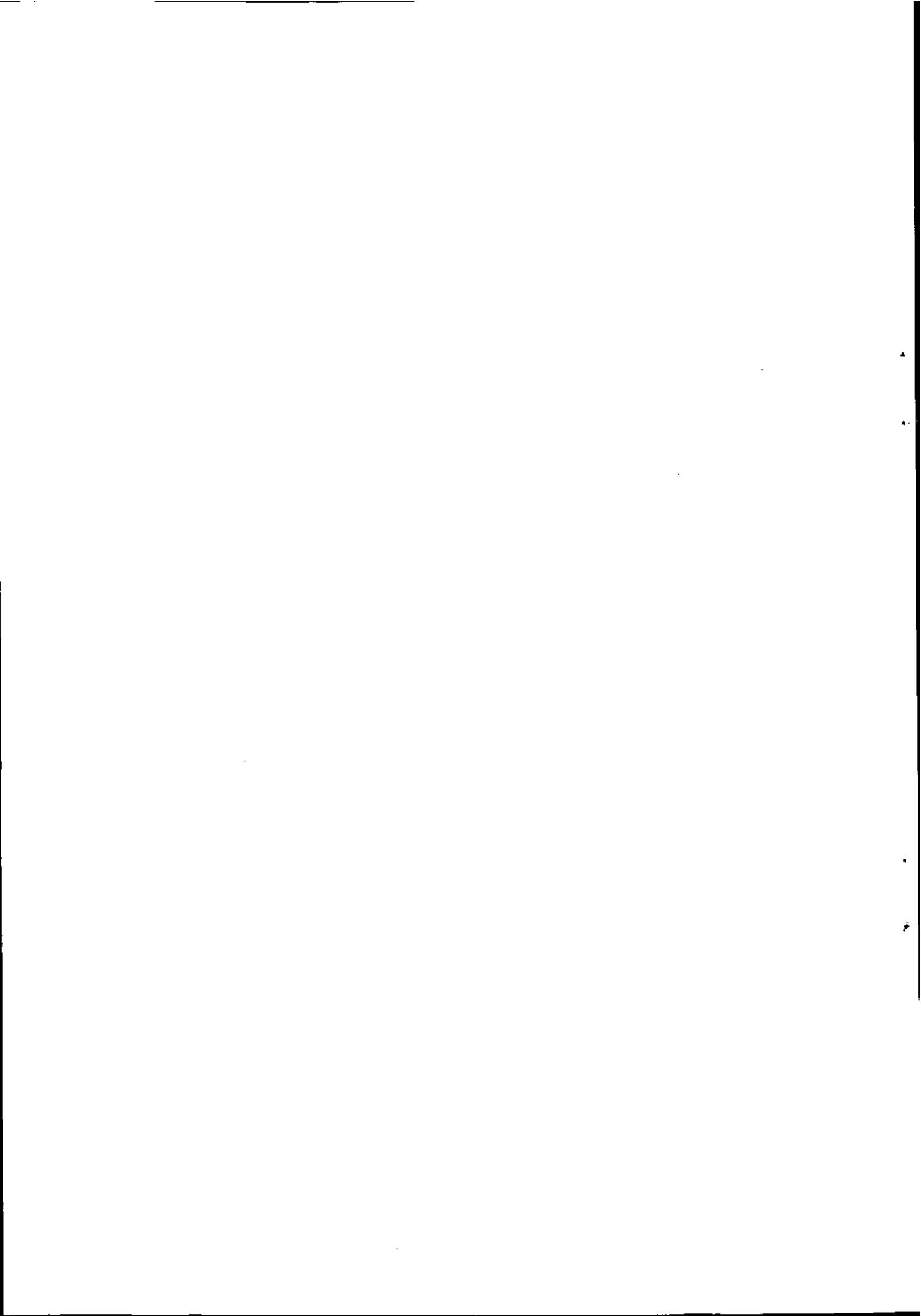
コンピュータを利用する人という意味での使用者にとっては、単にシステムを使用するというだけでなく、そのシステムの修正、維持、または新しいシステムの設計、作成あるいはその注文等々関連する全ての事柄に関心をもつ。

システムの設計、作成および維持、修正は多大のコストを費やすものであり、特に大規模なシステムは、簡単に作成できるものではない。システムは多様な処理内容に対処でき、長期間の使用に耐え得るものであるべきだが、そのためには、システムそれ自体の処理内容がそうになっていることはもちろんだが、加えて、容易に修正できるようにしておくべきであろう。いわば二重の備えが必要なのである。

容易に維持、修正可能であるということの他に、システムの作成それ自体も考慮しておかなくてはならない。

システム作成のコストは、誰が作るか、ということによって決まる。あるいは、誰でなくては行けないか、というべきかもしれない。つまり、コンピュータ・ソフトウェアは、容易に複製可能であるから、一般に適用できるものであるほどコストはかからない。個別企業に固有のシステムであると、実際に誰が作成したとしても、各企業別に開発することになる。この場合は、個別企業に個有の知識を知っていなくてはならないという、別の意味のコストも必要とする。一般に適用できるシステムであれば、それだけコストが減少する。

3. 経営計画情報システムの設計



3. 経営計画情報システムの設計

3.1 システムの設計基本方針

3.1.1 解 説

MDSは、前章で説明したいいくつかの一見互いに相反するような機能、性質をもつシステムでなくてはならない。この章では、そのような条件を満足するシステムの設計方法、およびそれに基づくシステムの基本構造を提案する。

3.1.2 情報処理機能による分割

MDSのような大規模なシステムを設計する場合には、全体を構成するサブシステムの設定の仕方が第一に重要である。各サブシステムは、問題オリエントに構成しないで、徹底して情報処理機能による。コンピュータ・システムにとって、情報処理機能とは、例えば、需要予測、生産計画、財務分析というような、個別の問題にかかわらず、共通にもっているいくつかの情報処理上の内容の区別である。それは、データの分析、処理、計算（コア上の処理）とデータ管理、ファイル管理に区別できよう。データ管理、ファイル管理と呼ばれるものは、処理対象になるデータそのものの記録、修正と使用者の行う入力を対象にしているもの、出力に関するもの、記録媒体それ自体の管理というように分割できよう。ほとんど全ての処理では、このような一連の処理を結合させて、全体を構成している。個別問題ごとに、このような情報処理機能を作るのではなく、全ての問題に共通の機能に注目して、システムを分割する。このようにして、システムを構成すると、各個別問題とは、このような機能を結合して作られるものにすぎないので、システム設計者は本来知る必要のない、各個別問題固有の知識を何ら知らなくてよい。それは使用者自らが指定することになる。

3.1.3 分析手法、言語とモデルの区別

データの分析、処理、計算（コア上の処理）に限定して、処理内容を検討してみると、具体的問題に対応した処理手順と問題とは独立な分析手法や言語としてのプログラムとがあることがわかる。この区別を認識することは重要である。なぜなら、具体的問題をプログラムとして表現するために必要な知識は本来使用者が知っているものであって、分析手法や言語は使用者が指示しなくても他の人によって作成できるものであるからである。従来、各個別問題の処理としてのプログラムの中に、分析手法を部分的に取り入れて全体を構成するものと、分析手法のみをパッケージ・プログラムとして作成するものがあった。パッケージ・プログラムの場合、いくつかの手法を連結させて用いたり、その間のデータを処理したり、報告書の形に形式を整えて出力したりする機能が十分備わっていないため、使用者が不便を感じる人が多いようである。問題に対応した処理手順を表現したプログラムは、その処理内容を厳密にシステム作成者に知らせるか、あるいは、かなりの部分をブラック・ボックスとして使用するということになる。この意味でそのシステムを使用する人間が積極的にシステム設計に参加する必要が生じることになる。また、そのプログラムの処理内容を多様にするためには、使用者の行う入力作業に余地を多く残す以外にない。結局、問題オリエンテッドに処理内容を規定すると、処理内容を固定的なものにするか、使用者の行う入力作業に余地を十分残すか、頻りにシステムを修正もしくは作り直すか、現状の処理に満足するかを選択をする以外にない。それで、従来のアプリケーション・プログラムの作成方法を反省し、次の点からシステムのもつべきデータの分析、処理機能を規定することを考えてみよう。第1に、使用者の知るべきことと、他の人間がプログラムを作成してもかまわない部分とに分離する。第2に、使用者の知るべきことは、使用者自身が処理手順を指定する。そして、使用者自身の行う処理内容の指定は、システムには組み込まない。システムに対する入力作業とする。第3に、使用者の行う入力作業は、コンピュータについての知識を必要とせず、簡単なものに

する。そのため、可能な限り、システムの中に、内容上本質的な部分は組み込むようにする。

これらの点から考えると、データの分析、処理についても、問題別の分割ではなく、その中で使われる、分析手法や言語によって分割して、サブ・プログラムを構成することが望ましい。分析手法や言語ではなく、問題に固有の処理手順としてのプログラムを、後の説明のために「モデル」ということにする。

「モデル」はいくつかの言語や分析手法を使って表現される。各問題に固有の処理手順のことである。

「モデル」は、使用者自身が記述することにし、システムは、問題とは独立なプログラムを組み込んでおく。このように、システムのもつべき処理機能を限定すると、そのシステムの作成者が、使用者と相談しなくてはならないことは、基本的になくなる。使用者は、そのシステムの処理内容をプログラムの中味に立ち入ることをせずに、例えば、適当な文献を参照すればそれで事足りる。

このような分析手法、言語としてのプログラムは複数個あるが、互いに独立であって、容易に追加、削除できる。それで、システム設計の時に、どのような処理を組み込まなくてはならないかということは考慮せずに、計画の過程で行う情報処理、問題分析の方法、スタッフの学習状態等によって、漸次追加すればよい。

3.1.4 会話的入出力

システムを情報処理機能によって分割し、システムはいくつかの各問題とは独立な処理ツールを、使用者に対して提供するようにすると、使用者は、各問題について、その処理手順をシステムに対して入力しなくてはならない。

そして、使用者は、コンピュータについての専門知識を知っていなくとも、あるいは、プログラミングという作業をしなくてもいいように、システムの側で、使用者の行う情報入力の論理構造を設定し、それを質問-応答の列として表現する。使用者は、システムからの質問に逐次的に答えることによって、結

果的に、入力作業をおこなうようにする。

こうして、使用者は、本来の分析、意思決定に専念できるようにする。

3.1.5 「モデル」の管理

(1) 「モデル」の機能

「モデル」は使用者によって記述された情報（処理手順を示す）である。システムは「モデル」を管理するという機能をもつ。「モデル」の定義（記録）、削除、修正である。使用者は、一度定義した「モデル」をその名称を示すことにより実行させることができる。あるいは、意図している処理内容と類似している「モデル」がすでに定義されていたならば、修正した後、実行させるとよい。「モデル」の詳しい説明は章をあらためて行う。

(2) 「モデル」管理の意義

「モデル」を管理するという機能は、使用者が煩雑な入力作業をしなくてもすむためのものである。いわゆるライブラリー・ファンクション（library function）、マクロ命令と異なるのは、具体的な処理内容を、システムの側であらかじめ決めるのではなく、使用者自身が記述するということである。従来、アプリケーション・プログラムとして、一括されていたものを、二つの階層に分離する。上層の「アプリケーション」は、使用者自身が記述する。しかし、この部分はプログラミングを必要とせず、システムからの質問に答えることによって、結果的に処理内容を記述していることになる。使用者にとって必要な知識は、その分析対象である問題の内容、および使用する分析手法や関数、シミュレーションの意味内容、データの意味や記録方法等であって、コンピュータに固有の知識ではない。その作業はさほど煩雑ではない。しかし、各使用者は類似した処理を何度か実行させたいことがよくあるはずである。さらに、全社的またはある一つの部門において、基本的な分析体系を設定しておいて、まず、その結果である情報を検討することから分析をはじめ、問題点を発見し、以後の分析を進めていくといった様式をとることが多いはずである。こ

のようなわけで、使用者は類似する入力作業を何度もおこなうことになっては、システムの使用がめんどうになる。個別問題に対応して、プログラミングされているシステムは、このような不便はない。

それで、使用者自身が指定した処理内容をシステムに記録しておき、以後類似した処理は、その記録した情報に基づいて、システムが処理を実行する。こうして、「モデル」が完成すると、個別問題に対応した、システムと同様になる。ただし、その「モデル」を削除する、あるいは一部修正するといった作業はシステム自体の変更にはならない。あくまで使用者側の入力作業としておこなわれ、プログラミングの必要はない。もちろん、「モデル」として定義しなくても、直接処理手順を指定して実行させることもできる。要点は、システムの処理内容を規定するのは、あらかじめ作られているプログラムだけではなく、使用者自身によって指定された部分も、事実上、プログラムと同様の働きをするようにしておくことである。

3.2 システム構造

3.2.1 システム構造の概要

我々の提案する設計方針によるシステム構造は、各サブシステム、および各サブシステムを構成する各サブプログラムが互いに独立であって、他のサブシステムやサブプログラムの変更、追加、削除によってほとんど影響を受けないのが特徴である。したがって、基本的構造は別にして、機能の多様性に依存するサブプログラムは、あるいはこのとおりでなくとも、削除してもかまわない。それは本質的ではなく、必要に応じて、サブプログラムを容易に追加、修正、削除できることが重要なのである。

3.2.2 各サブシステムとその機能

システムは次のサブシステムから構成される。サブシステムは各々個有の情報処理機能に対応しているので、その機能を列挙することによって、サブシ

テムを示す。

1. データの管理
2. カードからのデータの入力，テープとのデータの入出力
3. レポート作成，情報検索
4. システムの管理
5. 「モデル」の管理
6. 使用者とシステムの会話的入出力
7. システム全体の制御

データの入出力，レポート作成，情報検索は，通常データ管理に属する機能として扱われるが，MDSの場合，データの表現形式に関する機能とデータ値を設定する機能とは分離した方がよい。データ値の設定は，データの入力，演算，関数の結果，分析手法を表現したプログラムの出力として等多様だからである。

データの管理は，システムで直接取り扱う記録媒体に，データを記録するための組織的な骨格を規定したり，それを修正したり，削除するといった，表現形式に関する処理である。

カードからのデータの入力は，大量のデータを，あらかじめ定義された表現形式に基いて，データを入力し，個別データの値を設定することである。テープとのデータの入出力は，システムで直接取り扱う記録媒体上に記録してあるデータをテープに記録したり，逆にテープ上に記録してあるデータを移し替えるものである。

レポート作成，情報検索は，記録媒体上に記録してあるデータを，使用者の指定する内容や形式にしたがって，端末装置およびプリンターに出力する機能である。

システムの管理は，データの機密保護や保全性を保証するために，各データの参照，更新，修正，削除を，各使用者によって制限をつけるための機能と，記録媒体の定義や削除についての機能とがある。この機能は，ある特定の人間

によってのみ利用できることになる。

「モデル」の管理は、使用者がシステムの提供する処理ツールを使って表現した処理手順を記録したり、それを修正または削除する機能である。

使用者とシステムの会話的入出力は、端末装置への表示、使用者の指示した情報の解析、記録を担当する機能である。

3.2.3 メイン・ルーチン

システム全体の制御を担当するサブシステムは、プログラミング上ではいわゆるメイン・ルーチンである。この役割を説明するために、システムの実行の流れを概説する。このシステムは、使用者の要求に応じて、さまざまな処理をする。ある1回の実行では、実行されない部分もある。処理の内容は、基本的には、システムの提供するどの処理ツールを実行するか、それについての必要な情報を使用者が入力し、システムが解析、記録する部分と文字通りの意味での実行の部分とに分かれる。各処理ツールに対する入力情報は、処理ツールごとに異なるので、その解析、記録は処理ツールに対応するサブプログラムの中に組み込む。それで、各サブプログラムは、文字通りの実行と入力情報の解析、記録を担当するものが対になっている。個別処理ツールの追加、削除を容易にするためにも、処理ツール別に対しておくのがよいだろう。

全体の実行の流れは、処理ツールを指定するまでを、メイン・ルーチンがおこない、その後は、各サブプログラムに実行が移る。この過程で、メイン、サブプログラムが共に端末装置に関するサブプログラムを使用する。

全体の実行の流れは、使用者がダイナミックに替えるものと想定する。例えば、データを直接出力させてみることもあるだろうし、ある分析手法を要求することもある。このように、使用者の指示にしたがって、実行の流れを変更する。

3.3 データの管理

3.3.1 データ・ベース管理

(1) データ・ベース

データ・ベース (data base) という用語は1950年代後半から使われているものであるが、60年代後半から現在までに盛んに使われるようになった。ベースとは本来基地または根拠地という意味であるが、用語の由来とはわかりなく、要するに、組織的に記録された大量のデータという意味であると理解してよいだろう。ただし、単なる量的問題ではなく、個別のプログラムや問題領域に各々固有のデータをもつのではなく、それらとは独立に、データそれ自体として管理するという考えがある。さらに積極的に、コンピュータだけに限らず、ある統一的なデータの記録、管理を指している場合もある。

(2) 「データ・ベース」管理の必要性

データベースおよびその管理を必要とする理由、あるいはデータベースの目的は基本的に次のようにいえる。

(a) 個別プログラムに個有のデータを対応させておくと、あるプログラムによる出力データは他のプログラムの入力データであるといった関係があるので、ある一つのプログラムの変更が他のプログラムの変更を必要とする。少なくとも、必要とするかしないかを意識していなくてはならない。それで、データの取り扱いを各プログラムから独立させることにより、プログラムの中から、データについての操作を解放する。

(b) 個別プログラムに対応したデータを、各々もっていると、通常、組織内の部門等が必要とする各データを、それぞれ設定することになる。ところが、いくつかの部門、あるいはいくつかのプログラムが同じデータを参照または更新 (update) することがある。それで、例えば、あるデータのある部門なり人が修正する責任をもっていると、実際には、たくさんのデータを修正しなくてはならないことになる。したがって、修正したかどうかを参照する

方は常に意識していなくてはならない。

それで、データを重複して記録するようなことのないように、統一したデータの記録をおこなうようにする。

(c) 各プログラムの中で、データを取り扱う部分（端的に言えばファイル管理）は、個別プログラムにかかわりなく、比較的共通な操作手順がある。しかもプログラムの中に占める割合（量的にも、プログラムのむつかしさという点でも）が意外に大きい。個別プログラムの作成の際にこの部分をしないで済むようにする。

(d) コンピュータに対して、使用者が要求する処理手順の流れは、例えば、データの入力、処理、出力というように決まっているわけではない。すでに記録されているデータをそれとして参照したいこともよくある。コンピュータを記録装置として利用することもあり得るわけである。

このことは特に、大量のデータの中から意図するデータを取り出す時に、コンピュータを利用する効果があらわれる。大量のデータの中から、意図するデータを選び出すという作業もかなりの時間、その他のコストを必要とする。コンピュータにその作業を代替させる。

(e) 分析すべき問題が発生するたびに、それに必要なデータを収集、記録するのではなく、積極的に、データを常に収集、記録しておき、必要な時にはすぐに参照できるようにする。

以上のように、互いに関連するがしかし微妙に異なる目的がある。それで、このデータ・ベースというものについて、詳しく考えてみよう。データ・ベースというときに、データの表現形式等、管理の方法を組織的におこなうことと、データそのものを組織的に管理することを区別して考えるべきである。データそのものを組織的に管理する場合は、全社的にデータを管理しておくべきかどうかということを考えなくてはならない。そのためには、全社的に記録したデータが、各部門あるいは人間によって、それぞれがみな参照されるかまたはそのような可能性があるのかどうかを明らかにする必要がある。これは各企業に

よって異なるだろう。しかし、概していえることは、計画過程では、業務上の詳細な内容を表わすデータを参照することはあまりないであろうということである。企業内では、各レベルや部門ごとに参照するデータはある程度限られていると考える方が実際的なようである。全ての部門あるいはいくつかの部門によって共通に利用されるデータもある。このようなデータについては、その値を正しく設定しておくために責任を明らかにしておかなくてはならない。

問題の発生する以前にデータを収集、記録しておくことは望ましい。しかし企業活動の現実からみて、そう容易なことではないはずである。問題の発生する以前に、必要とするだろうデータを記録するとなると、その量が非常に問題になるし、発生する問題によっては、使用するデータを予測するのは困難である。したがって、実際には、ある程度基本的なデータを常時記録・更新し、その他のデータについては、必要に応じて収集、記録するという方法をとる以外にないであろう。むしろ、全データの整備のために時間を費し、そのために、必要な分析を待つようなことは好ましくない。経営計画の場合には、必要とするデータの内容も変化するであろうし、問題の内容によって、収集すべきデータが明らかになることもある。最初からデータの内容に立ち入って整備するよりは、可能なものから準備し、漸次発展させていく方が適切である。

このように、データそのものの組織的管理を全面的に実現するのは一見妥当な方法であるが、実際にはいろいろな問題が多い。

データの表現形式を整えて、管理の方法を確立しておくのは、さほど困難なことではない。この場合でも、企業内の各レベル、部門で取り扱うデータがある程度限られていて、各々に適する表現形式があれば、別々の管理方式を採用してもかまわないし、その方がよいだろう。MDSの対象とする経営計画で取り扱うデータは、表現形式の相同性という点からみて、同じ管理方式を採用してかまわないし、互に関連する部門が多くあるから、同一システムの記録媒体に記録しておくことが望ましい。この場合でも、参照する部門、人間、データ値の更新等をする部門、人間を明らかにし、共通に利用しないデータは、別

々の媒体へ記録しておくことも一つの方法である。その時々分析結果としてのデータの中にはこのようなものも多くある。MDSの場合に、データを組織的に管理するのは、表現形式を統一したり、容易に記録できるようにしておくことが目的で、データの内容については、あまり厳密な規定を設けることは考えるべきではない。むしろ、漸次発展するものであって、問題の必要を基準にして考えることである。

3.3.2 データ・ベース管理システム

(1) MDSのデータ・ベース管理システムに対する基本方針

データ・ベース管理システムと総称できるシステムは、現在、かなり商品として開発されている。データ操作システム、汎用情報検索システム、情報管理システム、ファイル管理システム等と呼ばれているものは全てデータ・ベース管理を目的とするシステムである。あるいはプログラム言語の一つである、COBOLもデータ管理機能が中心である。現在市場に出ているシステムの特徴、機能等については解説文献がある。^{*}これらのデータベース管理システムについて、MDSとの関係を考察してみよう。前節での説明のとおり、MDSにおいても大量のデータを組織的に管理する必要があるので、その管理方法を考える上で、既存のデータベース管理システムの技術をどのように導入すべきか注意深く検討してみることは当然である。MDSの中に含まれるシステムの中には、このようなデータ・ベース管理システムを一部としてそのまま組み込んでいるものもある。

(2) データの特徴及び表現形式

データ・ベースについて考察する場合、対象とするデータの特徴、表現形式を検討してみなくてはならない。つまり、全ゆるデータを対象とする場合には、それぞれの分野別に、特徴を吟味しておかないと、必ずしも共通な表現形式を

^{*} 本書の終りの参考文献C-1(1),(2)等

採用してよいかどうか明らかではない。それで、一見汎用的な管理方法であっても、実はそれに適さないデータを利用する場合もある。このことは特に、比較的データの管理が発展している分野、例えば、文献検索や事務管理上のデータの管理等における管理方法を他の分野においてもそのまま利用する場合におこる。MDSの場合、まだ確立したデータ・ベース管理の方法についての研究が発展していないので、単純に既存の方式を適用してしまふことになりかねない。

(3) MDSに適するデータ構造

それでは経営意思決定に役立つデータとはどのようなものであろうか。厳密な区分はできないであろうが、企業内で取り扱うデータには、従業員名簿、得意先名簿、仕入先名簿等々の事務管理上のデータと、経済統計データ、経営統計データとがある。この取り扱い方は異なる。経営意思決定において参照の対象となるのは後者の方が多い。さらにデータには、いわゆる原データ、客観データと処理・加工済みデータとがある。経営意思決定において参照するのは、後者が多い。要するに、コンピュータを利用するのは単なる記録装置としてではなく計算、処理のためである。

データ・ベース管理システムの文献によると、この点の議論がなく、どちらをも対象としているようにも受けとれる。しかし、よく検討してみると、データベース管理システムは、結局本格的な計算、処理を対象とするデータは想定していないと思われる合理的根拠がある。それは、1.検索の方法が単一データを取り出すことに主として力点をおいている、2.データ構造がCOBOL型階層構造になっていて、事務管理上のデータの表現に適しており、計算中心のプログラム言語によって記述されるような処理には適していない、等のことである。計算を中心とするプログラムでは、データを計算上使用する場合、単一のデータを個別に使うのではなく、かなり大量のデータを組にして用いる。そして、そのときには、ベクトル、行列等要するに配列 (array) として表現する。したがって、記録媒体上のデータの表現形式をそのまま使って、組としてのデータを参照できなくては、非常にめんどろなことがおこる。データ・ベース管

理システムは、記録媒体上での表現形式が、プログラムのためのコア上での表現形式と独立であることを意図している。つまり、データのプログラムからの独立をもっとも基本的な考えにしているのである。しかし、計算を中心とするプログラムでは、大量のデータをコア上へ読み込む場合に、記録媒体上での表現形式と異なると、それについてのデータの配列替えに特別の作業を必要とする。

3.4 データ管理の理論

3.4.1 データ構造

(1) データ管理の概略

データ・ベース・システムの開発と共に発展してきたデータ管理についての理論をサーベイしてみよう。ただし、データ管理についての理論の詳しい説明はすでにいくつかの文献あるいはたくさんの論文があるので、それに譲ることにし、ここでは簡単に概略を説明する。ここでの説明の目的は、我々の提案するシステムのためのデータ管理の方法を説明するために、データ管理においては、どのようなことが問題となるかを明らかにすることである。

データ管理は、基本的に二つの問題に分けて考える。一つは、大量のデータをどのような関係によって、組織化するか、つまり、記録するデータ相互間の論理的关系である。これをデータ構造 (data structure) と呼ぶ。そして、データ構造を前提にして、データを記録媒体上に記録する表現形式を考えるのがもう一つの問題である。これを一般には記憶構造または記録構造 (storage structure) と呼ぶ。ここでは記録構造ということにする。

(2) データの概念

データ構造を考えるために、データまたは情報とは何か、ということから出発する。そして、情報はある適用実体 (entity) に関する属性の値と規定する。適用実体は、何らかの基準によって集合を作り、その集合に含まれる適用実体のもつ属性は基本的に同じものである。この意味で、適用実体の集合に対

して、各適用実体についての属性の値の集合が対応する。これをファイル (file) と呼び、データ・ベースはファイルの集合として組織される。

ある一つの適用実体に関する属性は通常複数個ある。この属性相互間の関係を規定することが次の問題である。属性相互間の関係の規定の仕方は COBOL と同様の階層構造であることが多い。すなわち、属性を順々に分割していき、最小単位の属性を導びき出すというものである。

(例)

従業員(個々人)という適用実体を考えると、その属性は氏名、番号、年齢、給与となるとする。氏名は氏と名、給与は職給と金額というように分割する。

属性の最小単位を項目(item)と呼び、これに対して属性値、すなわち、データが対応する。

属性相互間の関係の規定の仕方は、階層構造が多いのであるが、これ以外に線構造、網構造もあり得る。網構造は階層構造の拡張で、ある属性が複数個の属性に含まれる場合である。線構造は、最小単位の属性のみを並べたものである。

一つのファイルに属する適用実体(またはそれを表現している金属性を一つの単位としてもよく、これをエントリ(entry)と呼ぶ)相互間の関係、あるいはファイル相互間の関係も、一つの適用実体に関する属性相互間の関係と同様に規定する。

このように、全データを適用実体を軸にして構成するのが基本的な考え方である

(3) データ・ベースの関係式

各適用実体についての属性を並列的に並べた表をデータの表現形式として使用する方法もある。これは、E. F. コッドによって提案されたものである。これを、データ・ベースの関係形式 (relational model of data base) という。データ・ベースの関係形式とは、次のような形式の表である。

R :

A_1	A_2	A_n
a_{11}	a_{12}		a_{1n}
a_{21}	a_{22}		a_{2n}
\vdots	\vdots		\vdots
a_{m1}	a_{m2}	a_{mn}

図 3 - 1

支 店 :

名 前	No.
東 京	1
大 阪	2
京 都	3
名 古 屋	4
札 幌	5

図 3 - 2

A_1, A_2, \dots, A_n は ITEM と呼ぶ。属性の名称である。各列ごとに ITEM に対応するデータを書き並べる。ただし、一つの関係形式に、同一のデータをもつ行が複数個存在することはない、各データは、それ以上に分割できない原子的情報単位である、といった条件がある。結局、関係形式による表は、適用実体に関する属性を並列的に並べたものであるといつてよい。このため、階層構造よりは単純化しており、表の間の相互関係がかなり密接にあるこ

とになる。つまり、異なる表の中に同一のITEMをもち、データが重複していることがある。このような共通したデータによって、2枚の表を関係づける。

仕入先

製品名	No	社名
A 1	1	S 1
A 2	2	S 2
A 3	3	S 3

図 3 - 3

倉庫

名称	製品名	在庫量
S 0 1	A 4	1 0 0
S 0 2	A 1	2 0 0

図 3 - 4

例えば、例示したこの2枚の表において、共通する製品があるので、社名S1の会社から仕入れた製品の在庫してある倉庫名がわかることになる。

このデータ・ベースの関係形式によるデータの表現については、検討すべき点が多いのであるが、我々はMDSのためのデータ管理について考えているので、これとの関係にのみ限定して検討した。そして、一部分についてはこの方法を採用することにしたのであるが、これについては節をあらためて説明する。

3.4.2 記録構造

記録構造とは、データ構造を前提にして、具体的に記録媒体上に表現する場合の形式である。データ構造は利用者の側からみた論理的形式であるので、その内容は区別しなくてはならない。

データの記録構造を考えるときには、記録の対象としているデータ群について、最小の処理量でそのデータを探索できること、データを部分的に追加したり、削除したりする場合に、そのため、必要な処理を最小にすることが問題になる。

(1) データ群の編成方法

データを編成する方法は、大別して順編成 (sequential organization) , 直接編成 (direct organization) , リスト編成 (list-organization) がある。

順編成は、対象とするデータ群をまとめて一定の順序にしたがって記録する方法である。この方法は領域が最小限ですみ、記録されている順に検索する場合 (そのデータ群を全てまとめて検索する場合) には効率がよい。反面、データを追加、削除する場合は、そのデータ群全てを書き替えなくてはならない。

直接編成は、各データを内容上のキー項目の値を一定の計算によってアドレスを導き、それに対応するところに記録する方法である。この場合、異なるデータであっても、計算結果としてのアドレスが同じ値になるものがある。これを同義語 (synonym) という。これについては、同義語だけを他の方法によって記録しなくてはならない。

この方法は、データをランダムに検索する場合に適している。そして、データの追加、削除についても、そのデータだけを対象にすればよい。

リスト編成は、個々のデータ中に次のデータのアドレスを合せて記録しておく方法である。ただし、必ずしも次のデータではなく、前のデータでも、両方でもよい。さらに最後のデータに最小のデータのアドレスを記録して輪状にしておくこともある。

この方法は、データの記録領域を多く必要とする。反面、データの追加、削除については、そのデータとそれに関連するデータだけを対象にして処理すればよい。

以後の説明のために要約しておく、次のようになる。

(a) 組にして検索するデータ群は順編成がよい。ただし、この場合は、データの追加、削除について特別の工夫を必要とする。

(b) 組にして検索するデータをまとめたものをいくつか考えると、その組全体としてのデータのアドレス (最初のアドレス) は、直接編成かリスト編成

がよい。この場合、同義語が生じないようにしておくことが望ましい。

3.5 MDSのためのデータ管理の方法

3.5.1 MDSで扱うデータ

(1) 条件

MDSで取り扱うデータは主として経済統計データ、経営統計データ等で、計算の対象となり、計算、処理結果としてのデータである。これらのデータは、収集、表示過程で独特の表現形式を使って表わされる。

統計データを中心とする、経営計画の中で使用する各種データは、分析者の立場からみて、分析上の単位となる概念、項目を中心に認識される。例えば、国民所得、国民消費支出、売上高、利益、等々である。このような分析上の単位となる項目は、分析上の必要から生じる明示的な指標によって分割される。

分析上の必要から収集するデータであると、必ずそのデータの意味を明示しなくてはならない。指標となるデータは、例えば、年度、四半期、月次等の時間に関するもの、あるいは、県、支店、品種等がある。指標を構成する要素であるデータ、例えば、支店については、東京、大阪、名古屋、京都等々であるが各適用実体に対応している。一つの分析上の取り扱いの単位となる項目は、一般には、複数個の指標によって分割される。例えば、ある製品の年度別、支店別、品種別の売上高という言い方がこれである。そして、これら指標どうしの関係は、互いに独立であることがほとんどである。つまり、ある品種について、存在する(データとして記録する)年度が、他の品種については存在しないといったことはむしろ稀である。この場合には、階層構造は適さない。

指標となるデータは、分析上の項目をいくつか共通にもっている場合もある。つまり、売上高、利益という項目は、ともに、年度、支店によって分割されたデータから構成されていて、その年度、支店を構成する要素であるデータは、売上高、利益ともに同じであるという場合が多くある。これは、ある意味では

当然のことであって、共通の指標によって分割されたデータを収集、整理することは、データ解析の基本的態度である。

コンピュータの記録媒体にデータを記録する場合でも、そのデータを記録、使用する人間は、そのデータの表現形式を適当なイメージをもって、想定できなくては取り扱いが不便である。一般にデータを収集、表示する場合の、表の作成方法、表現形式をそのまま、頭に描いておくことができるなら取り扱いが便利であろう。そして、いわゆる経済分析、経営分析上のデータの表現形式は、つまり表の作り方であるが、きわめて共通した方式があるのである。このことを示すために、異なる分野からランダムに引用した表を例示してみよう。

図3-5はある企業の売上高を示す表、図3-6は原価計算で表示されるものである。

支店 品 種 年月	札幌		仙台			東京	
	TOTAL	DAI	CHU	SHO	TOTAL	DAI	CHU
N46-2							
N46-3							
N46-4							
N46-5							
N46-6							
N46-7							
N46-8							
N46-9							
N46-10							
N46-11							
N46-12							
N47-1							

図3-5 売上高

費目	年 月	1 月	2 月	3 月	4 月	
	販 売 量					
期 末 在 庫 量						
合 計						
期 首 在 庫 量						
製 造 量						
見 積 製 造 原 価						
間 接 費 配 賦 過 不 足						
期 末 製 品 棚 卸 高						

図 3 - 6 製 造 予 算

(2) 記録媒体上への表現

コンピュータの記録媒体に記録したデータは、それ自体として、直接、端末装置やプリンターに出力する場合と、プログラムの処理のためのコア(core)上へ読み込む場合がある。この両方の場合について、記録媒体上のデータの表現形式と利用のための表現形式が同じであることが望ましいはずである。そうでなければ、データの配列を替えるという作業が必要になり、当然、一定の情報がシステムに対して、使用者から指示されなければならず、またシステムとし

ても、その指示に基づく処理をしなくてはならない。

使用者が、直接にデータを参照したいために、端末装置やプリンターにデータを出力する場合を考えてみよう。計画過程における分析、意思決定の参考となる資料は、多くの場合に、単一のデータであることはない。関連するデータを組にして、参照することになる。この意味で、単純な情報検索とは異なる。

その場合、データをまとめる基準は、分析上の項目であって、その項目別の指標つきの表が基本になる。だからこそ、通常、コンピュータとは無関係の分析においても、このような形式の表にまとめるのである。単一のデータを取り出す場合においても、分析上の項目、指標を軸にして考えるのが自然である。以上のようなわけで、分析上の項目、指標によってデータを組織的に記録するという方法は、使用者にとって取り扱いが易しいことになるわけである。もともと、使用者にとっては、処理結果としてのデータと原データとの区別はなく、一般に利用するデータの表現形式をそのまま、コンピュータの記録媒体への記録形式として採用するのであるから、当然のことではある。

次に、プログラム上での必要のために、記録媒体からデータをコア上へ読み込む場合、あるいは、処理結果であるデータを書き出す場合を検討してみよう。

MDSの対象となるデータの分析、処理、加工は、大量のデータを同時にコア上へ読み込んで計算をする。このとき、コア上に表現されるデータは配列（一般には多次元配列である）によって表現されることがほとんどである。計算を中心とするプログラムのためのコンパイラ言語であるFORTRANが、配列を（階層構造ではなく）採用しているのはこのためであろう。大量のデータの計算は、行列、ベクトルとして表現されるデータに対しておこなわれることが多いからであろう。そしてこの場合、その配列（array）を構成している個々のデータではなく、配列全体を1個の取り扱い上の単位とする方が便利である。

具体的には、配列に対して変数名を指定する。そして、このような配列は、分析上の概念、項目に対応していることもまた当然のことである。

以上のようなMDSの対象とするデータの表現形式を前提にしたデータ構造

を記録のために採用することにする。

3.5.2 データ構造, その他

(1) 前節でみたような, 経営計画で使用するデータの表現形式に基づいて, データ構造を規定することにする, この基本は指標付きの表によって, データをまとめることで, 指標を示す個々の要素も, 属性の名称ではなく, データそのものとするところである。

指標付きの表において, 指標以外の部分は, 指標の個数に対応する次元をもつ多次元配列である。このため, 各指標を構成するデータのサイズ(個数)によって各次元のサイズが規定される。各指標を構成するデータは, 適用実体に対応している。しかし, 適用実体についてのデータである指標は, 分析者の必要上決められたものである。指標を構成するデータは, 例えば, 支店についてのデータである東京, 大阪, 名古屋等々と, その支店の n である 1, 2, 3 等々のように, 複数の属性を表現していることもある。この場合, 東京もその n である 1 とは, 同一適用実体についての値である。したがって, 使用者が指標を軸にデータの意味を考える場合, その属性の中のどれであってもかまわない。

以上のような大まかな考えを, 厳密に規則として定める。それを次に説明する。

(2) データを, 指標となるデータと直接に分析の対象となるデータに分離し, 前者については, 関係形式 (relational model) によって表現する。関係形式による「一枚の表」を RECORD ということにする。これが取り扱いの単位になる。指標となるデータによる RECORD を TYPE-I RECORD ということにする。

TYPE-1 RECORD をいくつか組にして, TYPE-I FAMILY を作る。FAMILY は RECORD からなる集合(族)である。FAMILY の要素は RECORD の他に, FAMILY であってもよい。ただし, FAMILY の要素の中に事実上同一の RECORD が含まれていてはならない。例えば

FAMILY F1 = { R1, R2 }

FAMILY F2 = { F1, R1 }

ただし、R1, R2は RECORD とする。FAMILY は順序のある集合とする。
つまり、

F1 = { R1, R2 }

F2 = { R2, R1 }

とすると、F1 と F2 とは別のものとする。

こうすることの意味は後で説明する。

直接に分析の対象となるデータは多次元配列 (multiple array) によって表現する。ただし、各次元に対応する TYPE-I RECORD を明示して記録する。もちろん、TYPE-I FAMILYでもよい。この一つの配列を TYPE-II RECORD ということにして、取り扱いの単位とする。

TYPE-II RECORD をいくつか組にして、FAMILY を定義する。(つまり、使用者が定義してもよい)。TYPE-II FAMILY は順序のある集合ではない。FAMILY の要素は、RECORD と FAMILY である。TYPE-II FAMILY は、事実上、要素である RECORD を重複してもかまわない。

TYPE-II FAMILY を定義するのは以下のような事情による。すなわち、データを分析上の項目である RECORD としてまとめた場合に、いくつかの RECORD 相互間の関係を明示しておくといろいろな場合に便利なのである。

MDS で取り扱うデータは、不特定多数の人によって、参照、更新されることを想定すべきではあるが、しかし、全てのデータがそれぞれ全ての人によって参照または更新されるわけではない。特にデータの更新、すなわちデータ値を設定する作業は責任ある部門または担当者を決めておくべきであろう。MDS で取り扱うデータは分析結果である処理済みのデータが多く、この処理は単純な計算だけではなく、分析手法、シミュレーション等、処理のため多くの費用を費やしたものもある。任意の人によって共通に利用するデータを考えると結局、あらゆるデータがそのデータを処理・加工すると生成できるという意味で

の原データを記録することになる。そして、各使用者は原データを処理、加工して必要とするデータを作り出すことになる。このようなやり方は、単純な処理によって必要とするデータを生成できる場合、あるいは文献検索のように、データをそのまま利用する場合には、適切な方法であるが、MDSのように分析結果である情報を主たる内容とし、またそのための処理に複雑な計算を必要とする場合には、適当ではない。したがって、MDSの場合には、記録されているデータの中には、主として利用する部門または担当者がいろいろ違うことも十分あり得るのである。

そうであれば、そのようなグルーピングをしておく、いろいろな効果がある。第1にあるデータを検索する場合に、対象とするRECORDの含まれているグループに限定してからはじめればよいことになり、第2に、必ずしも全データを同一の記録媒体に記録する必要がなくなる。

FAMILY、RECORDを要素とする集合（RECORD自体が個別データの集合であるから集合族と呼ぶべきかもしれない）として定義し、あるRECORDがいくつかのFAMILYに属することもよいとするのは、取り扱われるデータと人間または部門の対応関係に融通性をもたせるためである。例えば売上高という項目によってまとめられたRECORDが、販売、営業、生産の各部門で共に使用する場合に、主として、販売部門で取り扱うRECORDの集合としてのFAMILY、同じく営業部門で取り扱うRECORDの集合等、いくつかのFAMILYが共にこのRECORDを要素としてもつ。

FAMILYはその名称と要素であるRECORD、FAMILYの名称を記録するだけである。データ値そのものを重複して記録するわけではない。

(3) データ管理の理論で展開されているデータ構造と我々がここで説明したものととはどこが違うのだろうか。端的に言えば、データ管理の理論では、適用実体の集合 E から j 番目の属性のとり得る値の集合 V_j への対応 $\varphi_j : E \rightarrow V_j$ を考えたのであるが、我々は複数個の適用実体の集合 E_1, E_2, \dots, E_n の直積 $E_1 \times E_2 \times \dots \times E_n$ から j 番目の属性のとり得る値の集合 V_j への対応

$$\tau_j : E_1 \times E_2 \times \dots \times E_n \rightarrow V_j$$

を考えたのである。ただし、各適用実体の集合、例えば、 E_1 から i 番目の属性値のとり得る値の集合 U_{1i} への対応、

$$\Psi_{1i} : E_1 \rightarrow U_{1i} \quad i = 1, \dots, N_1$$

を、 E_1, \dots, E_n について考え、 $U_1 = (U_{11} \dots U_{1N_1})$

$U_2 = (U_{21}, \dots, U_{2N_2}) \dots$, $U_n = (U_{n1} \dots U_{nN_n})$ を指標と呼び、 U_1, \dots, U_n の直積から V_j への対応 $\pi = \Psi^{-1} \varphi_j : U_1 \times \dots \times U_n \rightarrow V_j$ を指標から分析上の項目である RECORD への対応としたのである。ただし、

$$\Psi^{-1} = (\Psi_1^{-1} \dots \Psi_n^{-1})$$

$$\Psi_i^{-1} = (\Psi_{i1}^{-1}, \dots, \Psi_{iN_i}^{-1})$$

そして、属性値のとり得る値の集合 V_1, V_2, \dots, V_m をそれぞれ RECORD とし、それらの関係は何も考えなかった。^{*} $U_i = (U_{i1}, \dots, U_{iN_i})$ における U_{i1}, \dots, U_{iN_i} 、すなわち、各適用実体に関する属性についても、単に最小単位を並列的に記述するだけとし、関係形式を採用したのである。

結局、データ管理に関する既存の理論は、我々の考えでは、指標を構成するようなデータを対象にしているのである。したがって、指標を構成するデータについては、その属性がたくさんあり、複雑であれば、関係形式でなくても、階層構造でもよい。大切なのは、適用実体の集合を複数個考えることによつて、定義可能な属性もあり、これが分析の対象となるデータなのだということ

* FAMILY を定義することはある。

である。

関係形式によって表現される「一枚の表」では、指標を構成するデータを表現しにくいこともある。例えば、売上高についての年次データと利益についての年次データがあつて、これを示す年度が多少異なっていることがある。この場合は、売上高と利益については、別の指標を必要とする。しかし、共通にもっている年度は、両方の項目を関連させることのできる基準である。このように、同一集合の適用実体としてまとめることのできるもので、部分的にわかれている場合については、ある関連を示す情報を記録しておくべきである。このために、TYPE - I RECORDのFAMILYを考える。同一FAMILYに属するRECORDを指標としてもっている場合には、事実上同一のRECORDを指標にもっているRECORDに準じて、操作できる。いくつかのRECORDが一つの指標を構成していてもよい。したがつて、指標として、FAMILYを指定することもよいようにするわけである。

(4) MDSは経営計画に参加する多数の部門、人間によって利用される。そこで、データの機密保護 (data security)、データの保全性 (data dignity) を保証する手段を必要とする。

データの利用については、そのデータを削除、追加等の表現形式についての変更とデータ値の変更、参照ができるということと、参照のみが可能であるということとを区別する。システムはKEYを定義、削除できるという機能をもつ。

KEYの定義、削除はシステム管理の項で説明することにし、ここではあるKEYが割りあてられたものとしよう。システムの実行の先頭で、必ず自分のKEYを示すこととし、その実行中に定義されたRECORDは、そのKEY (KEYの名称)を組にして記録される。このRECORDは、定義した人 (KEY)の内容によって、RECORDの修正、削除、データ値の更新、参照をできる人 (KEY)と参照のみが可能な人 (KEY)が自動的に割りあてられることになる。このKEYの相互関係は、そのRECORDを定義した人を含めて、使用者各自が自由に決めるべきことではなく、組織全体の立場から規定

すべきことである。このため、このようなKEYの相互関係はシステム管理者が担当することになる。

もう一つの考えは、データの参照についての規定は、そのデータ(RECORD)を定義した人が決めるべきだということである。したがって、データを参照できる人は、それを定義した人を含めて、個々別々に決められるので、組織全体としては、間接的に個々の部門または人間に指示することによってなされる。

この場合は、RECORDを定義するとき、削除、修正、更新、参照が可能なKEYと参照のみが可能なKEYを指定する。どちらについても、その指示をする人は、KEY名称そのものは、他の部門または人間については知ってはならないのだから、KEY - FAMILYを指示する以外にない。だから、1つの部門または1人の人間であれば不都合なのである。それで、削除、更新等全てを操作できるのは、そのKEYだけとし、参照のみ可能なのは、必要な部門に対応するKEY - FAMILYを指示することになる。

これらの方法のどちらを採用すべきかについて、断定的なことはいわないことにし、とりあえず、前者の方法によって、データの機密保護と保全性を保証することにしよう。個々の企業によってどちらでもそれなりに納得のいく方法だからである。

つまり、データを定義したKEYによって、自動的に以後の操作が可能なKEYを割りあてるようにシステム管理者が決めておく。これ以後の説明は、システム管理の項ですることにする。

念のため、MDSの場合に、更新可能とは、カードからの入力、テープからの入力、演算式の左辺として、シミュレーション言語や分析手法の出力としての場合をいう。参照可能とは、演算式の右辺として、シミュレーション言語や分析手法の入力として、レポート等の端末装置やプリンターへの出力としての場合をいう。

(5) 次に、記録するデータのデータ属性について考えよう。データ属性は、実数、整数、ホレリス・タイプ、論理値等があるが、MDSの場合は、実数、整

数、ホレリス・タイプに限定してよい。データ属性が特に問題となるのは、その表現のために必要な長さが定められない場合である。しかし、実数と整数はいわゆるワード・マシン (word-machine) の場合の1ワードあるいは2ワーズで十分である。ホレリス・タイプをTYPE-II RECORDのデータとして記録することはないと考えても差し支えないだろう。指標となるデータはホレリス・タイプの場合が多くある。

結局、データ属性としては、使用者からみて、

- (a) 10ケタ以内の実数
- (b) 10ケタ以上の実数
- (c) 10ケタ以内の整数
- (d) 10ケタ以上の整数
- (e) 4文字以内の文字列 (ホレリス・タイプ)
- (f) 8文字以内の文字列
- (g) 12文字以内の文字列
- (h) 16文字以内の文字列

の8個のタイプで十分であろう。

TYPE-II RECORDは、そのRECORD内のデータは全て同じ、データ属性とし、TYPE-I RECORDは各ITEMについて、同一データ属性としてあまり不便はないはずである。各タイプに固定して固定長とする方が、記録のための領域がかえって少なくすむ。例えば、TYPE-I RECORDのITEMが何かの名称を表現しているデータの場合、各データによって、その文字列の長短があるかもしれないが、それは最大長に統一しておく方が便利である。

(6) このようなRECORD、FAMILYが定義されると、システムは、データそれ自体とそのRECORDについての情報を記録する。後者の方、つまりスキーマ (schema) に記録される情報は次のとおりである。

TYPE - I RECORD の場合

- ① RECORD 名称
- ② TYPE (= 1)
- ③ このRECORDのデータの記録されているアドレス
- ④ データサイズ
- ⑤ ITEMサイズ
- ⑥ 各ITEMの名称(1つまたは複数個)
- ⑦ 各ITEMのデータ属性(")
- ⑧ このRECORDを指標にしているRECORDの名称
(1つまたは複数個)

TYPE - II RECORD の場合

- ① RECORD 名称
- ② TYPE (= 2)
- ③ このRECORDのデータの記録されているアドレス
- ④ このRECORDの次元数
- ⑤ 各次元に対応する指標RECORDの名称 (複数)
- ⑥ データ属性

この他に、検索効率と検索範囲を限定するために、いくつかのRECORD 名称とKEY 名称等を結びつけたテーブルが作られることになる。

FAMILYは、スキーマとデータそのものの区別はなく、そのFAMILYの名称と要素であるRECORD およびFAMILY の名称を記録するテーブルが作られる。

3.5.3 記録構造

(1) 前節で説明したように、MDSでは、大量のデータを組にして取り扱うことになるので、このような組は順繰成で連続して記録することが望ましい。

記録するデータの組は、各RECORDのスキーマとしてのテーブル、データ、

FAMILY別の情報を記録したテーブルがある。この他にデータと直接関係ないが、KEY およびKEY - FAMILYのテーブル、後で説明するが「モデル」を表現している情報のテーブルがある。このような、一つ一つのデータの組を順編成で記録し、なおかつ、以下のような事態に対処できるようにする。つまり、各データの組は、それ自体として、削除されることもあり、サイズが長くなったり、短くなったりする。

(2) このために、記録方法として、いわゆるページング (paging) を採用するのが適切である。ページング方式はオペレーティング・システムで最近使われ出したものであるが、これから説明する内容は、必ずしもその方法と完全に同じではない。これはこれとして理解してもらいたい。

記録媒体上に、いくつかの領域を設定する。これをNOTE と呼ぶことにする。各NOTE について、最小記録領域 (例えば、1ワード) を単位として、同一のサイズをもつ領域を作る。これをPAGE と呼ぶことにする。1PAGE 中の最小記録領域のサイズをPAGE サイズ、1つのNOTE 中のPAGE のサイズ (個数) をNOTE サイズと呼ぶ。NOTE によって、NOTE サイズ、PAGE サイズは共にいろいろである。

各NOTE には、特別の記録領域を追加する。これを索引と呼ぶ。索引は、各NOTE に対して1つずつ割りあてる。その中味は各NOTE のNOTE サイズと同数の最小記録領域がある。

RECORD等、ある一つのまとまったデータの組は、必ず、このNOTE のあるPAGE の先頭から記録する。そのPAGE で全部記録できなかった場合は、他の任意のPAGE に続いて記録する。このNOTE に対する索引の要素は、空白PAGE , 続きのPAGE がある、続きのPAGE がない、の区別を表示する。そして、続きのPAGE がある場合は、そのPAGE の最後のワードにそのPAGE のナンバーを記しておく。

あるデータの組が削除されたときは、そのデータを記録しているPAGE の全てについて対応する索引の要素に、空白PAGE の印を記す。データが追加

する場合は、その PAGE 内でおさまらなければ、任意の空白 PAGE へ続ける。こうして、事実上無限の連続領域がとられることになる。

この方法によると、PAGE サイズとデータの組のサイズとの関係を均衡させておかななくては、問題がある。まず、PAGE サイズが大きすぎると、空白部分が大きくなる。逆の場合は、1つのデータの組がいくつもの PAGE に記録されることになる。だいたいの基準で、1つのデータの組と PAGE サイズが同じであるのが望ましい。特に問題なのは、RECORD のデータでありこれはサイズが大きいし、使用者にしかわからない。それで、PAGE サイズの異なる NOTE をいくつか用意して、そのどの NOTE に記録するかを、使用者が指定する。NOTE 自体を定義するのは、システム管理の機能とする。

PAGE	NOTE	索引
1		0
2	I N F O 4	1
3	I O N	2
4	R M A T 3	1
5		0

図 3 - 7

3.5.4 データ管理の内容

(1) データの管理についての操作は、表現形式上の操作、すなわち、データの記録、追加、削除等と、データ値の設定に関するものとを区別する。データ値の設定は、MDS の場合であるが、分析・処理の結果としてなされる場合、カードやテープからの入力等多様である。

表現形式上の操作は、RECORD の定義、削除、修正、RECORD-FAMILY の定義、削除、修正である。

(2) RECORDの定義は、ある論理的に構成されたデータの組を設定することであり、データ値を取り扱うものではない。RECORDの定義により、システム内にそのRECORDのスキーマが記録され、データ値を記録するための記録媒体上の領域が確保される。

RECORDの定義に必要な情報、すなわち、使用者がシステムに与える情報は、スキーマとして記録される。これは既に説明した。

RECORDの削除は、記録されているデータおよびスキーマをシステムから取り去ることである。この以後、このRECORDは操作の対象にならない。

RECORDの表現形式上の修正は次のとおりである。

(a) RECORD名称の変更

(b) TYPE - Iの場合、データサイズの変更、ITEMの追加、削除、ITEM名称の変更、各ITEMのデータ属性の変更。

(c) TYPE - IIの場合、データ属性の変更。

FAMILYの定義、削除、FAMILY名称の変更、要素であるRECORD、FAMILYの追加、削除がFAMILYについての機能である。

3.6 データの入出力

3.6.1 記録媒体

データの値を設定するための諸々の操作も、通常は、データ管理機能として取り扱われるのだが、MDSではデータの値の設定はいろいろあり得るので、表現形式に関する操作のみをデータ管理とし、データ値についての操作は別個に検討しよう。この中で特に、システムの直接的な記録媒体（物理的装置としてはディスクを想定している）に記録されているデータを大量に他の記録媒体へ移したりする機能だけを、ここでは取り扱う。

システムが直接的にデータを記録するのはダイレクト・アクセス（direct access）が可能な大容量記憶装置、つまりディスクあるいはドラムである。

しかし、この記録媒体に記録しているデータを、他の媒体にも記録しておく
と便利である。使用者が何らかの間違いによって、意図しないデータ値を記録し
た場合に、元の値を保持しておく、再度処理をやり直せばよい。このための
媒体としては、テープで十分である。

使用者がデータを入力する場合には、端末装置を使用することもできるのだ
けれども、大量のデータを一度に入力する場合には、カードから（カード・リ
ーダー）入力するほかない。このような場合はかなりあり得る。

3.6.2 データの入出力の単位

(1) 記録媒体間でのデータ入出力単位

前節での説明の中で出てきた、システムが直接的にアクセスする記録媒体を
直接記録媒体、データを保存するために記録しておく媒体を間接記録媒体と呼
ぶことにする。直接記録媒体と間接記録媒体の間でデータを移し替える場合の
操作について考える。

この場合のデータの移し替えは、RECORD または FAMILY を単位として
おこなう。大量のデータを一度に簡単に移し替える必要があるからである。デ
ータのコードを変更する必要はないから、単にその名称のみを指定するだけで
十分である。間接記録媒体上のデータは重複していてもかまわないし、その方
がよいこともある。

(2) カード入力のデータ単位

カードからのデータの入力も RECORD を単位としておこなう。ただし、こ
の場合、カードの配列と定義した RECORD の配列方法とが異なることがあ
る。使用者はなるべく、定義にしたがって、カード上のデータの配列方法を考
えるのがよいのであるが、必ずしもこうはしないこともある。

コードの変換は必要なので、FORMAT 文を指定する必要もある。結局、
TYPE - I RECORD は 2 次元配列（データサイズ × ITEM サイズ）とし、
TYPE - II RECORD はその次元の多次元配列として、FORTRAN と同様

の入力形式にするのがもっとも適している。こうすると、使用者が、この知識を必要とするが、さしてむずかしいことではないし、このような大量のデータの入力は、各使用者が全てしなくてはならないものではない。個別データの入力は、端末装置を利用して、会話的におこなうようにする。

3.7 レポート作成，情報検索

3.7.1 レポート作成の手順

レポート作成 (report writing)，情報検索 (information retrieval) も、通常、データ管理に属する機能であるが、前述のとおり、一応分離してここで取り扱う。もちろん、前述のデータ管理方式を前提とするので、内容的には密接に関連している。

MDSで取り扱うデータを、使用者が端末装置およびプリンターに出力させる場合、どのような手順がよいかということに問題を限定する。データの分析、処理のための情報検索は別に取り扱う。

使用者が、データを出力させる場合には、ある程度大量のデータを組にして表の形式でまとめると便利なが多い。そのため、報告書の作成の手順からまず考える。元々MDSで取り扱うデータは、ある分析上の項目によってまとめられた特徴ある形式の表によって表現することが、データの収集、表示過程でおこなわれているから、そのとおりの表を記録上の表現形式として採用した。使用者にとってもっとも頻繁な操作は、項目ごとの表を、指標をつけて出力することである。この場合、使用者は対応するRECORDの名称を指定すればよいのだが、その名称は知っているはずである。そうでなければ、このデータを出力しようという発想は出てこない。正確な名称を忘れていたことはあるかもしれないが、そのRECORDに対応するデータ(群)の意味は知っているものとしてよいだろう。

次に、あるRECORD内のいくつかのデータを取り出す場合について説明す

る。このときは、指標であるデータに限定をつけて、対応するデータを取り出すことになる。逆に、分析上のデータ値に限定をつけて、それに対応する指標データを取り出すこともある。

次に、多少複雑であるが、異なる項目 (RECORD) に属するデータ間の関係を利用する場合を考える。例えば、年度別、支店別の売上高、および同様の利益という2つの項目 (RECORD) について、データの相互関係を調べる。これは、共通の指標を媒介して検索する。ある値を上まわる売上高をあげた年度、支店は、利益としてどのような値になっているか、というような場合である。

3.7.2 レポートのフォーマット

レポートのフォーマットについては、作成手順との関係で変えるのがよい。まず、1つのRECORDを単位として出力する場合は、システムの側に必要な情報を全て記録してあるので、標準的な表を出力することができる。この場合は、指標付きの項目別の表になる。指標を構成するITEMの一部を取り除くことも容易にできる。

RECORDの集合としてのFAMILYの場合も、全く同様である。構成要素であるRECORD別に順次、適当な形式で出力する。

RECORDの中のある特定のデータを出力する場合には、多様な形式になるので、使用者が必要な情報を指示しなくてはならない。

指標を共通にするいくつかのRECORDを出力する場合は、それらRECORDのデータを合成して、ある程度複雑な形式になる。この場合も使用者が特定の指示をしなくてはならない。

このような場合は、多少の知識が必要である。それで、次のような方法も合わせて考えるようにする。出力については、まず、端末装置に標準的な形式で、システムが一方的に表示する。使用者はこれを基に、部分的な修正をする。この方法によると、全く専門的な知識を必要としない。

3.8 システムの管理

3.8.1 データの機密保護と保全性

(1) 機密保護

システムは不特定多数の人によって利用されるが、その中に特別のユーザーの存在を想定する。ここでは、そのようなユーザーをシステム管理者と呼ぶことにする。

システム管理者は特別の KEY (KEY 名称) をもつことによって区別する。システム管理者は KEY を定義、削除することができる。また、いくつかの KEY を組にして KEY - FAMILY を定義し、削除することができる。システム管理者は KEY を定義し、ユーザーに割り当てる。したがって、各ユーザーは他人の KEY (KEY 名称) を知らないのは当然である。

(2) 保全性

システムに処理を実行させるときには、必ず最初に KEY を指定する。RECORD を定義した KEY が明示されることになる。以後この RECORD は、修正、削除、データ値の設定、参照がおこなわれる。これらの場合に、この操作をしてもよい人間 (KEY) を規定しておくのが、システム管理者の一つの役割である。この場合には、前記全ての操作が可能な人間 (KEY) と参照のみ可能な人間 (KEY) を規定することはすでに説明した。もっと細かく、例えば、修正、削除とデータ値の設定と参照とに分けることも考えられるのであるが、一応前記のとおりの区別で十分であろう。大切なことは、MDS の場合、参照に関する限定よりも、データをこわさないことを配慮する必要があるということであろう。

システム管理者が 2 種類の KEY を各 RECORD に割りあてる形式的手順は、定義した KEY と 2 種類の KEY との対応をつけておくことである。例えば、前記の全操作が可能な KEY は定義した KEY と同じものに限定し (RECORD を定義した人がその RECORD を修正、削除、更新、参照の権利と責任をもつ)

参照のみ可能なKEYはいくつかのKEYからなるFAMILYを対応させる。こうするのが常識的であるが、しかし、例えば、2人の人間または部門等KEYを割りあてられた人がいて、この中のどちらかがあるFAMILYを定義し、その後の操作は定義した人とは別の人でもよい、という場合のように必ずしも固定的ではない。

KEYについての機能は、KEYの定義、削除、KEY - FAMILYの定義、削除、KEY - FAMILYの要素の追加、削除である。これら一切の機能は、特別のKEYをもつ人にも実行可能であるものとする。

3.8.2 記録媒体の管理

システムが直接的、間接的に利用する媒体は、直接記録媒体（ディスク、あるいはドラム）と間接記録媒体（テープ）である。直接記録媒体の中には、いくつかのNOTEが設定される。このNOTEを定義するのが、システム管理者の一つの仕事である。さらに、記録媒体そのものを定義することも必要になるが、これは、オペレーティング・システムとの関係が深く、システムで提供する機能ではない。

テープの場合は、各使用者がそれぞれ、使用してもかまわないのだが、相互に間違えると（例えば、他人のテープに書き込み、データを消してしまう等）、トラブルが生じる。それで、統一的に管理をするために、次のような規定を設ける。

各テープに名称をつける。その名称はシステムに記録する。このとき、操作可能なKEYとの関連を合せて記録しておく。テープとのデータの移し替えを、使用者が要求すると、KEYをチェックしてから、そのテープの名称をオペレーターに表示する。これはシステム管理上の機能であるが、システム管理者がおこなうのではなく、各使用者がそれぞれこの機能を利用する。

3.9 データの分析・処理

3.9.1 データの分析・処理

(1) アプリケーション・プログラムの階層

アプリケーション・プログラムは基本的に2つの作成方法がある。問題オリエンテッドに処理手順を決めていく方法と、問題とは独立な手法中心に作成していく方法である。もちろん、両者に厳密に区別できない場合もあるが、この方法の違いは常識的に明らかである。我々の提案するシステムは、アプリケーション・プログラムの中のデータ管理、入力、出力その他の機能を分離しているので、アプリケーション・プログラムの中のデータの処理・計算に限定して、プログラムを作成することになる。ここまではデータ・ベース管理システムと同様である。しかし、データの処理・計算に限定しても、MDSの対象とする処理内容はやはり複雑である。アプリケーション・プログラムの中には、各問題に固有の処理手順と問題とは独立な分析手法が混在している。また、このようになっているので、分析手法もその問題に応じていろいろ変形していることが多い。使用者の指定すべき処理手順は個別問題に固有のものだけでよいのであり、その部分を可能な限り少なくするためには、分析手法、シミュレーション言語等をそれ自体として、システムに組み込んでおき、使用者がそれらを使って、各個別の問題に対応した処理手順を表現するという方法が適切なのである。使用者のニーズからのプログラム全体の独立、である。そして、このような分析上のツールはいろいろな問題で使われるものが多いのである。

(2) 問題処理と分析手法

あるプログラムが使用者のニーズから独立であるか、問題オリエンテッドなのかという区別は、常識的に明らかであるが、そのプログラムを作成するにあたって、システム作成者が、それを利用する使用者に処理についての知識を一切聞く必要がなく、逆に使用者が、システムの中味について知らなくとも、他の文献等によっても完全に知ることができるかどうかを判断基準とするのがよい。

このような意味での、処理ツールとしてのデータの分析、処理、計算のためのプログラムを必要なかぎり、システムに組み込んでおくことにする。しかし、これらのプログラムは、互いに独立であって、容易に追加、削除できる。だから、どのプログラムを組み込んでおくかということは本質的なことではない。必要に応じて、追加すればよいし、それは、使用者とは何の関係もない人に注文しても全く不都合なことではない。ここでは、以上のことを前提にした上で、一応の基準として、データの分析、処理、計算のためのプログラムを解説しておく。

(3) ここでのもう一つの課題は、前に説明した、データの管理方法、特にデータ構造が、いわゆるコア上の計算の対象として利用される際のデータの表現形式と適合していることを具体的にみていくことである。

概要をいえば、配列 (array) を中心とするデータの表現形式は、だいたいの場合は、そのまま、計算の中で使用する表現形式と同一であるから、RECORD を単位として、記録媒体から移し替えればよい。だから、使用者は RECORD として定義したものを、そのまま、計算の対象として記述すればよい。

ただし、RECORD 中の1つの次元についてのデータの列を利用したり、配列をし直して利用する方が便利なこともある。完全に一致するわけにはいかない。このような場合には、その実行中でのみ有効な配列 (array) を定義するようにする。これは演算の一部としての機能とする。単一変数は、そのまま記述するようにし、その他に数字と文字列をそのまま記述できるようにする。このような、明示的であると否とを問わず、コア上の領域はシステム側としては、あらかじめ確保してあるワーキング・エリア (working area) に割りあてる。

3.9.2 演算・関数

(1) 演算式と関数の形式及び特別に組み込まれる関数

演算式は通常の入力文 (例えば FORTRAN 等の) の形式で記述することにな

る。この代入文の中に、いわゆる関数を混在させて記述することもできるようにしておく方がよい。ある処理を関数とするか、別個の分析手法とするかの区別は、その出力結果が単一データであるか否かによるのがよい。

関数としては、FORTRANライブラリー・関数とこれ以外によく利用されるものとのを、なるべくたくさん組み込んでおくと、使用者は便利である。例えば、いわゆる Σ 演算、記述統計学上の統計量、すなわち、平均値、分散、相関係数等である。

この中で、特記しておくべきだと思うのは、いわゆる Σ 演算である。例えば次のような演算式、

$$\bar{x} = \sum_{j=1}^N (x_j + \sum_{i=1}^M (y_{ij} + 2)(z_{ij} - 1))$$

がある。 Σ 演算はあるデータ列の単純な合計直を求めるものではなく、式全体を対象にして合計値を求めるものである。このような、式を直接記述してもよいとする、言語または処理プログラムはないようである。これは、論理的に困難なためか、計算時間が非常にかかるためであると思われる。しかし、経営計画のために、使用者が記述する演算式は、たいした量にはならないはずである。むしろ、この演算式がそのまま書けるようにする方が便利である。それで、^{*}カード・ベース・システムの作成に際して、この Σ 演算をそのまま記述することができる、システム側のプリ・プロセッサを作成した。その方法はここでは省略するが、今後これを使用して、処理時間を計測してみるつもりである。

(2) 処理時間の問題

演算式を解析し、実行するために、システムがもっているプログラムは、このシステム自体を記述する言語によって、その処理方式は多少異なるが、コンパイラに対するソース・プログラムを生成するためのプリ・プロセッサではなく、いわゆるコンパイル・アンド・ゴーの形になる。したがって、その都

※ 報告書中 3-14 「カード・ベース・システム」

度、解析と実行をおこなうので、計算量の大きい、頻繁におこなう計算の場合は、計算時間が問題になる。しかし、そのような計算は、それぞれ分析手法や関数として組み込んでおくことになるものと考えてよいだろう。処理時間は分析手法や関数がどの程度使用者にとって有効なものであるか、すなわち、使用者自身によって記述する演算式のもつ割合との関係で考えるべきことである。

(3) 分析手法

① 分析手法の範囲

分析手法として、どのようなものが必要であるかということは、利用する場合の条件によって決めるべきことであるが、経営計画過程で使用する分析手法を一応の参考として列挙しておくのも意味のあることであろう。

統計学、特に多変量解析論の分野については、次のとおりである。

(a) 主成分分析

(b) 因子分析

(c) 分散分析

計量経済学の分野では

(d) 回帰（最小二乗法）

(e) 二段階最小二乗法

(f) 制限情報最尤法

(g) 時系列分析（EPA法，CENSUS局法）

経営科学の分野では

(h) L. P. , Integer L. P.等の数理計画法

(i) PERT, CPM等のスケジューリング手法

さらに、手法とはいえないが、よく使う汎用的なデータ処理として、

(j) ソート（SORT）

(k) 移動平均法

等が挙げられる。管理会計、経営分析等の財務分析上の手法は、計算手続としては単純であるから、しいて手法としてプログラムを作成しておく必要は

ないであろう。

② 分析手法の相互独立性と関連、その他

分析手法は、互いに独立であって、個別に作成すればよいのだが、ある手法によっては、他の手法を使用しているものもある。例えば、二段階最小二乗法は通常の最小二乗法を用いる。この場合、そのプログラムを用いる方がよい。こうすると、任意に、削除するわけにはいかないが、実際には一度組み込んだ手法を削除することの効果は何もない。常時、コア上にセットしておくわけではないからである。もし、問題があれば個別に作成して、事実上重複していてもかまわない。

どちらにしても、このプログラムは簡単に作成できる。使用者の意図とは無関係であるし、通常、このようなプログラムは既にあるから、それを基に修正してもよいし、新しく作成するとしても、正確な処理をしているか否かのチェックが容易にできるからである。

列挙した分析手法は、行列演算が中心である。それで、汎用のサブプログラムとして、行列演算を処理内容とするものを作ることとすると、分析手法のプログラムを作成するのが楽になる。特に、アレイ・プロセッサ(array processor)をもっているコンピュータの場合は、さらに効果があるだろう。行列演算のサブプログラムはそれ自体を分析手法として、使用者に提供してもよいかもしれないが、この場合、使用者の側の諸々の条件を考慮しておかないと無意味なことになる。

③ データの表現形式

記録媒体上での表現形式が分析手法に対する入出力のためのデータの表現形式と適合していることを具体的に吟味してみよう。もちろん、全てについてみてみなくてはならないし、現にそのような吟味はしたのであるが、ここでは、その中の簡単な例を挙げて、それについてみてみよう。

(1) EPA法の場合：EPA法の場合の入力データは次のとおりである。

(a) 分析(分解)の対象になる1次元配列のデータ

(b) 期間，加法的分解か乗法的分解かの別，月次データか四半期データかの別を示すそれぞれ単一データ（パラメータ）

出力データは，次のような2次元配列である。（図3-8）

(c) 最小二乗法の場合：最小二乗法の場合の入力データは次のとおりである。

(a) 被説明変数を表す1次元配列のデータ

(b) 説明変数を表わす2次元配列のデータ

ただし，各説明変数を別個に1次元配列としてもよい。

(c) 処理上の情報を示す単一データ

出力結果であるデータは各種統計量を表わす表である。使用者の側のオプションによるものもあるし，全部を例示する必要はないからその中から，代表的な表を挙げてみると，次のようになる。（図3-9）

以上みたように，入力，出力の各データは指標つきの配列表現になっているものと，単一データとがある。単一データは，取り扱い上何の問題もない。入力，出力データは共にRECORDの表現形式と同様であるといつてよい。

(4) シミュレーション言語

① シミュレーション・モデルにおける変数と関係式の設定

シミュレーションとは，分析対象とするいくつかの変数，それら変数相互の関係式を規定した場合に，ある変数に特定の値を設定したときの，他の全ての変数の値を見つけることである。ただし，このようにいえば，だいたいの処理プログラムは，入力データに対する出力データの値を決めるものであるから，なんでもシミュレーションとなってしまう。実際には，シミュレーションというときには，関係式は単純な方程式で，それは局所的な関係を規定していて，逐次的に変数の値が決められていくもの限定していると思つてもよい。シミュレーションはあるいくつかの変数については，外生的に値が定められる。この外生的に支えられる変数は，環境，政策を表現しており，これらの変数を特にパラメータ（parameter）と呼ぶ。シミュレーションは，パラメータにいくつかの値を入れてみたときの，その分析者の知りたい状態，

項目 年,月	原データ	季節指数	季節変動 調整済	傾向・循環	不規則変動

図 3 - 8

相関係数行列

回帰係数, その他

変数	変数	定数項	回帰係数	係数の 標準偏差	T-値	偏相関
			変数	変数	"	

*の部分は空白である

図 3 - 9

つまり変数の値を探ることが目的である。あるいは、パラメータの値の変化と知りたい変数の値の変化との対応関係を探るのも一つの機能である。

シミュレーションの実行を、ある入力データによる他の変数の値の設定までに限るとすると、この実行を逐次的に数度おこなうことも考えられる。もちろん、この場合には、実行中に前段階での結果を情報として受けとり、それを参考にして、次の実行をおこなうことになる。これがゲーミング(gaming)と呼ばれるものである。だから、ゲーミングはシミュレーションとほとんど同じものであるといってよい。データ入力(パラメータの設定)をする人を参加者というが、それは一人でも複数でもよい。

コンピュータ・システムの側からみると、シミュレーションのためのプログラムの作り方は、関係式をプログラムの中に組み込み、この場合は変数の個数も決めなくてはならないが、使用者がデータを入力するだけにしておくものと、関係式それ自体も使用者に入力させるものがある。後者の場合は、関係式についての表現形式を規定しておき、その基でのみ表現される関係式を使用者が入力する。そうでなければ、一般のコンパイラと差がなくなる。この二つの作り方は、問題向けに作られるものと、汎用言語として作られるものに対応している。

② シミュレーション・モデルの拡張と分離

(a) MDSにおけるシミュレーション

MDSとして、シミュレーションのための処理プログラムを組み込む場合に、関係式までプログラムの中に組み込むようにすべきか、関係式は使用者が指定するようにすべきかを考える。シミュレーションは、結局、ある政策や環境に対する意思決定の参考になり得る評価尺度である変数の値をみることが目的である。この場合、その政策や環境と評価尺度との関係自体が保証できるものであればさして問題ない。しかし、このようなことはむしろ稀なのである。変数相互間の関係こそ実は問題なのである。この意味で、多くの場合、シミュレーションといっても推定の要素を含むのである。

定型化されていない問題について、シミュレーションを利用することを主張する人も少なくない。しかし、極所的にせよ、関係式を固定的に設定できる問題は、むしろ定型化された問題というべきであろう。真に定型化されていない問題は、モデルの設定から始めなくてはならない。

だから、シミュレーション・モデルの中に占める関係式の役割は重要であり、それが一番問題なのである。使用者は、いかなるシミュレーション・モデルによったかを吟味せずして、その結果をみるのは危険である。

(b) シミュレーション処理の問題点

シミュレーションに関わる処理を、問題別にプログラム化していない場合は、次のような問題がおこる。すなわち、システムに組み込んであるシミュレーション言語が十分汎用的でなければ、結局、使用者が演算、関数を利用して記述しなければならず、非常に不便になる。また、あまりに汎用的な言語であると、各問題によっては必要のない処理まで含んでいることになり、そのため、使用者のおこなう入力作業が複雑になり、またシステムの側としても不必要な処理をしなくてはならない。

前者の場合についていえば、実際のシミュレーションのための処理で、そのための言語を用いなくて、一般のコンパイラ言語によってプログラマとして記述している場合は、関係式の表現形式が多少複雑なためであることが多い。それで、関係式の表現形式を単なる方程式に限らず多様なものを許容できるような言語を開発することが望ましい。

さらに、既存のシミュレーション言語は、各々が汎用性を求めるあまり、不必要な処理を含んでいることが多い。それで、シミュレーション言語は構造を規定し、複数個、別個に開発し、システムはそれを全て組み込んでいるようにするのがよい。もちろん、このためには、MDSのようなトータル・システムが前提である。

現在、計画過程で主として用いられるシミュレーション言語またはシミュレーションのためのプログラムは、計量経済学におけるファイナル・テスト、

トータルテストと、システム・ダイナミックスがある。これらは共に前述したとおり、関係式の表現形式がかなり固定的である。もちろん、これらのものはそれなりに有用なものである。

それとは別に、計画過程での意思決定手順そのものを表現できるシミュレーション言語が必要である。つまり、ある程度最適化を求めたり、満足基準によって変数の値を設定したり、使用者の計画手順としてのアイテレーション（物理的時間軸ではなく）を許容し得るような構造をもった言語を開発することである。

3.10 「モデル」の管理

3.10.1 「モデル」の表現

「モデル」を表現する、ということについて考えよう。システムは、情報処理機能別に問題あるいは問題領域とは独立な処理ツールを機能としてもつことにし、使用者がこれらの処理ツールを使って、処理手順を記述することにした。意思決定者またはそのスタッフは、計画策定過程で直面する問題を設定し、必要な情報を考え、それらの情報を生成するための処理を定式化し、あるいは報告書を作成したり等々の一切の作業をする。この過程で、処理システムを有効に利用するとよい。

問題領域とは独立な処理ツールを前提したときに、どのような形式の記述方法によれば容易に、正確に問題の処理手順を表現できるのだろうか。もっとも簡単であるが、しかしもっとも頻度が高いと思われる表現は、処理ツールをいくつか、直列的に指定するというものである。しかし、処理の内容を、その処理最中のある変数の値によって変更したい、ということも当然ある。また、コンピュータによる処理の特徴は、類似した処理を数度くり返すという方法、すなわち逐次的処理（recursive processes）が多いことである。このように、単に処理ツールをいくつか並べる（その最初のものから順々に実行してい

く)という表現方法は、限られた適用能力しかもたない。

このようなことを考えると、使用者の記述可能な形式を若干複雑にしておく必要があることがわかる。しかし同時に、各処理ツールによる処理量と呼ぶべきもの(処理時間や取り扱うデータ量、計算量)は大きいし、使用者にとっては論理的にも容易に記述可能である必要がある。それ故、使用者が「モデル」を正確に記述するように、「モデル」の表現について厳密な規定を必要とするし、それは一般に適用できて、なおかつ単純なものではなくてはならない。

それで、一般的なアルゴリズムの表現形式として、コンピュータに限らず共通に記述する方法を採用するのが適当であるとする。それは、以下のような構成になっている。

- (a) 定義された処理、すなわち、すでにそれなりに意味が明らかなもので、実行できる処理。
- (b) 判断分岐、すなわち、基本的には、ある命題であるが、その命題の真偽が判定可能なもの。
- (c) 処理の順序の指定。

以上の3つの種類のどれかに属する処理を要素としてもつ、順序のある集合である。この表現形式にしたがって、「モデル」を表現することにする。

「モデル」の表現形式

(1) システムが使用者に提供する処理ツールは各々固有の名称をもっている。各処理ツールをシステムが実行するためには、それぞれ入力情報を指定されなくてはならない。処理ツールの名称とその処理ツールに対応する入力情報を組にした情報を、「定義された処理」と呼ぶことにする。

次のようにして定義される命題を「判断分岐」ということにする。

- ① 左辺, 判断オペレータ, 右辺という形式になっている。ただし, 左辺, 右辺はともに, RECORD, 配列変数, 変数, 数字, 文字列のどれか, であって, RECORD, 配列変数, 変数は, すでに定義されているものである。判断オペレータは, =, ≠, <, ≤, >, ≥; である。

② 「判断分岐」；論理オペレータ，「判断分岐」という形式になっている。ただし，論理オペレータは， \wedge ， \vee (AND，OR) である。

①，②どちらかの条件を満たす情報を「判断分岐」という。論理オペレータを用いると「判断分岐」の表現は，非常に複雑な形式になり得る。論理オペレータを数個用いると AND と OR の実行順序が問題になる。これは通常形式にしたがい，AND の方が順序が先であるということにし，左右のカッコではさむことも許す。カッコの中の順序は先になる。

「判断分岐」は，実行されると，真偽が判断され，真の場合は，次の処理，そうでなければ，次の次の処理へ実行が移ることにする。

「定義された処理」，「判断分岐」のどれかに属する情報を要素とする有限個の要素をもつ順序のある集合を考える。この集合は要素を列挙して表現でき，各要素は対応する自然数を添字 (suffix) としてもつ。

$$A = \{ a_1, a_2, \dots, a_n \}$$

「実行順序の指定」とは，この対応する自然数を示す情報である。

(2) 「モデル」とは，次のような情報を要素とする，順序のある有限集合である。

① 「定義された処理」，「判断分岐」，「実行順序の指定」の別を示す情報。

② 「定義された処理」の場合は，システムが提供する処理ツールの名称と各処理ツールの実行に必要な入力情報。「判断分岐」，「実行順序の指定」は前述のとおり情報。

(3) ①と②の情報を組にして表現したものを要素とするが，以下のものは要素とならない。

① 要素の中に，「定義された処理」がない場合の他の全ての要素。

② 「実行順序の指定」で，その要素の示す添字のナンバーがその要素の添字と一致する場合の要素。

③ 同じく，その要素の示す添字のナンバーが1からこの集合の要素の個数

までの自然数になっていないもの。

要素の一つもない集合は「モデル」とならない。「モデル」には、各々固有の名称をつけることにし、この名称は、使用者が指定する。

3.10.2 「モデル」を定義する

前節で規定した情報の組としての「モデル」は、もちろん、システムがその情報に基づいて、実際に処理を実行することを想定している。ただし、使用者は「モデル」を情報として、システムに記録させておくことができ、その実行はまた別のことと考えるよいわけである。

使用者が「モデル」をシステムに記録させるのは、類似した処理（全く同じでもよいが）を実行させたいからである。だから当然、内容的にはかなりの場合に固定しておきたい部分と、かなり頻繁に変更したい部分があるはずである。計画立案作業において、全社的にあるいは、ある部門によって、基本的な分断の手順、構造を定めておいて、データの内容をいろいろ変えたり、問題のある点について、さらに詳しい分析をしたりすることもあるはずである。

以上のようなわけで、「モデル」を使用者が定義する場合には、すぐ実行可能であるという意味で完備である必要はない。完備でない「モデル」についても、それはそれとして、システムが記録しておく意味は十分ある。この「モデル」はその後、未定義の部分を補足すればよい。

「モデル」は以下のような階層を設けて定義することにする。

- (a) 「モデル」の名称。
- (b) 「モデル」を構成する各要素について、「定義された処理」、「判断分岐」、「実行順序の指定」の別。
- (c) 「定義された処理」については、処理ツールの名称。
- (d) 同じく、指定した処理ツールに必要な入力情報。
- (e) 「判断分岐」については、その条件式。
- (f) 「実行順序の指定」については、指定する要素の添字のナンバー。

「モデル」の名称は必ず指定しなくてはならない。それ以外は指定しなくとも、その時点では、システムは「モデル」として記録し、未定義の部分のまま残しておく。「定義された処理」、「判断分岐」、「実行順序の指定」の別を指定しないで、それ以後の、例えば処理ツールを指定することはできないことにする。同じく処理ツールを指定せずに、入力情報を指定することはできない。階層を設ける、とはこの意味である。処理ツールに必要な入力情報は通常複数個ある。この中のどれかを未定義にし、その他は指定することができる。入力情報の中には、パラメータ、データその他性質の異なるものが含まれるからである。

3.10.3 「モデル」の管理と実行

「モデル」管理に関する機能は前節で説明した、「モデル」を定義すること、一度定義した「モデル」を削除すること、同じく修正すること、の3つである。

「モデル」の削除は、定義してある「モデル」をシステムから取り去ることで、この後は操作の対象にならない。

「モデル」の修正は、次のとおりである。

- (a) 名称を変更する。
- (b) その「モデル」を構成する要素を追加、削除する。

これ以後は、「モデル」の定義として、情報を与える。

「モデル」の管理とは別に、定義してある「モデル」を実行させるという機能がある。「モデル」の実行は、その名称を指示するだけでよい。「モデル」の実行の際には完全な定義がなされていなくてはならない。

「モデル」を実行すると、この「モデル」に表現されている内容を連続的に処理することになる。この内容は、システムの側からみれば、いくつかの処理ツールを、指定された順序で逐次実行していくことになり、所用者の側からみれば、需要の予測、生産計画についてのシミュレーション、見積財務諸表の作成等々の問題別処理をおこなっていることになる。あるいは、全社的な行動に

についてのシミュレーションであったりする。こうして、問題オリエンテドに作られているシステムと同様の機能をもつことになる。

3.11 使用者とシステムの会話的入出力

3.11.1 端末装置の利用

一般に、コンピュータに、使用者とシステムとの入出力媒体として、グラフィック・ディスプレイ (graphic display) やキャラクター・ディスプレイ (character display) 等の端末装置を利用することの効果は次のことに集約できる。

- (a) 使用者が処理を依頼してから、出力結果が手元に届くまでの時間間隔 (turn around time) を減少もしくはなくする。
- (b) 使用者がコンピュータに対しておこなう入力作業を容易にする。
- (c) 特にグラフィック・ディスプレイの場合であるが、文字による出力の他に図形表示を可能にする。

使用者が処理を依頼してから、出力結果が手元に届くまでの時間間隔の問題は、経営計画の場合、ある程度限定された効果しかないだろうと思われる。計画過程では、かなりの量のデータを組にして参照することが多く、このため、正に逐次的に処理内容を指示するというわけにはいかないことが多いからである。この問題を特に感じるのは、使用者がエラーをしたときに多い。この意味では、入力作業を一つ一つチェックし、エラーをしたときには、その時点でやり直すようにすると効果的である。

図形表示は、今のところ、それほど意味はない。グラフを表示することによる効果が期待される程度であろう。

入力作業を容易にするという点では、かなり効果的である。このためには、それに必要な入力情報についての論理構造を十分考えておく必要がある。以下この点について説明する。

3.1.1.2 会話の論理構造

(1) 問題点

システムの処理様式が使用者の処理要求とは独立に作られるべきことは前述のとおりである。そして、このように構成されたシステムは使用者の処理要求との接点に工夫が必要であることも自明であろう。使用者は問題を定式化し、システムの提供する処理ツールを使って表現しなくてはならない。このとき、システムとしては、その問題の定式化に固有な知識、あるいはその作業についてサポートすることはできないが、それ以外のコンピュータに固有の煩雑な作業については徹底してサポートする。端的に言えば、使用者からプログラミング作業を取り除くことである。このため、システムは使用者のおこなうべき情報入力を論理的に再構成し、質問-応答の列として表現するのが望ましい。この方法は、使用者がプログラミングから解放されること以外に、重要なメリットがある。すなわち、使用者が、コンピュータに処理を要求してから出力結果が手元に届くまでの時間間隔の問題を感じるのは、入力情報にエラーが含まれていたときに多いのであるから、質問-応答の列として表現されるならば、正に逐次的にそのことがわかり、この問題がなくなるのである。

入力情報を分解して、質問-応答の列として表現するとなると、ある質問に対する答によって次の質問が変わるものはなるべく先に質問すべきである。こうすると、質問-応答の列が最小限になる。前の質問の答によって規定される質問の列を組にすると、質問の内容は前の質問に対する答によって全て規定されるから、基本的には木構造 (tree structure) として表現できる。第1の階層は処理ツールを選択するまでで、ここまでは適当に階層を設けることにより、答の候補を全て列挙できる。次の階層は各処理ツールによって異なる。概して、入力情報の論理的関係から質問の順序を規定できる。

ただし、使用者はシステムからの質問に対して的確に解答できるとは限らない。ある質問に対して、別の質問を逆にしたいこともある。カードから入力したいこともある。ある段階で間違いに気づき、やり直したいこともある。それ

で、次のような特別のメッセージを設ける。それをF1, F2, F3, F4とする。F1はシステムに記録されているデータ, 「モデル」の内容を表示することを要求する。この流れにも階層があり, 漸次細かくデータが示される。この流れの最中にF1を要求すると, 元の流れに戻る。F2はカードからの入力を要求する。その時点での状態を前提した質問に対する解答の列がカードから入力されるものとする。カードからの入力の終了時点で論理的にその時点での状態を端末装置に表示し, 元の流れにもどる。F3は質問が前の時点に戻る。この時, それ以前の答の列はそのまま採用される。F4は, 質問の列の先頭に戻る。

以上のようにして, 質問と応答の流れは, きりかわることがある。使用者はこれらのツールを適当に利用して入力情報を表現する。ある処理を要求する時には, そのためのデータを事前に検討しておくのは当然であり, プリンターから出力させておきたいものはその都度指定しておく。分析・処理とデータの参照は交互におこなわれることになるだろう。

(2) 会話の理論構造

システムの実行開始とともに, まず, KEY (KEY 名称) を指定するよう
に, 使用者に求める。その指定されたKEY が定義されていなければ, エラー
である。エラー処理はこれに限らず, エラーの内容を表示し, その時点のまま
にしておく。

KEY 指定の次は, 主要木構造に実行を移す。主要木構造の第一の分岐は,

- (a) 「モデル」の実行
- (b) 「モデル」の管理
- (c) データの管理
- (d) データの入出力
- (e) レポート作成, 情報検索
- (f) システムの管理
- (g) 分析, 処理

である。この表示について、応答すると、各々にしたがって、次の分岐が表示される。「モデル」の実行の場合は、その使用者の に属する「モデル」の名称が表示される。ただし、まだ不完全な定義の段階にある「モデル」は表示しない。使用者は意図する「モデル」の名称が表示されない場合は、実行を「モデル」の管理に移すとよい。

表示された「モデル」の中から意図するものを指定すると、その「モデル」の実行を確認するメッセージが表示される。一般には、「モデル」の実行はある程度の処理量になり、その間に、使用者がやり直しを要求することはできないから、確認を求める必要がある。確認を求めている表示を見て、それに応答せずに、他の処理に実行を移してもよい。その時は、この実行の流れは、そこで中断している。「モデル」の実行の後に、必要なメッセージを表示して、処理の完了を示す。この流れは、そこで終了するので、他の処理に実行を移すか、実行を終了させる。

例えば、このようにして、逐次的に必要な情報を指示し、その流れを変更することもできるようにする。

データの検索、表示は、その使用者に参照可能なRECORD名称が表示され、その選択をするだけである。この場合にRECORD - FAMILY を仲介することもある。RECORD名称を指示すると、そのデータが表示されることになる。

312 システムの機能と利用方法

3121 システムの提供する処理ツール

前章までに、我々の提案するシステムの設計方針、システム構造、各サブシステムについて説明してきた。

この章では、システムの提供する処理ツールを具体的に紹介し、使用者がこれら処理ツールをどのように使うのかを説明する。

(1) データ管理の処理ツール

データ管理に属する処理ツールは、データの組織的な記録、削除等に関するものである。記録されている個別データの値の設定は別の処理ツールを用いる。

① RECORDの定義

RECORDとは取り扱いの単位となるデータ群である。RECORDの定義に必要な、つまり使用者がシステムに与える情報は、次のとおり。

TYPE-I (指標RECORD)の場合は、RECORD名称、データ・サイズ、ITEM・サイズ、各ITEMの名称、各ITEMのデータ属性である。

(例) 指標であるRECORD、例えば、支店について。そのRECORDの名称は、例えば、SHITEN、データ・サイズは3、ITEMサイズは2、各ITEMの名称は、NAME、NO.、各ITEMのデータ属性は、ホレリス・タイプ、整数。

TYPE-IIの場合は、RECORD名称、次元のサイズ、各次元に対応するTYPE-I RECORDの名称、データ属性である。

(例) ある製品の年度別、支店別の売上高の場合には、RECORD名称は、例えばURIAGE、次元のサイズは2、各次元に対応するTYPE-I RECORDの名称はNENDO、SHITEN、データ属性は、例えば実数である。

以上のようなRECORDの定義を使用者が要求すると、システムはスキーマ・テーブル(schema table)を作成し、データ記録領域を確保する。

② RECORDの削除

RECORDの削除とは、そのRECORDをシステムから取り除くことで、データ値の変更ではない。この処理が行われると、以後このRECORDを対象にした処理はできない。RECORDの削除のために、使用者が与える情報はそのRECORDの名称だけである。TYPE-II RECORDの場合は、無条件に削除できるが、TYPE-Iの場合は多少異なる。TYPE-I RECORD

は、他のTYPE-II RECORDの指標になっているのだから、そのRECORDを指標にしているRECORDが存在する限り、削除してはならない。システムはこの点を吟味し、使用者のエラーを指摘する。システムにとってRECORDの削除とは、該当するスキーマ・テーブルを削除し、データ記録領域を解放することである。

③ RECORDの修正

RECORDの修正とは、次のことである。

(a) 名称の変更

(b) TYPE-Iの場合は、データ・サイズの変更。

ITEMの追加、削除。各ITEMの名称、データ属性の変更。

(c) TYPE-IIの場合は、データ属性の変更。

名称の変更は、その時点のRECORD名称と新しい名称をシステムに与える。

データ・サイズの変更は、例えば、SHITEN(支店)というRECORDのデータ記録域を追加するということである。TYPE-I RECORDのデータ・サイズが変更されると、それを指標としているTYPE-II RECORDはそれに合せて、自動的に変わる。指標データの追加は、それが指標になっているRECORDのデータの追加を予定しているはずであるし、データを一部取り除く場合もそうである。もちろん、そくざにその部分のデータ値が設定されるかどうかはわからない。しかし、「近い将来」にはその作業が行われると想定しておくのが自然である。

(例) NENDO(年度)

ITEM = NENDO, NO.	
1960	1
1961	2
1962	3
1963	4

図 3-10

SHITEN(支店)

ITEM = NAME, NO.	
TOKYO	1
OSAKA	2
KYOTO	3

図 3-11

SHITEN NENDO		NENDO			NENDO NO.	
		1	2	3		
1		1.1	1.2	1.3	1960	1
2		2.1	2.2	2.3	1961	2
3		3.1	3.2	3.3	1962	3
4		4.1	4.2	4.3	1963	4
					1964	5

☒ 3 - 1 2

☒ 3 - 1 3

SHITEN NENDO		URIAGE		
		1	2	3
1		1.1	1.2	1.3
2		2.1	2.2	2.3
3		3.1	3.2	3.3
4		4.1	4.2	4.3
5		*	*	*

☒ 3 - 1 4

(2) データの入出力に属する処理ツール

④ カードからのデータの入力。

定義されたRECORDに対応するデータ(値)を設定する一つの方法は、カードから読み込ませるものである。この処理ツールは、大量のデータを一括して読み込む場合に利用されることが多いはずである。

RECORDは多次元配列の形式になっているから、FORTRANのREAD

命令と同一に取り扱う。使用者の指定する情報は、記録されるべきデータの RECORD 内の位置と、FORMAT である。

システムは、データを読み込み、RECORD の記録領域（ディスク内の）に書く。

⑤ テープからの入力、テープへの出力

テープとの入出力は、データを保護することが主たる目的である。使用者が誤まって、意図しないデータ値を設定する場合がある。データを事前にテープに記録しておくこと、このような時には、そのテープからデータを入力し、元の状態に戻すことができる。

テープとの入出力は、データの保護が目的であるから、システム内の記録様式をそのまま用いる方が使用者にとって、取り扱いが容易であり、望ましい。それで、テープとの入出力は RECORD かその集合である FAMILY を単位として行うことにする。コードの変更は必要ないので、FORMAT の指定はない。使用者がシステムに与える情報は、RECORD または FAMILY の名称、テープの名称である。

(3) レポート作成、情報検索に関する処理ツール

⑥ 端末装置への出力

⑦ プリンターへの出力

(4) システムの管理についての処理ツール

⑧ KEY の定義

⑨ KEY の削除

⑩ KEY-FAMILY の定義

⑪ KEY-FAMILY の修正

⑫ KEY-FAMILY の削除

⑬ NOTE の定義

⑭ NOTE の削除

⑮ 間接記録媒体の定義

- ⑩ 間接記録媒体の削除
- (5) 「モデル」についての処理ツール
 - ⑪ 「モデル」の定義
 - ⑫ 「モデル」の修正
 - ⑬ 「モデル」の削除
 - ⑭ 「モデル」の実行
- (6) データの分析・処理に属するグループ
 - (a) 演算・関数
 - (b) 分析手法
 - (c) シミュレーション言語

各グループはそれぞれ、いくつかの処理ツールから成る。それらは互いに独立であるから追加、削除は容易にできる。データの分析・処理に属する処理ツールは、データ値の変更設定である。この場合、取り扱いの単位はRECORDであるが、その他にテンポラリーに定義できる配列、明示的には定義しないが、そのまま記述する変数（単一変数）、数字（実数、整数）、文字列がある。

3122 利 用 方 法

(1) データの収集と問題の分析

経営計画過程で利用されるデータは、企業内の個々の事象や業務レベルでの技術情報だけでなく、外部情報や分析上のデータも多くある。だから、分析に先だって、データを常に整備しておくことは望ましいが、必ずしも実現可能ではない。むしろ、分析上の必要からデータを収集することもあり得るだろう。

また、全てのデータを常にコンピュータの記録媒体に記録しておく必要もない。

データの収集・整理は決して完全にできるものではなく、システムとは離れて、使用者が主体的に行うことにし、この意味でもシステムと使用者とは独立な関係にあるようにしておくのが妥当な方法である。システムには、基本

的なデータだけを常時記録するようにし、その内容は容易にわかるようにしておくことが望ましい。この意味で、データを組にして考えるのがよい方法なのである。

(2) 基本モデルを複数個開発する

我々の提案するシステムは、処理ツールを使って、使用者が主体的に処理内容をその都度記述することを期待している。経営計画では、こうすることがもっとも適していると考えられる、使用者の記述した処理内容は「モデル」として、記録し、以後それを修正して実行する。だから、基本的な処理手順は「モデル」として記録しておくといよい。

この場合、個々の企業によって異なるであろうが、基本的なモデルを固定的に作成してはいけないということに留意しておくことを期待する。できれば、複数個のモデルを作成し、比較検討してみるのが望ましい。企業モデルといわれる、シミュレーターとしてのプログラム等のように、簡単にモデルを変更できないものではなく、これこそが重要な部分であるとして、特別の工夫をしたシステムであるから、特に利用方法によって、その効果が変わるのである。

3.13 ま と め

(1) 経営計画を対象とする情報処理システムの作成には二つのアプローチがある。一つは従来主流になっていた考え方で、現在でもその傾向が強い。それは、経営計画に関する情報処理を調査・分析し、その結果をそのままシステム（プログラム）の中に表現していくというものである。この考え方によると、経営計画における情報処理手順がほぼ機械的に処理システムの中に組み込まれることになる。

我々の提案は、一見合理的なこのアプローチを否定することである。そうではなくて、計画過程での一般的情報処理とは独立な構造をもつシステムを作り、使用者が必要に応じて、これを利用するようにするというものである。もちろ

ん、このことは、計画過程での分析、意思決定の手順を可能なかぎり、定型化していく努力を否定するものではない。むしろ、そのような努力のために、処理システムがネックにならないように工夫することである。

(2) この主張を具体化するために、情報処理機能によるシステムの分割に加えて、データの処理、加工、分析等々の狭い意味でのデータ処理についても問題オリエンテッドな処理内容は機能としてもたず、一切が使用者によって記述されるべきものとした。問題と独立な処理であるか否かの区別は、結局、誰がその処理内容を知っているかということである。

(3) 使用者とシステム（システム作成者）の役割分担を定めたために、事実上、使用者によって処理内容が決められることになる。使用者の意図する情報処理体系（個々の問題）とシステムの情報処理体系は全く独立なものになる。それでこの二つの情報処理体系の接点に特別の工夫をした。端末装置を利用した質問—応答形式による逐次的入力と、「モデル」を管理する。すなわち、使用者の記述した処理手順そのものをシステムが記録するという機能をもつこと、この二つである。

(4) 以上説明したシステムを設計する場合に、技術的な決め手になるのは、システムを構成する各処理において取り扱うデータの表現形式を統一しておくことができるということである。このために、経営計画で用いるデータの表現形式を考察し、それを基に一般のデータ・ベース管理システムとは異なるデータ構造を提案することにした。

3.1.4 カード・ベース・システム

3.1.4.1 カード・ベース・システムの目的

経営意思決定サポートシステム（MDS—management decision support system）は、キャラクター・ディスプレイ、グラフィック・ディスプレイ等の端末装置を使用すると、いろいろな意味で効果的である。

端末装置を使用する場合の、利用方法、効果、限界については別稿で議論したとおりである。[※]

それとは別に、端末装置を使用しないで、経営計画に関する情報処理を適用対象とするシステムを考察する。

各企業において、端末装置そのものと、その使用のために必要なコンピュータ・ソフトウェアの機能（例えば、マルチ・プログラミング可能なオペレーティング・システム等）を備えておくべきか否かは、単にMDSというシステムの効果だけではなく、その他諸々の条件を考慮しなくてはならない。したがって、端末装置を使用できない場合もある。さらに、各企業では、既に開発している個別のプログラムをもっていて、それを利用しつつ漸次発展させて、より完成した処理体系にしていこうという考えもある。これは一つの方法であろう。それに加えて、MDSの本質は端末装置にあるのではなく、要は、計画過程における、多様な情報処理に対処できるシステムであることである。

このようなわけで、別稿で説明した考え方をそのまま適用した、同じ目的のカード・ベースのシステムも作成しておくのは意義のあることである。

3.1.4.2 カード・ベース・システムの機能

使用者との入出力を、カード（カード・リーダー）だけを使用して行うことになる、煩雑な入力作業ができなくなる。具体的には、シミュレーション言語に対する入力情報としての関係式が一番問題である。例えば、システムダイナミクスに基づいたシミュレーション言語であるDYNAMOを使用して、シミュレーションを行うときには、分析者である使用者が関係式を含めて入力情報を記述しているようである。このことを考えると、必ずしも、問題とは独立な手法、言語をシステムとして備えておき、使用者はこれを使って処理手順を記述するという方法は不可能ではない。しかし、一般の意思決定者、そ

※ 本書 3.1.1 「端末装置の利用」

のスタッフに、このようなことを求めると、やはり容易にシステムを使用するというわけにはいかなくなるかもしれない。

カート・ベース・システムを作成しておく一つの理由は、個別処理を主として行っている企業において、それまで開発したプログラムを利用しつつ、漸次MDSの完成した処理システムを利用するように導くことである。

このようなことを考慮した結果、データの分析、処理の部分については以下のようになるのが適当であろうと考えた。

- (1) システムの側としては、データの分析、処理を担当するサブプログラムは、それが分析手法であるか、問題別に作られているものかは関知せず、処理プログラムとして扱う。
- (2) 処理プログラムの目的、機能は、データの分析、処理（いわゆるコア演算）に限定し、入力、出力データは、各サブプログラムの引数として扱う。
- (3) このような処理プログラムとは別に、演算、関数を使用者がそのまま記述できるようにし、それに必要なシステム側の機能を組み込む。

処理プログラムは、基本的には、使用者の側で作成するものと考えている。ただし、使用者の側という場合、意思決定者自身である必要はなく、関連する部門の担当者を想定している。使用者の側としては、処理プログラムの内容を検討し、なるべく分析手法は分離する方向で、各サブプログラムを整理するようになっていくことが望ましい。これは漸次そのように発展させる、という意味である。システムの側としては、各処理プログラムを個々別々に容易に追加、修正、削除できるように工夫する。

もつとも基本的な設計方針である情報処理機能別にサブシステムを構成し、システムは問題とは独立な処理ツールを提供するという方法は何ら変更する必要はない。前述のとおり、データの分析、処理に関する部分を若干修正するだけである。そして、これ以外の処理ツールについては、使用者の入力作業の煩雑さという点からの機能の変更も全く必要ない。

3.1 4.3 設 計 手 順

(1) 解 説

使用者の情報入力を会話的にできないために、使用者にとって容易で、複雑な処理を指定できるように工夫しなくてはならない。

一つの方法として、コンパイラ言語（例えば FORTRAN）を、使用者が使用し、システムの機能をユーティリティ・ルーチンとして、組み込むという方法がある。この方法によると、複雑な処理内容を記述でき、なおかつ、実行効率もよい。しかし、この方法は、その言語を記述できる人にしか利用できない。

システムとして、機能をもっておき、単純なパラメータを指定すると、それに基づいた処理をするようにしておく方法もある。一般のパッケージ・プログラムはこの方法を用いている。しかし、この方法では、処理内容がかなり限定されてしまう。それで、システムに組み込んである処理ツールを指示し、それに必要な情報を合せて指定することと、実行順序についての制御、判断分岐についての処理を組み合わせ、情報入力を行うようにするのがよい。こうすると、使用者は、このシステムの規定する規則にしたがって、単純な入力作業をすればよく、処理内容も、必要があれば、複雑なものにできる。

設計手順は、システムへの情報入力に基づいて、次々に解析し、その解析した情報にしたがって、処理を実行することになるよう、論理構造を作る。メイン・プログラムの設計は端末装置を使用する場合と違い、逐次的に入力情報を読み込み、解析し、各処理ツールに分離するということになる。いわば、一般のコンパイラと同様のものになる。

(2) コマンド形式

使用者の行う情報入力は厳密な規定を設ける。逐次的なエラー・チェックができないからである。使用者は、次のようなコマンド形式によって、処理内容を表現する。

コマンドは、各々全て、

LABEL, OPERATION, OPERAND

の3つの部分からなる。ただし、LABEL は必ずしも指定しなくてもよい。OPERATION が、システムの提供する処理ツールになり、OPERAND は各処理ツールに必要な情報である。

このようなコマンドの列を入力することによって、システムの処理内容が規定されることになる。

3.14.4 処理ツール

前述のとおり、カード・ベース・システムでも、端末装置を利用する場合と基本的に同じである。端末装置を使用する会話的の入出力がないことと、処理プログラムが多少異なる。ただし、我々が今回設計、作成したシステムは実験的システムであるため、いくつかの点で、機能を縮小した。これは、本質的なことではなく、システム開発上の時間的問題を考慮したためである。

(1) KEY

KEY については、次のように構造を設けて、システムの中に組み込む。

今 A, B, …… , I の9つのKEY を組み込むとして、

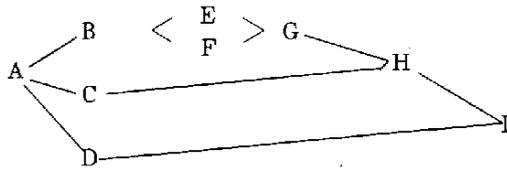


図 3 - 1 5

図 3 - 1 5 のようにする。コマンド列の先頭で KEY 名称を指定する。RECORD を定義する場合は、その RECORD はその KEY が割り当てられる。以後の削除、修正、データ値の設定の操作はこの KEY の使用者に限られる。参照については、図 3 - 1 5 の A の KEY の人は全てのデータを参照でき、B は B, E, F, G, H, I を参照できるというように、右側にある KEY によって定義された RECORD を参照できる。I によって定義された RECORD は全

ての人によって参照される。

RECORDについては、TYPE-I、TYPE-IIともに、FAMILYを削除した。これは後日拡張してもよいものである。

NOTEについては、固定的なものをいくつか、システムとして組み込み、使用者はそのNOTEについて選択するようにした。

(2) カード・ベース・システムにおける処理ツール

以上のようにして、次の処理ツールを機能としてもつことにする。

- (a) RECORDの定義
- (b) RECORDの修正
- (c) RECORDの削除
- (d) データの入力
- (e) テープとのデータの転送
- (f) レポート作成
- (g) 処理プログラムの実行
- (h) 演算、関数
- (i) システム管理上の情報の出力
- (j) 判断分岐
- (k) 実行順序の指定
- (l) MODELの定義
- (m) MODELの出力
- (n) MODELの削除
- (o) MODELの実行

以下、具体的に例示する。

① RECORDを定義するためには、それが、指標となるデータか、分析上の必要のためのデータかを区別する。後者の場合は、指標となるRECORDが既にシステムに対して定義されているかどうかを確認し、もしされていないければ、先に指標となるRECORDから定義する。

いま、ある製品の品種別、月別売上数量をデータの組として定義するものとする。この場合、品種と月次が指標となる。品種については、RECORD名をHINSHU、月次についてはNENGETとする。そして、そのRECORDが図3-16のようになっている。

HINSHU		NENGET	
NAME	CODE	NAME	CODE
DAI	1	46-2	1
CHU	2	46-3	2
SHO	3	⋮	⋮
		⋮	⋮
		49-1	36

図 3 - 1 6

この2つのRECORDについて、定義すると、

(a) DEFREC HINSHU, 1, 2, NAME, CODE, 3, 2,
3, 1

(b) DEFREC NENGET, 1, 2, NAME, CODE, 3, 2,
12, 1

となる。(a)の場合、工場名、RECORD TYPE(=1)、ITEMサイズ、各ITEM名称、各ITEMについてのデータ属性、最初がホレリストタイプ(4ケタ)次が整数、データサイズ、記録するNOTEのNO.(=1)というようになる。

次に、売上数量のRECORD名称がSALESとして、図3-17のような表で表わされるとする。

品 種 年 月	1	2	3
46-2			
46-3			
⋮			
49-1			

図 3 - 1 7

(c) DEFREC SALES, 2, 2, NENGET, HINSHU,
 36, 1

RECORD 名称, RECORD TYPE (=2), 次元のサイズ, 各次元に対応する RECORD の名称, データ属性, この場合, 倍精度整数, NOTE № (=1) となる。

② この RECORD に対応するデータが, カードに, このままの順番でパンチされていた場合, カードから入力すると,

(d) DTSET SALES / 10115 /

となる。次元の配列が逆にパンチされていると,

(e) DTSET ((SALES(I, J), J=1, 3), I=1, 36),
 / 10115 /

となる。

③ このデータを出力する場合は, FORMAT 文のないものと, あるものがある。先の例で,

(f) REPORT SALES

とすると, 図 3 - 1 8 のように出力される。

SALES

NENGET		HINSHU		
		DAI 1	CHU 2	SHO 3
46-2	1	11	13	13
46-3	2	⋮	⋮	⋮
		⋮	⋮	⋮
49-1	36	361	362	363

図 3-18

FORMAT文のあるものは、FORTRANと同様である。

④ このデータのある品種についての時系列データをEPA法によって分解するものとする。処理プログラムの実行に必要な入出力データは、各処理プログラムの引数として受け渡す。使用者がコマンドとして記述するのは、その対応する名称等である。入力データは配列と単一データ（これはパラメータとして用いることが多いはずである）で、出力は配列だけである。配列はRECORDとコマンドRES（コア上の配列宣言）によって定義されたものがある。この名称を指定する単一データは値を直接指定する。使用者は入出力データを示す情報（配列の名称か単一データかの値）をコマンドに記述する。このときに、類似したデータによって、同一処理プログラムを実行させたいことがあるはずである。それで、このような入出力データを示す情報それ自体をデータとして、RECORDに記録することにする。

このRECORDは全てTYPE-1 RECORDを使用することにし、使用者が定義する。このRECORDの横一列のデータが、処理プログラムの実行に際して必要な情報である。

入力データは、例えば、品種DAIについての時系列データとすると、こ

れを配列として処理プログラムの入力データを作らなくてはならない。もし、RECORDのデータ全てをデータとすると、この必要はない。出力データはRECORDとして記録しても、コマンドRESによって定義された配列変数に記録するよう指示してもよい。

(g) DEFREC INFEPA, 1, 5, DATA, KIKAN, N4, KJ,
OUTPUT, 13, 2, 2, 2, 13, 10, 1

(h) DEFREC EPAINDX, 1, 1, EPAINDX, 43, 5, 1

(i) COMP EPAINDX(1, 1) 'ORIGINAL'
EPAINDX(2, 1) 'SEASONAD'
EPAINDX(3, 1) 'TRENDCYC'
EPAINDX(4, 1) 'IRREGULER'
EPAINDX(5, 1) 'SEASONID'

(g)は、EPAのための入出力情報をデータとして記録するためのRECORDの定義である。(h)は、EPA法による出力データの指標を定義したもので(i)はその値を演算としてセットした。以上は、煩雑であるが、誰かが一度しておけばよいものである。

(j) COMP INFEPA(1, 1)='SHINSH'
INFEPA(1, 2)=36
INFEPA(1, 3)=12(月次の意)
INFEPA(1, 4)=1(乗法的分解)
INFEPA(1, 5)='EPASH'

(k) DEFREC EPASH, 2, NENGET, EPAINDX, 12, 1

(l) RES SHINSH, 36(データサイズ),
12(データ属性)

(m) RCOMP(I=1, 36) SHINSH(I)=SALES(I, 1)

(n) EXC EPA, INFEPA, 1

(j)は情報をデータとして、セットしたもので、(k)は、出力用RECORDを

定義し、(i)は入力用1次元データをセットするための配列を定義し、mは、データを移し替え、(n)は、処理プログラムEPAを実行させるものである。

3.14.5 MODEL

(1) 「MODEL」

別項(1)で説明したシステムは、端末装置を利用して、使用者からの情報を一つ一つ入力するという方法を採用した。これに対して、カード・ベース・システムは必要な情報を組にして入力する。このため、処理ツールの中に、「判断分岐」、「実行順序の指定」を機能としてもつコマンドがある。これにしたがって、使用者の指定する処理手順の表現方法も多少異なる。そこで、カード・ベース・システムにおいて、使用者の表現した処理手順をシステムが記録するとき、これをMODELと呼ぶことにする。

(2) 「MODEL」の表現形式

MODELとは、以下の条件をもつ、コマンド列である。

- ① コマンド列の中に、「定義された処理」に属するコマンドが含まれる。
- ② コマンド列の中に、「実行順序の指定」に属するコマンドがある場合、そのコマンドの示す添字のナンバー（実行の行先）はこのコマンド列のコマンドの個数以内である。

(3) 「MODEL」の管理

システムが、MODELの管理として提供する処理ツールは、次のとおり。

- (a) MODELを定義する。
 - (b) MODELを削除する。
 - (c) MODELの内容を出力する。
 - (d) MODELを実行する。
- ① MODELの定義は次のようにする。

OPERATIONとして、DEFMOD(define MODEL)と記述し、OPERANDにMODELの名称を指定する。次のカードから、MODELと

して定義するコマンド列を記述し、次に、OPERATION として、MEND と記述する。ここに指定されたコマンド列が、その MODEL として、システムに記録される。MODEL の定義、すなわち、DEFMOD から MEND までが、一つの「定義された処理」である。だから、1 回の実行（セクション）すなわち、KEY の指定から KEY までの間に、MODEL の定義を何度指定しようと、それ以外の処理を合わせて指定しようとかまわない。

MODEL の定義によって、システムはコマンド列そのものを記録するので、この定義の中に含まれているかもしれないエラーはほとんど発見できない。これは MODEL の実行まで待つほかない。

② MODEL の削除は、使用者によって定義された MODEL をシステムから取り除くことである。このコマンドの実行後は、この MODEL は実行できない。このコマンドは、OPERATION として、DELMOD (delete MODEL) OPERAND として、削除したい MODEL の名称を指定する。

③ MODEL の出力は、その MODEL の内容である、コマンド列および名称、定義した人の KEY をプリント・アウトすることである。すでに定義している MODEL について検討したいとき等のために用いる。

MODEL の出力は、OPERATION として、REPMOD (report MODEL), OPERAND として、その MODEL の名称を指定する。

④ MODEL の実行は、その MODEL を表わすコマンド列を実行することである。MODEL の実行は、OPERATION として、EXCMOD (execute MODEL), OPERAND として、実行したい MODEL の名称を指定する。

MODEL の定義 (DEFMOD) の中のコマンド列の中に、MODEL の実行 (EXCMOD) を示すコマンドが含まれることはかまわない。使用者は、各部門または個別問題ごとに MODEL を作り、それらを結合して、別の MODEL を作ることができる。

(4) 「MODEL」の融通性

MODEL として定義する処理手順は、すぐ実行できるという意味で、形式的

には完全に固定している。しかし、実際の処理内容は、以下のようにして、融通性をもたせることができる。

- ① 演算，関数，処理プログラム，レポート等の処理で対象になる RECORD，配列変数，変数は，その名称が指示されるだけであるから，その RECORD なり配列変数，変数にセットしておくデータ（値）は，MODEL の実行以前にいろいろ変えることができる。ある MODEL の実行（EXCMOD）のコマンドの前に，その MODEL の中で用いられるデータをセットするコマンド（演算，入出力等）を指定して，それらを組にして，コマンド列を作ればよい。
- ② 処理プログラムの実行に際して，使用されるデータは，そのデータを記録している RECORD，配列変数，変数の名称を指定することにより定まり，処理結果であるデータも同様である。そして，処理プログラムに対する入出力データを示す RECRD 等の名称それ自体もデータとして，ある RECORD に記録し，処理プログラムの実行については，その RECORD の名称を指定する。だから，MODEL の実行に先だって，その入力情報 RECORD のデータ（すなわち，実際の入出力対象になる RECORD 等の名称）を書き替えると，処理プログラムで使用するデータ（値）は変わるし，処理結果において，あるデータの記録される RECORD，配列変数等も変わる。
- ③ ある処理プログラムの実行についての入力情報は，ある入力情報 RECORD の横一列を参照する。その横一列のデータの位置は，コマンドの中で，数字または変数で指定する。

```
EXC   EPA, IEPA,  3
```

```
EXC   OLS, IO LS,  I
```

変数の値は，別に設定されるものであるから，MODEL の実行を指示するコマンドに先だって，その変数の値を決めることにより，処理プログラムに使用されるデータが規定される。こうして処理内容が多様になる。

(例)

DEFMOD	JUYO
EXC	EPA, IEPA, I
EXC	OLS, IOLS, J
MEND	

	I = 3
	J = 2

EXCMOD	JUYO
	I = 2
	J = 3

EXCMOD	JUYO
--------	------

3.1.4.6 システム構造

(1) 条件

システム構造，すなわち，サブプログラムの設定仕様，必要な情報の記録の仕方を，次の点を考慮して決める。

- ① サブプログラム相互の独立性，すなわち，あるサブプログラムの修正，削除，追加を，他を考慮せずにできるようにする。
- ② 機能の変更とサブプログラムの変更との対応，すなわち，ある処理機能の修正，削除，追加を，できるだけ少ない（ほとんどの場合は一つの）サブプログラムの変更によってできるようにする。
- ③ 単純な構造，すなわちプログラムの構成が論理的に単純な構造になっていて，システムの維持（maintenance）が容易にできるようにする。
- ④ 使用者または使用者に近い人（システム管理者等）が変更する必要の生じる部分，つまり，処理プログラムの修正，削除，追加，および記録領域（ディスク上の）の拡張については，システム全体の構成とは関係なく，その部分についてだけ，変更すればよいようにする。特に処理プログラムは，

使用者自身によって、問題意識、環境の変化、企業自身の発展等に応じて、漸次追加されていくという事態を期待している。それで、この部分については、処理プログラムそれ自体の作成以外については、ほとんど何も変更しなくてもよいようにする。

(2) サブプログラム

次のようにサブプログラムを作成する。

① メインプログラム

- (a) コマンドを読み込み、記録する。
- (b) コマンドについて、処理ツールの別を解析し、その処理ツールに応じて必要なサブプログラムに実行を移す。
- (c) 終了を示すコマンドと、エラーの発生した場合は、実行を終了する。
- (d) コマンド列の先頭で指定されるKEY について、それが登録されているものかどうかをチェックし、そうであればそれを記録し、登録されてなければエラーの発生に伴う処理をする。

② グループ-1 サブプログラム

グループ-1 サブプログラムは、各コマンドに対して作る。

- (a) DEFREC (define RECORD)
- (b) DELREC (delete RECORD)
- (c) UPDREC (update RECORD)
- (d) DTSET (data set = card read)
- (e) DMOVE (data move = tape read/write)
- (f) REPORT (report writer)
- (g) EXC (execute process program)
- (h) COMP (computation)
- (i) MANAGE
- (j) BRANCH
- (k) JUMP

- (l) DEFMOD (define MODEL)
- (m) DEPMOD (report MODEL)
- (n) DELMOD (delete MODEL)
- (o) EXCMOD (execute MODEL)

③ グループ-2 サブプログラム

グループ-2 サブプログラムは、処理プログラムである。前述のとおり、処理プログラムは、分析手法となるべく分離して、一つの処理プログラムとして作成することが望ましい。また、処理プログラムは使用者自身によって作成してもかまわない。問題オリエンテッドな内容をもつプログラムは使用者自身が作成することが望ましい。処理プログラムとして、システムに組み込んでおくべきものを決めるのは、各企業のその時々の状態に依存する。

以上のことを前提とした上で、我々が現時点として、システムに組み込んだ処理プログラムを示す。これは単なる例にすぎない。

- (a) EPA (EAP法による時系列分解)
- (b) OLS (最小二乗法による回帰分析)
- (c) TSLS (連立方程式モデルの二段階最小二乗法による回帰分析)
- (d) TEST (トータル・テストによるシミュレーション)
- (e) SEIHIN (製品需給を分析するシミュレーション。製品の品種別の需要量と各工場別の生産能力、工場内、工場外の在庫能力等々の関係を月次に調べる。)(2)
- (f) ZAIMU (原価計算、部門費集計等を積み重ねて、見積り財務諸表を作成する。)

④ グループ-3 サブプログラム

グループ-3 サブプログラムは演算式の中に組み入れて記述できる関数である。関数も処理プログラムと同様に、使用者のニーズによって作成されるものである。ここで示すのは例にすぎないが、しかし、さまざまな経営計画に関する情報処理において共通に使われるものと考えてよい。(3)

- (a) SIGMA (Σ 演算) [4]
- (b) MAX (最大値をとり出す)
- (c) MIN (最小値をとり出す)
- (d) LOGE (自然対数)
- (e) LOGD (常用対数)

⑤ グループ-4 サブプログラム

グループ-4 サブプログラムはメインプログラムとグループ-1 サブプログラムが共通に使用するサブプログラムである。このサブプログラムは次のものからなる。

- (a) 記録領域へのデータの書き込みと記録領域からのデータの読み込み。
- (b) コマンドの文字列の解析。
- (c) データ属性、すなわち ホレリスタイプからの整数、実数タイプへの変換。

⑥ グループ-5 サブプログラム

グループ-5 サブプログラムは処理プログラムによって使用されるサブプログラムである。これらのサブプログラムは、処理プログラムの変更によって、変更することになる。これらのサブプログラムは、使用者によって作成されるものとする。

3.1.4.7 記録する情報

(1) 記録する情報

記録する情報は、使用者によって定義されたRECORDのスキーマ、データそのもの、同様に定義されたMODELの内容を示す情報、使用者のKEY (KEY の名称) である。

(2) 記録する情報の論理的構造

このシステムはKEYの個数、名称をシステム作成時にプログラムとして組み入れたので、KEYの個数は固定している。

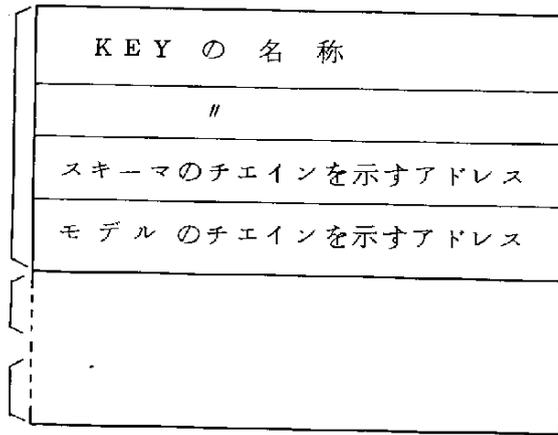


図 3-19

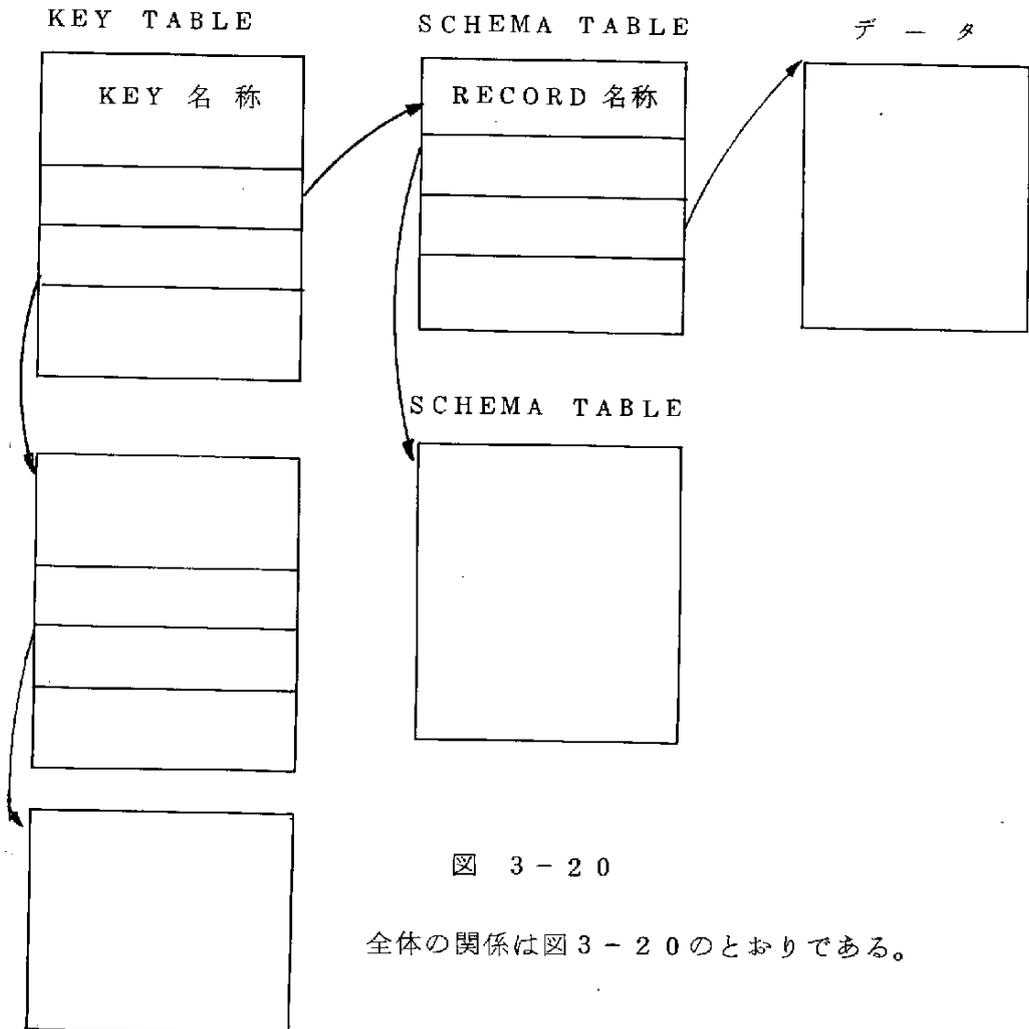
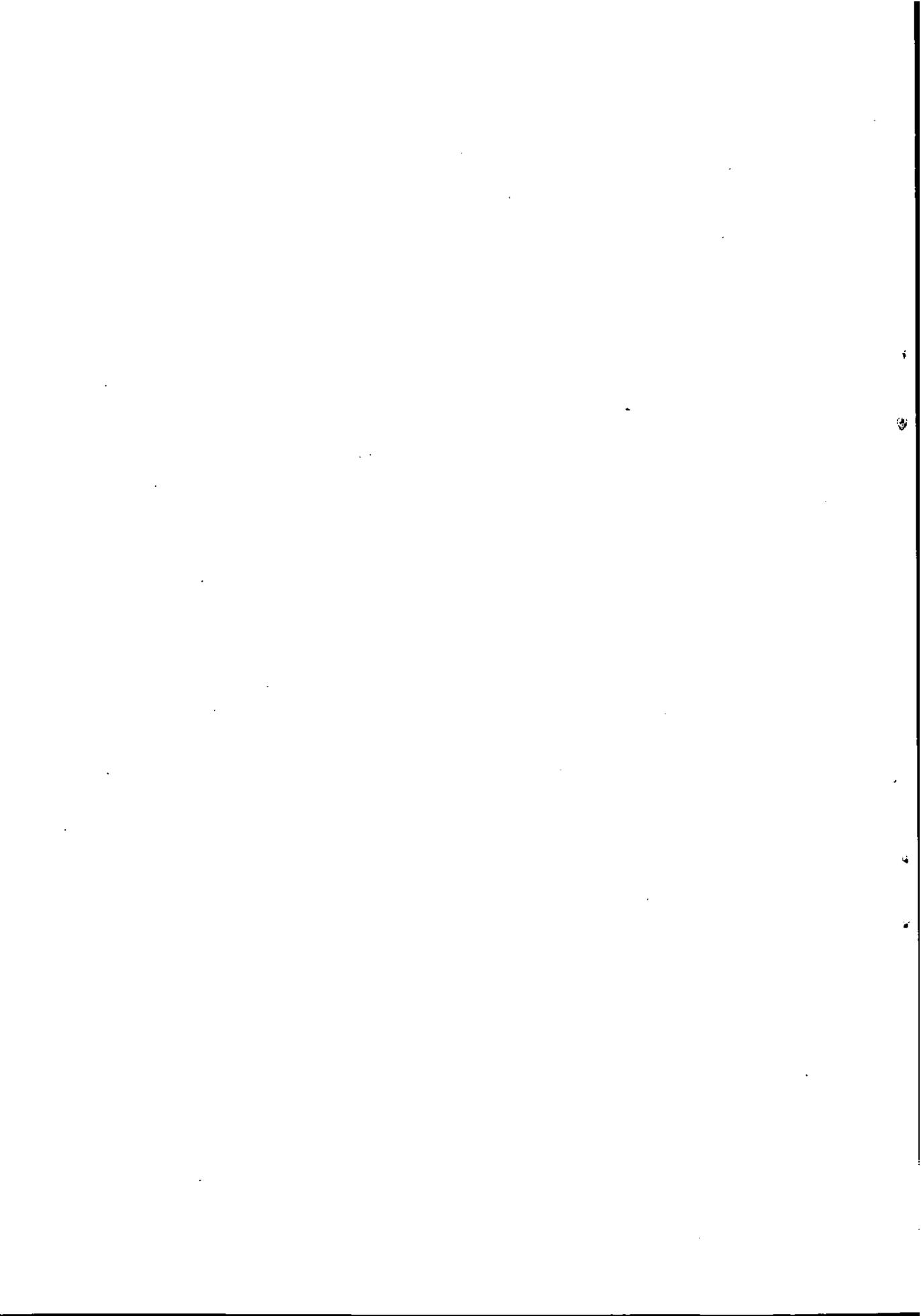
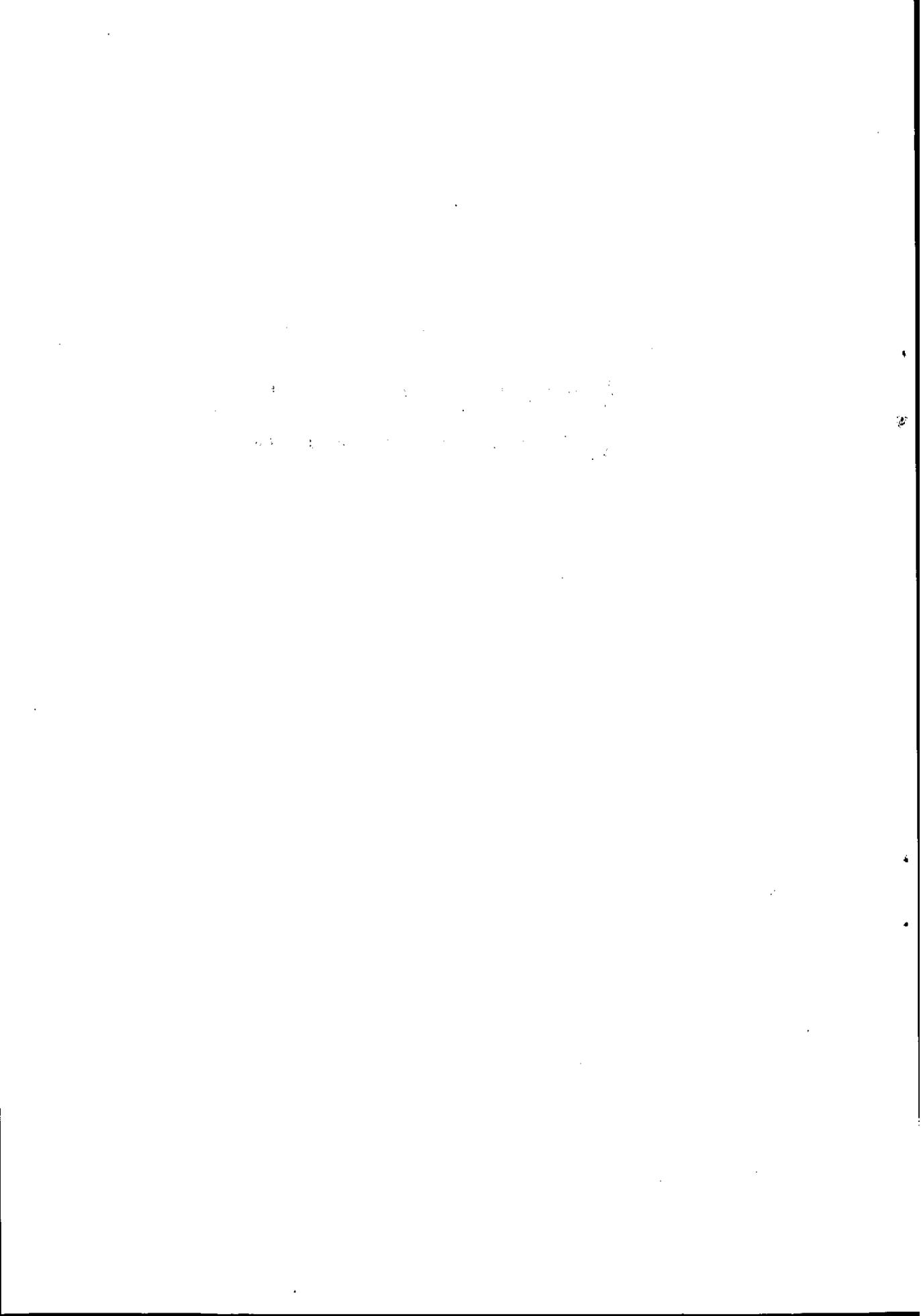


図 3-20

全体の関係は図3-20のとおりである。



4. 経営計画情報システム
に関するケーススタディ



4. 経営計画情報システムに関するケーススタディ

4.1 経営計画情報システム形成に関するアンケート調査

はじめに

経営計画プロセスにおけるコンピュータ利用の現状について、その利用形態利用体制、利用上の問題点などを知ることは今後の研究にとって大いに参考となると思われる。また同時にMDSにおけるわれわれの意図を理解して貰い、いろいろなコメントを得ることも有用であろう。そこでわれわれは今回のプロジェクトの一環として付表に示した内容からなる上記の標題のアンケート調査を実施した。本章はその結果についてのとりまとめを目的としている。

今回のアンケート調査の対象となったのは、当協会が昭和50年度のプロジェクトの一つとして実施した「経営情報調査(VII)」において行われた「経営科学手法の導入活用に関する調査」に回答を寄せた193社の企業であり、3月下旬の期限までに55社(回収率28.5%)からの回答を得ることができた。これらの企業はおそらくこの種の調査に多大の関心をもっているものと思われるし、またコンピュータ利用についてもかなりの水準に達しているものと予想される。このような対象企業のかたよりや、回答企業数などを考えると、アンケートの結果から一般的に当てはまるよう結論を得ようとするような詳細な分析を行うことは無理のようなので、ここでは一応の傾向といったものを読みとる程度にとどめることにする。なお次年度においても同種のアンケート調査を試みる予定もあるので、その時点ではもう少し綿密な検討も有効になってこよう。

4.1.1 経営計画システムの構成

各社の経営計画システムがどのような個別計画から成っているかを聞いた結

果が表4-1である。さすがに生産、販売、財務といったところはほとんどすべての企業において計画策定が行われているが、研究開発計画は意外に少ないようである。

表4-1 計画システムの構成 (問1)

個 別 計 画	件 数	% (55)
1. 販 売 計 画	52	95
2. 生 産 計 画	50	91
3. 設 備 計 画	46	84
4. 原 材 料 調 達 計 画	42	76
5. 研 究 開 発 計 画	35	64
6. 人 員 計 画	45	82
7. 財 務 計 画	49	89
8. そ の 他 の 計 画 在庫, 物流, 外注, 利益, 営業, 全社長期マクロ, プロジェクト管理, 環境 アセスメント, 投融資, 部門費, 関連事業, 教育訓練, 原料乳配分		

次にこれらの個別計画は全体(全社)計画とどのように関連づけられているだろうか。その結果をみると(表4-2)ほとんどの場合において何らかの形で個別計画の結果が全体計画に反映されていることがわかる(91%)。個別計画の間が完全に連動する形で全体計画につながっている例は6社にすぎないが、一部の個別計画の間で連動している企業も含めればその割合は約半数になる。個別計画の間の連動の仕方は表3にみるようにコンピュータを用いている企業と手作業による企業とが半々である。企業数が少ないので何ともいえないが、個別計画の調整プロセスにおいてコンピュータ・モデルの利用が有効であると思われる。

表 4 - 2 個別計画相互および全体計画との関係 (問 2)

個別計画と全体計画の関係	件数	%	4.の内容
1. 完全に連動	6	11	販売 3 生産 3
2. 一部連動	18	33	設備 3
3. 全体計画の中で調整	22	40	原材料調達 2
4. 個別計画の一部を用いる	4	7	人員 1 財務 1
5. 全体計画を別途に策定	4	7	その他 1

表 4 - 3 個別計画の連動方式 (問 2 - 1)

個別計画の連動	件数
コンピュータ・モデルによる	2
手作業による	2
その他	1

全体計画の中に個別計画の結果をとり入れる際に個別計画の数値を調整する必要がある企業が多いが、その理由は個別計画の前提条件の調整や計画期間の違いによる調整の必要などといったことにあるようである。個別計画を全体計画の期間に合わせて整理したり、新しい条件のもとで計画を立て直してみるため

表 4 - 4 個別計画の間で調整が必要な理由 (問 2 - 2)

調整の理由	件数	%
1. 期間の調整	14	35
2. 前提条件の調整	17	43
3. 連動は不可能	6	15
4. 将来は連動させる予定	6	15

にコンピュータの処理能力が有効となると思われる。

個別計画とは別途に全体計画を策定している企業においては、現在のところそのほとんどが手作業ないし部分的なコンピュータ利用によって行っており、コンピュータ・モデルを利用している企業はわずかに1社にすぎない。

最後に全体計画の担当部門であるが、その多くは企画部ないしそれに類するものと経理部である。

表4-5 全体計画策定担当部門 (問2-4)

企画部, 企画室, 企画課	11(社)	経営計画部	1(社)
総合企画(本)部, 室	7	管理経理本部	1
経理部	6	事業本部	1
管理部(調査グループ)	6	商 務 部	1
社長室	4	総 括 部	1
社長室 企画部	1	経 理 統 括 室	1
〃 経理部	1	経 営 会 議	1
〃 コンピュータ室	1	経営計画策定委員会	1
生産, 工場 管理室	1	担 当 部 門	2
技 術 部	1		
生 産 部	1	N. A.	8
経 営 管 理 部, 室	2		
経 営 企 画 室	1		
企 画 情 報 室	1		

4.1.2 個別計画におけるコンピュータ利用

個別計画におけるコンピュータ利用の状況は表4-6の通りであり、販売、生産、原材料調達、財務計画などにおける利用が多い。研究開発、設備計画などはコンピュータを利用している企業は少ないようであるが、これはコンピュータ利用が進むにつれてこれらの計画にも利用が及ぶのか、それとも計画の性

表4-6 個別計画におけるコンピュータ利用(問3)

	コンピュータ利用		コンピュータ利用の形態				使用されている手法 数									使用プログラムの種 類		計 画 の 期 間										計 画 の サ イ ク ル							
	有	無	バッチ	オンラインバッチ	社内TSS	商用TSS	1	2	3	4	5	6	7	8	9	P	U	半年未満	半年	1年	2年	3年	4年	5年以上	その他	1ヶ月と	3ヶ月と	半年と	1年と	2年と	3年と	5年と	その他		
1. 販売計画	40	11	40	2	1	2	21	7	11	3	2		3	21		14	34	9	14	12	6	13		1	7	2		6	9	28	10	2	1	1	4
	73	20	(40) 100	5	3	5	(35) 60	20	31	9	6		9	60		(40) 35	(48) 85	19	29	25	13	27		2	15	4	(48) 13	19	58	21	4	2	2	8	
2. 生産計画	40	8	40	4	1	2	4	3	1		14	2	6	18	2	13	35	14	16	14	4	10			8	1	12	13	23	11	1		1	3	
	71	15	(40) 100	10	3	5	(34) 12	9	3		41	6	18	53	6	33	88	30	34	30	9	21		17	2	(46) 26	28	50	24	2		2	7		
3. 設備計画	20	25	19	2		1	2	1	1		3		4	9	2	6	16	1	6	17	4	7		1	8	3	2	2	17	14			1	4	
	36	45	(20) 95	10		5	(25) 8	4	4		12		16	36	8	(20) 30	(35) 80	3	17	49	11	20		3	23	9	(33) 6	6	52	42			3	12	
4. 原材料調達計画	30	11	28	3	1		4	1	1		5		3	15	4	8	27	12	12	13	3	3		3	2	11	9	16	8					2	
	55	20	(30) 93	10	3		(25) 16	4	4		20		12	45	16	(30) 27	(38) 90	3	32	32	34	8	8		8	5	(36) 31	25	44	22				6	
5. 研究開発計画	9	28	9	2	2		3	2	1		1	1	1	9		2	8	2	6	7	3	4		3	3	2		11	7	1		1	4		
	16	51	(9) 100	22	22		(11) 27	18	9		9	9	9	82		(9) 22	(23) 89	9	26	30	13	17		13	13	(22) 9		50	32	5		5	18		
6. 人員計画	22	22	22	1	1		2				2		5	14	1	3	19	2	7	8	2	7			8	1	3	1	15	13	1		1	3	
	40	40	(22) 100	5	5		(19) 11				11		26	74	5	(22) 14	(32) 86	6	22	25	6	22		25	3	(32) 9	3	47	41	3		3	9		
7. 財務計画	29	15	28	3	3	1	5	4	1			11	20		5	26	5	11	13	3	7		1	8	1	6	6	22	8	1	1	1	2		
	53	27	(29) 97	10	10	3	(28) 18	14	4			39	71		(29) 17	(37) 90	14	30	35	8	19		3	22	3	(38) 16	16	58	21	3	3	3	5		
8. その他	11	4	9	1		1			1	2	1	4	5	2	1	11			6	5		3		4	1	3	2	6	4				3		
	20	7	(11) 82	9		9	(11)			9	18	9	36	46	18	(11) 9	100		50	42		25		33	6	(12) 25	17	50	33				25		

()内数字は回答件数 上段:件数 下段:％ 「コンピュータ利用の有無」については回答件数55とした。

- 兼 手法 1. 時系列分析(指数平滑法, EPAなど) 5. 数理計画法(LPなど) 9. その他
 2. 多変量解析法(因子分析, 相関分析など) 6. ネットワーク理論(PERTなど)
 3. 計量経済分析手法(回帰分析, 連立モデルなど) 7. シミュレーション分析
 4. 産業連関分析 8. 積み上げ方式

質上利用するメリットがないのかは判らない。しかし全般的にいつてかなりの程度計画業務においてもコンピュータが利用されるようになってきている様子がうかがえる。

コンピュータの利用は計画策定の頻度とも関連があると思われるが、この点について計画のサイクルをみると、研究開発計画や人員計画は大半が半年ないしそれ以上と答えているのに対して、生産、販売などは半年以下が多い。また計画の対象期間をみると各計画ともかなりのバラツキがみられ、傾向ははっきりしない。

次にコンピュータ利用の形態をみると、現在のところ、バッチ処理ないしオンライン・バッチが大半で、TSSを利用している例は少ない。また計画策定において使用している手法は積み上げ方式、時系列分析が多い。個別には生産計画における数理計画法、設備計画、人員計画、財務計画におけるシミュレーション、販売計画における計量経済分析が目につく。使用プログラムは使用者が作成する場合が多いようである。プログラムの開発状況は表4-7にも見ることができるが、これからも内部での、しかもコンピュータ部門の全面的な協

表4-7 計画用プログラムの開発 (問4-1)

計画用プログラムの開発体制	件数	% (55)
1. 外部に依頼	3	6
2. コンピュータ部門	32	58
3. 計画部門とコンピュータ部門の共同	18	33
4. 計画担当部門	1	2

力のもとでの開発がほとんどであることがわかる。したがって現在のところコンピュータ部門の協力なしには計画用の新しいプログラムを開発することは困難であり、それが計画業務におけるコンピュータ利用の大きな制約になっているようである。コンピュータ部門の協力は既存のプログラム利用に際しても不

可欠で、計画担当者のみで計算処理を行っている企業は10%強にすぎない。

表4-8 コンピュータの利用方法 (問4-2)

コンピュータの利用方法	件数	%
1. コンピュータ部門にすべてまかせる	10	18
2. データのみ準備	34	62
3. 計画担当者がほとんど処理する	7	13
4. その他	1	2

計画策定におけるコンピュータ利用上の問題については、すでにみたように、プログラム開発の時間、人力的な制約をあげる企業がもっとも多く、この面でのサポートをねらいとするMDSの意義が裏づけられた結果ともなっている。また、計画用モデルに関する問題を指摘する企業もかなりあることも注目される。この点はORをはじめとする計画用モデルの開発、普及に依存するところが大きい。計画モデルの欠点についてアメリカの調査結果を参考に示しておく。表4-10のような結果が出ている。そのいくつかの点はコンピュータの利用方法を工夫することによって改善することが可能と思われる。計画策定における問題点として、コンピュータ・コストがほとんどとりあげられていないが、これはまだそれほど厳密な費用計算が行われていないためであろう。

4.1.3 計画データの維持、管理

計画策定のために必要なデータの収集、維持はそれ自体一つの大きな問題である。まず各部門で発生するデータの管理については部門ごとに維持している企業が多いが、共通利用が可能なような集中管理を行っている企業の例もある。

表4-9 計画策定におけるコンピュータ利用上の問題点(問6)

問 題 点	件 数	%
1. 問題ごとに各々プログラムを作成しなければならないので時間的, 人的にコンピュータ利用が制約される。	23	42
2. 専門のプログラマー(システム分析者)に頼めるが, 必要な処理内容を十分に理解してもらうのが大変な仕事である。	15	27
3. プログラム作成に時間がかかるので, 継続的に利用が見込まれる処理プログラムが中心で, そのときどきに必要が生じた処理は手作業で行うことが多い。	36	65
4. すでにあるプログラムを利用する場合でも, 処理の必要を感じてから結果が得られるまでの時間(Turn around time)が長くかかる。	13	24
5. コンピュータ利用に際し, 利用者側に余りにも多く利用上の知識, 作業が要求されるので利用しにくい。	5	9
6. コンピュータの使用コストが利用部門に賦課され, その額が大きい。	1	2
7. コンピュータを使って計画を策定しても表現形式が数値データ中心であるため, トップマネジメントの利用にとって難解である。	11	20
8. 経営計画策定用モデルに適当なものがない。	15	27
9. その他(具体的に御記入下さい。)	7	13

表4-10 企業モデルの欠点 [出所: T. H. Naylor et.,
Corporate simulation
models, SIMULATION,
June, '75.
企業モデルについてのアンケート調
査(346社)の結果である]

欠 点	百 分 率
柔軟性に欠ける	25%
ドキュメントが貧弱	23
入力データが多過ぎる	23
出力様式に柔軟性がない	11
開発期間が長すぎる	11
運用費用が高すぎる	9
欠点なし	9
開発費用がかかりすぎる	8
モデルのユーザがモデル開発を理解しない	8
分析過程が理解できない	5
アウトプットが詳細でない	4
ユーザ向きでない	3
結果が明らかに不正確	3
アウトプットが詳しすぎる	3
意図したことをモデル化していない	2

表4-11 データの管理体制 (問5-1)

データの管理体制	件 数	%
1. ほとんどのデータが共通に利用される	6	11
2. 全社的に扱うデータについて集中管理	12	22
3. 部門で維持・管理	34	62

次に業務上発生するデータと計画策定に用いられるデータとの関連であるが、表4-12をみると大半の企業が業務データを何らかの形で参照しており、またそれとともに計画用の独自のデータを作成している企業も多い。別途に作成する計画用のデータのみを用いる例は数社にすぎない。

表4-12 業務データと計画用データの関係 (問5-2)

業務データと計画用データの関係	件数	% (55)
1. 業務データの大部分を参照	10	18
2. 業務データを加工して利用	13	24
3. 2に加えて計画用の独自のデータも利用	27	49
4. 大部分のデータは別途作成される	4	7

データの維持および表現形式については表4-13および表4-14のような結果が得られている。これから全社的ないしは部分的にでも表現形式を統一している企業が60%をこえ、部門ごとにバラバラな表現形式をとっている企業数を上回っている。コンピュータにおけるデータの記録形式は関連するプログラムの間での統一を考えている企業が多く、データの集中管理システムを採用している企業の例はまだ少ない。

表4-13 データ表現形式の統一 (問5-3)

統一表現データの採用規模	件数	%
1. 全社的に統一した表現形式	11	20
2. 部分的に統一した表現形式	24	44
3. 表現形式は統一されていない	18	33
4. その他	1	2

表4-14 記録媒体上におけるデータの表現形式(問5-4)

記録媒体上におけるデータ表現形式	件数	%
1. 情報管理システムによる集中管理	3	6
2. 共通利用が可能な形式を採用	9	16
3. 部分的に統一	25	45
4. プログラムごとに個別の表現形式を用いる	12	22
5. その他	2	4

最後に、経営計画に対するサポートをねらいとするMDSアプローチについて各社の感想を聞いてみた結果は表4-15のとおりであり、80%の企業がMDSに関心をもっていたり、あるいはすでに類似のシステムを計画ないしは導入していることが分り、この種のシステムに対する関心の高さをみることができる。

表4-15 MDSに対する関心(問7)

MDSに対する関心度	件数	% (55)
1. MDSに関心がある	25	45
2. MDSと同様のシステムを導入ないしは計画している	11	20
3. 関心がない。その場合	11	20
A. 現在のシステムで十分である	A 4	
B. MDSとは別のシステム構想をもっている	B 6	

「経営計画情報システム形成に関するアンケート調査票」

- 本調査についてのお問い合わせ先 -

(財)日本情報処理開発センター 調査課

電話 東京 03-434-8211 内線 539

〒105 東京都港区芝公園3-5-8 機械振興会館内

問1 貴社の経営計画システムは次のどのサブシステム(個別計画)から構成されていますか。回答用紙(別添)の該当する項目の番号に○印を御記入下さい。

1. 販売計画(需要予測をも含む)
2. 生産計画
3. 設備計画
4. 原材料調達計画
5. 研究開発計画
6. 人員計画
7. 財務計画
8. その他計画(具体的な名称を御記入下さい。)

問2 貴社の上記サブシステムと全体的(全社的)経営計画システムとの関連はどのようになっていますか。該当する項目の番号を回答用紙に御記入下さい。

1. 各計画サブシステムが完全に連動する形で全体計画システムが構成されており、サブシステムにおいて策定された個別計画はそのまま全体計画システムに用いられる。
2. 全体計画サブシステムにおいて一部のサブシステムは連動しているが、他のサブシステムとの間では、調整を行なっている。
3. 各計画サブシステムにおいて策定された計画を用いて全体的計画が策定さ

れるが、その際にはサブシステムの間でかなりの調整を行なっている。

4. 全体的経営計画システムは上記計画の一部（例えば販売，生産，財務等）をもちいて構成されている。その場合構成されている計画の番号または名称を全て記入して下さい。

5. 個別の計画サブシステムとは別途に全体的経営計画が策定されている。

問2-1 問2で1とお答えの場合、全体計画策定の方法について該当する項目の番号を回答用紙に記入下さい。

1. コンピュータ・モデルによって計画サブシステムを連動させている。
2. 各計画サブシステムから出された計画を調整を加えないで手作業によって全体計画にまとめあげている。
3. その他（具体的に御記入下さい。）

問2-2 問2で2または3とお答えの場合、調整を必要とする理由について該当する項目の番号を回答用紙に御記入下さい。

1. 計画サブシステムによって計画期間が異なるので全体計画の期間に合わせて調整する必要がある。
2. 計画の前提条件などについて各サブシステム担当部門間で意見をかわし計画を調整する必要がある。
3. 各サブシステムはそれぞれ性格が異なっているので連動させられないし、その必要もない。
4. 現在は手作業で調整しているが将来はコンピュータ・モデルを利用して連動させるべく検討している。
5. その他（具体的に御記入下さい。）

問2-3 問2で4または5とお答えの場合、全体計画システムの概要について該当する

項目の番号を回答用紙に御記入下さい。

1. 全体計画策定のためのコンピュータ・モデルを利用している。
2. 個別のサブシステムとは別の方式、スケジュールにもとづき手作業によって全体計画を策定している。(部分的なコンピュータ利用をも含む)
3. その他(具体的に御記入下さい。)

--

問 2-4 全体的(全社的)経営計画策定を担当される部門の名称を回答用紙に御記入下さい。

--

問 3. 問 1 でお答えいただいた計画サブシステムにおいてコンピュータは、どのように利用されておりますか。下記記入要領および記入例を参考にして該当する項目に御記入下さい。御回答は、回答用紙を御利用下さい。

計画サブシステム名称	コンピュータ利用の有無	コンピュータ利用の形態	使用されている主要な手法	使用されているプログラムの種類	計画の期間	計画のサイクル (作成間隔)
(例) 販売計画	有, 無	B, O, T1, T2 その他 ()		P, U	1, 2, 3, 4 5, 6, 7, 8	1, 2, 3, 4 5, 6, 7, 8
1. 販売計画	有, 無	B, O, T1, T2 その他 ()		P, U	1, 2, 3, 4 5, 6, 7, 8	1, 2, 3, 4 5, 6, 7, 8
2. 生産計画	有, 無	B, O, T1, T2 その他 ()		P, U	1, 2, 3, 4 5, 6, 7, 8	1, 2, 3, 4 5, 6, 7, 8
3. 設備計画	有, 無	B, O, T1, T2 その他 ()		P, U	1, 2, 3, 4 5, 6, 7, 8	1, 2, 3, 4 5, 6, 7, 8
4. 原材料調達計画	有, 無	B, O, T1, T2 その他 ()		P, U	1, 2, 3, 4 5, 6, 7, 8	1, 2, 3, 4 5, 6, 7, 8
5. 研究開発計画	有, 無	B, O, T1, T2 その他 ()		P, U	1, 2, 3, 4 5, 6, 7, 8	1, 2, 3, 4 5, 6, 7, 8
6. 人員計画	有, 無	B, O, T1, T2 その他 ()		P, U	1, 2, 3, 4 5, 6, 7, 8	1, 2, 3, 4 5, 6, 7, 8
7. 財務計画	有, 無	B, O, T1, T2 その他 ()		P, U	1, 2, 3, 4 5, 6, 7, 8	1, 2, 3, 4 5, 6, 7, 8
8. その他 ()	有, 無	B, O, T1, T2 その他 ()		P, U	1, 2, 3, 4 5, 6, 7, 8	1, 2, 3, 4 5, 6, 7, 8
()	有, 無	B, O, T1, T2 その他 ()		P, U	1, 2, 3, 4 5, 6, 7, 8	1, 2, 3, 4 5, 6, 7, 8

〔記入要領〕

① コンピュータ利用の有無

有無いずれかに○印を御記入下さい。③の場合には②、④については回答する必要はありません。

② コンピュータの利用形態

下記のいずれかに○印あるいは名称を御記入下さい。

B : バッチ処理

O : オンラインバッチ処理

T1 : タイムシェアリング処理 (社内システムを利用している場合)

T2 : " (商用 TSS (DEMOS, MARKII など)
を利用している場合)

その他の場合は具体的に御記入下さい。

③ 使用されている手法

下記のいずれかの番号あるいは名称を御記入下さい。利用なさっているもの全てについてお答え下さい。

1. 時系列分析 (指数平滑法, EPA など)
2. 多変量解析法 (因子分析, 相関分析 など)
3. 計量経済分析手法 (回帰分析, 連立モデル など)
4. 産業連関分析
5. 数理計画法 (LP など)
6. ネットワーク理論 (PERT など)
7. シミュレーション分析
8. 積み上げ方式
9. その他 (具体的に御記入下さい。)

④ プログラムの種類

P または U いずれか該当するものに○印を御記入下さい。

P : アプリケーションパッケージプログラムまたは外部作成プログラム

U : 利用者が独自に開発したプログラム

⑤ 計画の期間

下記のいずれか該当するものの番号に○印を御記入下さい。

1. 半年未満
2. 半年
3. 1年
4. 2年
5. 3年
6. 4年
7. 5年以上
8. その他

⑥ 計画の策定サイクル（作成間隔）

下記のいずれかの該当する番号に○印を御記入下さい。

1. 1ヶ月ごと
2. 3ヶ月ごと
3. 半年ごと
4. 1ヶ年ごと
5. 2ヶ年ごと
6. 3ヶ年ごと
7. 5ヶ年ごと
8. その他

問 4. 経営計画策定におけるコンピュータ利用について該当する項目の番号を回答用紙に御記入下さい。

問 4-1 プログラムの開発は主にどの部門が担当されておりますか。

1. 外部のソフトウェア会社に依頼する。(外部パッケージ活用も含む)
2. コンピュータ部門がおこなう
3. 計画担当部門とコンピュータ部門とが共同しておこなう
4. 計画担当部門がおこなう
5. その他(具体的に御記入下さい)

--

問 4-2 コンピュータの利用方法は次のどのように行なっておりますか。

1. 処理プログラム名および使用データ名を指示する。その他はコンピュータ部門にまかせる。
2. 使用データは準備するがあとは全てまかせる。
3. 使用データ、コンピュータ制御データまで利用者が準備する。
4. その他

--

問 5. 経営計画策定過程で取り扱うデータの維持、管理はどのようになされておりますか。該当する項目の番号を回答用紙に御記入下さい。

問 5-1 貴社で経営計画策定に関係している各部門についてある部門で取り扱われるデータは、他の部門ではどのように参照されておりますか。

1. 社内で取り扱うデータは、ほとんど、部門とはかかわりなく共通に利用されている。
2. 全社的に取り扱うデータがあり、それについては各部門共通の資料管理者(部門)がある。
3. データは各部門ごとに維持、管理されているが他部門からの参照、問い合わせもかなり頻繁に行なわれている。

4. その他（具体的に御記入下さい）

--

問5-2 経営計画策定において利用されているデータは他の業務上で取り扱われているデータとどのように関連しておりますか。

1. 業務上の処理過程で取り扱われるデータは大部分計画過程でも参照される。
2. 業務上で取り扱われるデータを処理、加工、集積して計画策定に利用されている。計画過程で取り扱われるデータの大部分を占めている。
3. 2のようにして作成されるデータと計画策定で独自に作成するデータも利用する。
4. 計画策定過程では大部分が業務上で取り扱われるデータとは別のものである。
5. その他（具体的に御記入下さい。）

--

問5-3 データは何らかの形で資料として記録され利用に便利なように工夫されていると思いますが、データの表現形式について貴社ではどのような工夫を行なっておりますか。

1. 全社的にある統一した表現形式を採用し、ほとんどがそれに従って記録されている。
2. 部門別あるいはデータの内容によりいくつかの表現形式を採用し、各々それに従って記録されている。
3. データの表現形式については各自（各部門）が、その都度適した表現形式で記録する。
4. その他（具体的に御記入下さい。）

--

問5-4 貴社で取り扱いデータの全部または一部について、コンピュータの周辺機器を記録媒体として利用されている場合、そのデータについて表現形式をどのように工夫されておりますか。

1. 情報管理システム（データベース管理システム）を導入し、データの管理は統一して行なっている。
2. 情報管理システムは導入してはいないが、全社的に全ての処理に利用可能なように統一した表現形式を採用している。
3. 関連する処理（プログラム）をグルーピングして、その間では統一した表現形式を採用している。
4. 各処理（プログラム）に応じて別個に表現形式をもちいている。
5. その他（具体的に御記入下さい）

問6. 経営計画策定におけるコンピュータ利用上の問題点について該当する項目の番号を回答用紙に御記入下さい。

1. 問題ごとに各々プログラムを作成しなければならないので時間的、人力的にコンピュータ利用が制約される。
2. 専門のプログラマー（システム分析者）に頼めるが、必要な処理内容を十分に理解してもらうのが大変な仕事である。
3. プログラム作成に時間がかかるので、継続的に利用が見込まれる処理プログラムが中心で、そのときどきに必要が生じた処理は手作業でおこなうことが多い。
4. すでにあるプログラムを利用する場合でも、処理の必要を感じてから結果が得られるまでの時間（Turn around time）が長くかかる。
5. コンピュータ利用に際し、利用者側に余りにも多く利用上の知識、作業が要求されるので利用しにくい。
6. コンピュータの使用コストが利用部門に賦課され、その額が大きい。
7. コンピュータを使って計画を策定しても表現形式が数値データ中心であるた

め、トップマネジメントの利用にとって難解である。

8. 経営計画策定用モデルに適當なものがない。

9. その他（具体的に御記入下さい。）

--	--

問 7. 週刊コンピュータ連載中の「MDS - MISの新しい展開」（1～3）を同封しましたが、貴社における今後のMISの展開を含めて御感想を回答紙にお聞かせ下さい。

1. MDSに関心がある。（MDSのどのような点について関心をお持ちですか。具体的に御記入下さい。）

--	--

2. MDSと同様のシステムを導入ないしは計画している。（さしつかえない範囲で具体的に御記入下さい。）

--	--

3. 関心がない。その場合

A. 現在のシステムで十分である。

B. MDSとは別のシステム構想をもっている。（さしつかえない範囲で具体的に御記入下さい。）

--	--

4.2 経営計画プロセスにおけるコンピュータ利用

はじめに

MDSは実戦的なMISを目標としているところから、その概念的な研究およびシステム構築とともに、実際の企業における経営計画プロセスを対象としたMDSの実験的な導入、運用が望まれる。この点に関して、今年度は幸いにも麒麟麦酒株式会社の御協力を得ることができたので、とくに同社の長期計画プロセスについてのデータを基礎としながら具体的なシステム設計を試みた。

しかし、なにぶんにもMDSの基礎的研究と並行しての研究であったので今年度中においてはまだ実際の経営計画プロセスと関連させてシステムの運用を試みるところまではいっていない。これは近い将来の課題にしたい。

今回、麒麟麦酒株式会社の長期計画プロセスを研究対象としたのは、第一に当協会における他のプロジェクトにおいて同社を対象とするケーススタディが行われていたこと、第二に同社の製品が少品種生産であり、最初のシステムとして適当と思われたこと、そして第三に長期計画はそのプロセスが部外者にも比較的理解しやすいことが予想されたからである。しかしこれらのことはむしろプロジェクトの進行上の都合からといった方がよく、MDSは長期計画や少品種生産企業にのみ適用されるというのではなく、汎用性を備えている。

したがって、このプロジェクトも次年度においては多品種生産企業を対象としたシステム開発を予定している。その際にはこの麒麟麦酒株式会社を対象とした研究の経験が基礎となることはいうまでもない。

以下、同社における長期計画プロセスの概要とコンピュータ利用についてみていくことにするが、その内容は必ずしも同社の計画プロセスを正確にフォローしたものとはなっていないかもしれない。部外者にとって、多数の人々と多くの時間を費して行われている現実の計計プロセスを短時間の間に完全に理解することは不可能であるし、また実験的システムの性格から、細部にわたる部分をかなり省略してもいる。MDSのアイデアを実際の計画プロセスにな

じませていく努力はやはり企業サイドの人々に期待しなければならない。

4.2.1 需要予測と販売計画

どのような業種であれ、長期計画はまず需要予測から始まり、生産、販売、設備、人員計画などの計画に進み、そして最後に予想財務諸表にもとづく業績の目標を立てるという一連のプロセスにはそう大きな違いはないと思われる。これからみようとす長期計画プロセスもその一つの典型的な例といえよう。

需要予測はまずビールの全消費量の予測から始められる（以下の説明では清涼飲料など他の製品は省略する）。このために過去の市場需要を説明するための回帰モデルが作られる。説明変数としてはG N P、物価指数、人口、天候指標、ビール販売価格などが用いられる。長期計画は5年間を計画期間として毎年改訂していくことにしているので、回帰モデルについても前年度の経験を利用できるが、それでも安定した回帰式を得るためにはかなりの本数の回帰式を求める必要がある。回帰計算にあたってコンピュータが利用されていることはいうまでもない。全ビールの需要予測は回帰モデルによる結果とともに、各支店に対してそれぞれの担当地域の需要見通しを求め、それらを積み上げた結果も参考にされる。また、製品需要の過去の伸び率などにもとづく担当者の経験、勘などによる調整が加えられる。このようにして決定されるビールの全需要量は麒麟麦酒社と他社との販売量に区分され、さらに品種別に分解されていく。

同社と他社の区分については過去のシェアの分析とともに販売部門における各市場ごとのすう勢、その他多角的判断が加えられ決定される。また、品種別の分解に際しては過去の品種別の割合の推移が分析され、その結果にもとづく品種の構成比の想定値が用いられる。

麒麟麦酒社全体の品種別の販売量が決定されると、これをさらに支店別に割当て、品種別、月別の販売量が決定される。その際には支店別の過去の販売実績や販売データの時系列分析にもとづく季節指数などが用いられる。このように需要予測、販売計画の段階では製品需要や販売シェア、そして一般経済環境

などについてのデータの収集，維持と時系列分析，回帰分析を中心とする分析手法が必要とされる。

4.2.2 需要と供給能力 — 需給計画

現行の設備能力，稼働状況を前提としたときに販売計画を十分にまかなうだけの生産が可能かどうか，合せて設備能力の新增設の必要性を検討し，設備計画へつないでいこうとするのがこの需給計画である。

工場の生産能力の算定にあたってはさまざまな要因が考慮されなければならない。工場別の生産能力，稼働日数，残業，ストなどによる減産などがその例である。そこでこれらの基礎データの収集，維持とともに，将来に予想されるいくつかのケースについて，あらかじめ供給能力の幅を求めておく必要があるが，そのためにシミュレーション・モデルによる分析を行っている。分析の結果，供給能力の不足が見込まれる場合には，現行設備の操業条件を許される範囲で変更し増産をはかり，なおかつ不足する場合には工場の新増設が検討される。

次に全体としての供給能力を各工場の間にもどのように配分するかが検討される。このためには各支店の需要量，各工場と各支店，各倉庫との距離的關係，工場，倉庫における在庫能力などを織り込んだ形で線型計画法を用いた工場別の最適生産計画が求められる。この様にして最終的な需給計画が決まり，支店別の販売計画，工場別の生産計画そして工場別の新增設計画が決定されていくのである。

このように，この需給計画の段階では各工場から計画の基礎となる各種のデータを求めたり，シミュレーション，線型計画法など各種の数量分析を行うために社内の専門家との協力が必要になってくるのである。

4.2.3 容器，原料，およびその他の材料の受払い計画

ビール会社によって，ビンや箱などの容器類の必要量を正しく予測し，それ

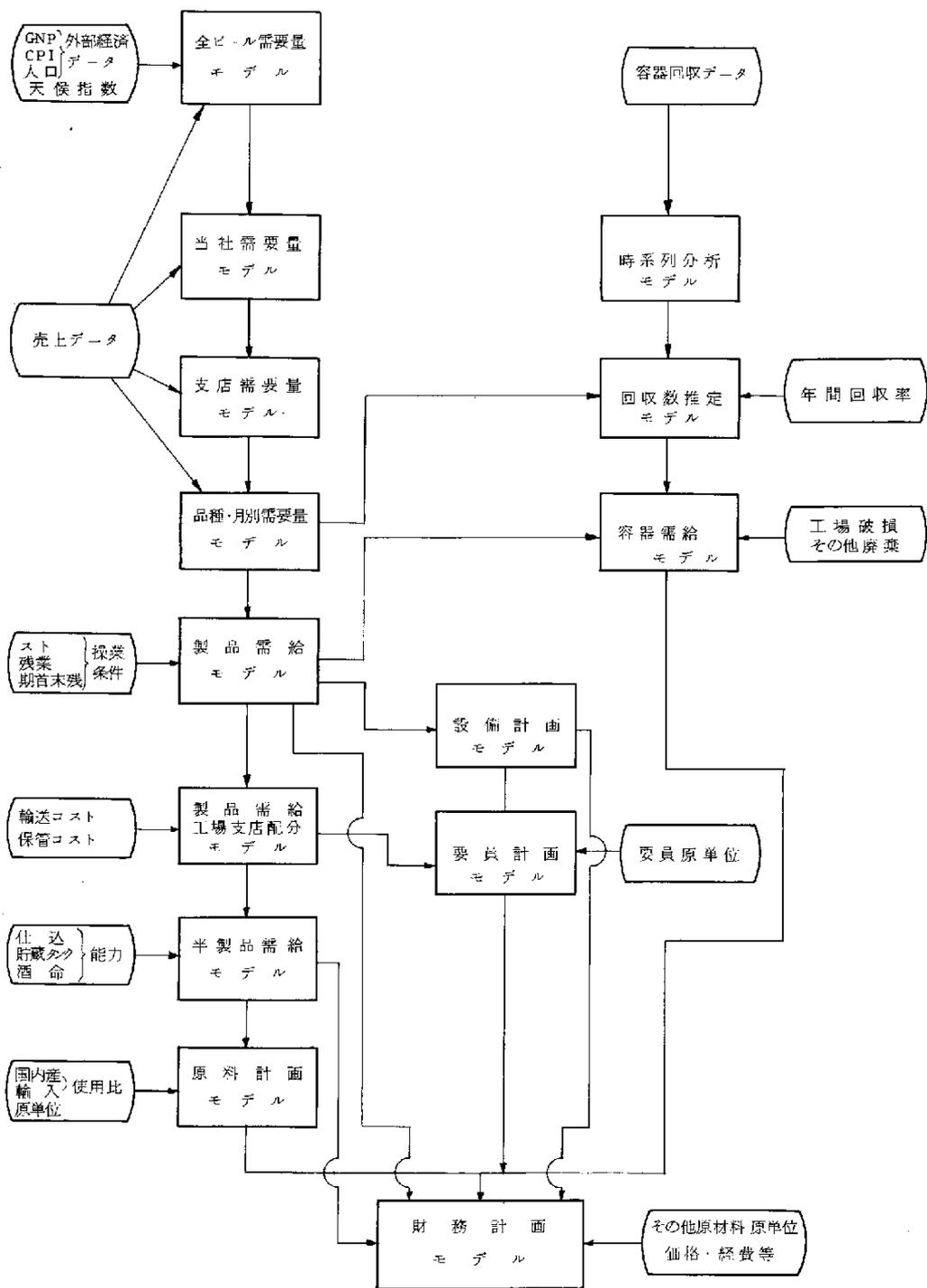


図4-1 長期計画サブモデル関連図

に見合った数量を確保することは生産のための必要条件であり、また容器類のコストの低減をはかることは利益率を高める上での重要な方策の一つとなっている。現在のところ、ビールのビン回収・再使用されるので新ビンの投入必要量を求めるためにはまずそのような古ビンの回収数の予測が行われなければならない。ビンの回収数は、いうまでもなく、出荷されたビールの本数と密接な関連をもっており、それらの関係を統計的に処理した回収予測モデルが作られる。それは基本的にはEPA法などによる時系列分析を用いて行われる。そしてこの回収予測モデルを用い、需給計画において得られた販売数量をインプット・データとして月別の古ビンの回収数が求められるのである。

次にビンの必要本数は需給計画における月別の生産計画に基づいて求められる、これからビンの繰り越し在庫と回収される古ビンを引いた部分が新ビンの必要本数ということになる。なお、ビンの必要本数を算定する場合には工場段階におけるビンの破損に加えて、商品の安全性および品質保全の見地から、不良ビンの廃棄が加味される。

もう一つの容器である箱については、古ビンの回収と関連づけて、箱の回収が予測され、同じく生産本数との関連で求められる必要な箱数と合わせて受払い計画が作られる。

ビールはモルト（麦芽）を主原料として、これにホップや米、でんぷんなどを加え、一定の期間熟成したものをビンにつめて生産される。そこで需給計画に見合ったビールの仕込み量（半製品）の受払い計画が立てられ、それに基づいて醸造工場の需給計画が作られる。次にこの半製品需給計画に基づいて原料の受払い計画が作られる。原料の受払い計画は半製品の仕込み量に各原料の使用比率をかけて各原料の必要量を求め、繰り越した在庫量を差し引いて各月の必要調達量を求めるという比較的簡単な計算によって作られる。

なお、この計算の前提としては、モルトとホップについては国内産と輸入分の割合をどう見込むかといった判断要素が加えられる。同様に缶、王冠栓、ラベルといったその他の材料の受払い計画も製品、半製品需給計画に基づいて

計算し、作成される。

4.2.4 その他の計画

その他の計画には設備計画、要員計画、そして財務計画がある。まず設備計画は製品および半製品の需給計画において決定された設備の新增設について具体的な計画を検討するものである。また設備の新增設にともなり人員の増減をふくめて、生産に必要な人員を確保し、そのために必要な労務費などの算定を行うのが要員計画である。

以上が長期計画に含まれる主な計画であるが、これらの計画は最終的には財務計画に関連づけて統合される。

財務計画の編成にあたっては、まず、それぞれの計画にもとづいて醸造部門、仕上部門といった部門別に費用が集計される。次いで工程別、製品別に原価が分類、整理され、原価計算が行われる。他方で販売計画にもとづいて販売収入が求められ、これらの結果によって、通常の計算手続きを経て予想損益計算書が作成される。財務計画ではさらに予想貸借対照表および資金繰り表が作成され、そして最後に各種の経営分析指標が求められることになっている。以上が麒麟麦酒株式会社の長期計画プロセスのあらましであるが、それはまた他の多くの会社の長期計画プロセスと多くの共通点をもっていることが容易に予想される。

4.2.5 長期計画プロセスにおけるコンピュータ利用

現在のところ、麒麟麦酒社では毎年一回、企画部を中心として長期計画の策定を行っており、需要予測から始まって計画の最終的な承認が得られるまでに相当の期間を要している。長期計画プロセスにおけるコンピュータ利用については需要予測、需給計画などの段階で回帰分析、時系列分析、シミュレーション、線型計画法などに利用されているが、データの集計・整理や原価計算財務計算などにおいては人手による計算処理がかなり含まれている。

計画プロセス	使用者のニーズ	モデル	情報処理機能
製品需給計画	<ul style="list-style-type: none"> ○ 需要予測 ○ 積み上げ予測 ○ 支店・銘柄・月別分解 ○ 需給バランス ○ 	↔	データの入出力 四則演算 伸び率，構成比の計算 回帰分析
容器・原材料の受払い計画	<ul style="list-style-type: none"> ○ 容器の回収予測 ○ 半製品仕込量の算定 ○ 原料必要量の算定 ○ 原料の受払い計算 ○ 容器の受払い計算 ○ 	↔ ↔	E P A 法 線型計画法 シミュレーション ……
設備・要員計画	<ul style="list-style-type: none"> ○ ビン詰設備の投資額計算 ○ 醸造設備の投資額計算 ○ 人員の増減計算 ○ 人件費の計算 ○ 	使用者のニーズと情報処理機能のセットとを対応づける	
財務計画	<ul style="list-style-type: none"> ○ 部門費集計 ○ 工程別原価計算 ○ 製品別原価計算 ○ 予想財務諸表の作成 ○ 		

図4-2 長期計画のシステム構成

同社の長期計画プロセスにはこれまでみてきた範囲ではそう大きな問題点は見当らない。製品や原材料の種類も少ないし、工程も比較的単純である。またコンピュータ利用についてもシステム担当者と計画スタッフの意思疎通はうまくいっているようである。ただ、計画策定に要する期間がかなり長いこと、計画策定の途中や予想利益などの最終結果をみた上で計画を作り直すといったことができにくい、などの点を指摘することができる。

われわれは麒麟麦酒社の長期計画を対象とするMDSの第一段階として、とりあえず現在使用されている各種の分析手法を含むようなシステムを構築することにした。その概要は図4-2のとおりである。

たとえば需要予測や容器の回収予測にはデータの入出力、回帰分析、EPA法などが用いられるが、それらは情報処理機能の一部として組み込まれ、モデルによる定義を媒介として使用者のニーズと関連づけられることになる。同様に容器や原材料の受払い計算はデータの入出力や四則演算機能と対応づけられよう。システムの具体的な構成については前述のカードベース・システムの中で説明されている。

参 考 资 料

- (1) ADL systems, Good management of computer operations, Part II,
data bases, Computers & Automation, April, 1973.
- (2) Auerbach Corp. "Data Management System Study Final Report"
Technical Report 1469-TR-4, 1968.

B-1

- (1) Barnes J., "A Survey of Information Storage and Retrieval
Software Packages", Continuing System Education, 1969.
- (2) Benjamin R.I., A generational perspective of information system
development, Com. of the acm, July, 1972.
- (3) Berkeley E.C., Computer programming using natural language, Computers
& Automation, June, 1973.
- (4) Berkeley E.C., Computer programming using natural language,
continued, Computers & Automation, July, 1973.
- (5) Blumberg D., Information systems and planning process, Data
processing magazine, May, 1966.
- (6) Blumenthal R., Management in real-time, Data processing magazine,
1965, April.
- (7) Boehm B.W., Software and its impact, A quantitative assessment,

Datamation, May, 1973.

- (8) Bonner, R.E. "On some Clustering Techniques," IBM Journal Res. Develop., January, 1964.
- (9) Borko, H. & M.D. Bernic, "Document Classification," JACM, April, 1968.
- (10) Bowman, R & A. Lickhater "Graphical Data Management in Time Shared Environment," "SJCC, Proc., 1968.
- (11) Brady, R.H., Computers in top-level decision making, HBR, July-August, 1967.
- (12) Brandon D.H., Computer acquisition method analysis, Datamation, Sep., 1972.
- (13) Brown J.F., Interactive simulation for banking, IBM system journal, No. 2, 1973.
- (14) Burch J.G., Tailoring the information system, Journal of systems management, Feb., 1973.
- (15) Burkhard W.A., Some approach to best-Mach file searching, Com. of the acm, April, 1973.

B-2

- (1) Benton W.K., The use of computers in planning, Addison-wesley, 1971, (IIT)

- (2) Berztiss A.T., Data structures, Academic press, 1971. (IIT)
- (3) Boutell, W.S., Computer-oriented business systems, 2nd ed.,
1973, Prentice-Hall. (YCU)
- (4) Brink, V.Z., Computers and management, Prentice-Hall, 1971. (Top)

C-1

- (1) CODASYL Systems Committee, A survey fo generalized data base
management systems, May, 1969. available through ACM.
- (2) CODASYL Systems Committee, Feature analysis of generalized data
management systems, May, 1971. available through ACM.
- (3) Coleman, MIS, Management dimensions, Holden Day, 1973. (JPC)

C-2

- (1) Carbonell, J.R., On man-computer interaction, a model and some
related issues, IEEE Trans. Systems Science and Cybernetics,
Jan., 1969.
- (2) Chauhan, R., A computer graphics assisted system for management,
AFIPS, NCCE, Vol. 42, 1973, pp. 197-202.
- (3) Chen, P.P.S., Optimal file allocation in multi-level storage
systems, AFIPS, NCCE, Vol., 42, 1973, pp. 277-282.

- (4) Coleman, MIS Management Dimensions, Holden Day, 1973.
- (5) Coleman, M.L., ACCNET- A corporate computer network, AFIPS? NCCE, Vol. 42, 1973. pp. 133-140.
- (6) Conway R.W., et., On the implementation of security measures in information systems, Communication of the acm, April 1972, pp. 211-220.
- (7) CODASYL Development Committee, "On Information Algebra-Phase 1 Report," CACM April 1962.
- (8) Crick, M.F.C., A data-base system for interactive applications, report no., G320-2058, IBM Corp., Scientific Center, Cambridge, Massachusetts.
- (9) Cronin, Interactive processing, - A user's experience, AFIPS FJCC, 1972, Part II, pp. 1037-1064.

D-1

- (1) Dana C., An information structure for data base and device independent report generation, AFIPS, FJCC, 1972, pp. 1111-1116.
- (2) Day, L.H., The future of computer and communication services, AFIPS, NCCE, Vol. 42, 1973, pp. 723-734.
- (3) Dearden J., et., How to control the computer resource, HBR

Nov. - Dec., 1973.

- (4) Decarbonnel, F.E., Information source for planning decisions,
California Management Review, Summer 1973.
- (5) Dodd, G.G., Elements of data management systems, Computing surveys,
June 1969.
- (6) Dijkstra, E.W., Notes on Structured Programming, "Structured
Programming", pp. 1-82, Academic Press, London.

E-1

- (1) Ellis M.F., SIMS- An integrated user oriented information systems,
AFIPS, FJCC, 1972 Part II, pp. 1117-1132.

F-1

- (1) Fraser A.G., On the interface between computers and data
communication, systems, Com. of the acm, July, 1972.

G-1

- (1) Gold, M.M., Time-sharing and batch-processing: an experimental
comparison of their values in a problem solving situation,
Com. of the acm, May, 1969.

- (2) Gourley D.E., Data communications, Datamation, Oct., 1972.
- (3) Green, L, E.S., E.C.Berkeley & C.Gotlieb, Connersation with
Computer, Computer and Automation, Oct. 1959.

H-1

- (1) Hanold T., An executive view of MIS, Datamation, Nov., 1972.
- (2) Hassitt A., Implementation of a high level language machine,
Com. of the acm, April, 1973.
- (3) Higgins J.C., Computer aids to corporate planning,
The computer bulletin, Sep., 1972.
- (4) Hsiao D., A system to manage an information system, Proceedings
of the IFIP joint conference on mechanized information
storage retrieval, Rome, Italy, 1967, pp. 637-660.
- (5) Huang J.C., A note on information organization and storage,
Communication of the acm, July 1972.

H-2

- (1) Handbook of data processing management, Vol. 1 - Vol. 6,
Auerbach Publications, 1971. (IIT)
- (2) Hanssmann F., Operational Research in the design of electronic

data processing systems, The English University Press, 1973.

- (3) Harrison M.C., Data-structures and Programming, Scott,
Foresman and Company, 1973.

I-1

- (1) Ishizaki Sumio, MIS to database, Computer Report, 1973 June.

J-1

- (1) Judd D.R., Use of files, Macdonald-Elsevier, 1973.

K-1

- (1) Katz, Jasac H. & William C. McGee, An Experimental in Non-procedural
Programming, 1968, FJCC Proceeding.
- (2) King P.F., Database sharing- An efficient mechanism for supporting
concurrent processes, AFIPS, NCCE, Vol. 42, 1973, pp. 271-276.
- (3) Kingston P.L., Concepts of financial models, IBM System Journal,
No. 2, 1973.
- (4) Kobayashi, I., Introduction to the information system, PART 1
Data Base Structure Model, Soken Kiyo, Apr., 1971.
- (5) Krinos, J.D., Interaction statistics from a database management

system, AFIPS, NCCE, Vol. 42, 1973, pp. 283-290.

- (6) Kurz R.C., Long live the data administration, Datamation, March, 1973.

K-2

- (1) Kaimann R.A., Structured information files, Wiley, 1973.
- (2) Kanter, J., Management-oriented management information systems,
Prentice-Hall, 1972. (YCU)
- (3) Kelly, J.F., Computerized management information systems,
Macmillan, 1970.
- (4) Kindred A.R., Data systems and management, Prentice-Hall, 1973. (IIT)

L-1

- (1) Lande H.F., Planning-data systems, IBM system journal, No. 2, 1973.
- (2) Larson W.R., The meaning of an integrated data systems,
Computers & Automation, April, 1972.
- (3) Lewis J.C., Xerox computer service, interactive accounting,
Datamation, 1973, July.
- (4) Little J.D.C., Models and managers, the concepts of a decision
calculus, Management science, April, 1970.

L-2

- (1) Lancaster, F.W., Information retrieval on-line, Wiley, 1973.

M-1

- (1) Mancineilli C.T.T., Management Information Systems,
The trouble with them, Computers & Automation, July 1972.
- (2) McLaughlin, Building a data base, Datamation, July 1972.
- (3) Meiners E.E., A machine-independent data management systems,
Datamation, June, 1973.
- (4) Minker, J & J.Sable, File Organization and Data Management,
Annual Review of Information Science and Technology, 1967.
- (5) Mintzberg H., The Myths of MIS, California Management Review,
Fall, 1972.
- (6) Morton M.S.S., Terminal costing for better decisions, HBR,
May-June, 1968.
- (7) Moskowitz H., The value of information in aggregate production
planning- a behavioral experiment, AIIE Transaction, Dec., 1972.

M-2

- (1) Martin, J., Design of man-computer dialogues, Prentice-Hall, 1973.(IIT)

- (2) Martin, J., Design of real-time computer systems, Prentice-Hall, 1967.
- (3) Martino R.L., Management information systems, MDI Publications, 1969.
- (4) Matthews D.Q., The design of the management information system,
Auerbach Publishers, 1971.
- (5) McFarlan F.W., Information systems administration, Holt,
Reinhart and Winston, 1973. (YCU)
- (6) Meadow C.T., The analysis of information systems, Melville
Publishing Company, 2nd ed., 1973. (IIT)
- (7) Morton M.S.S., Management decision systems, Harvard University, 1971.
- (8) Murdick R.G., & Ross J.E., Information systems for management,
Prentice-Hall, 1971.

N-1

- (1) Nakamura Yoshiro, Data base Nyumon, Business Communication,
Vol. 8, No. 7, 1973.
- (2) Nunamaker, J.F., Specifications for the development of a
generalized data base planning system, AFIPS, NCCE, Vol. 42,
1973, pp. 259-270.

O-1

- (1) Ollie T.W., MIS, data bases, Datamation, July, 1970.

P-1

- (1) Patterson A.C., Data base hazards, Datamation, July, 1972.
- (2) Patterson A.C., Requirements for a generalized data base management system, AFIPS, FJCC, 1971.
- (3) Prywes N.S., Man-computer problem solving with multilist, Proceedings of the IEEE, Vol. 54, No. 12, Dec., 1966.
- (4) Pyles N.R., Planning a data communication systems, Datamation, Nov., 1972.

P-2

- (1) Philippakis, A.S., Information Systems Through COBOL, McGraw-Hill, 1974.

R-1

- (1) Ran P., Evaluating EDP function, Datamation, Sep., 1972.
- (2) Ravin, J., An interactive graphics system for analysis of business decisions, IBM system journal, No. 3, 1973.

- (3) Rhodes, T., The computer aided design environment, COMRADE- An overview, AFIPS, NCCE, 1973, Vol. 42, pp. 319-324.
- (4) Rogers W.P., Instant financial information, Datamation, Feb., 1973.
- (5) Ruth S.S., Data compression for large business files, Datamation, Sep., 1972.

R-2

- (1) Radford, K.J., Information Systems in Management, Reston, 1973.
- (2) Rustin R., Data base systems, Prentice-Hall, 1972. (IIT)

S-1

- (1) Salasin J., Hierarchical storage in information retrieval, Com. of the acm, May, 1973.
- (2) Schubert R.F., Basic concepts in data base management systems, Datamation, July, 1972.
- (3) Senko M.E., Data structures and accessing in data-base systems, IBM system journal, No. 1, 1973.
- (4) Shneiderman B., Optimum data base reorganization points, Com. of the acm, June, 1973.
- (5) Simmons, S.F., Answering English Questions by Computer; A Survey,

CACM Jen., 1965.

- (6) Sprague H.H., System support for a financial planning models,
Management accounting, Jun., 1972.
- (7) Steig D.B., File management systems revisited, Datamation, Oct., 1972.
- (8) Streeter D.N., Centralization or dispersion of computing facilities,
IBM systems journal, No. 3, 1973.
- (9) Stutzman B.W., The meaning of the IBM 3705, Aug., 1972, Datamation.
- (10) Sundeen, D.H., General Purpose Software, Datamation, Jern., 1968.

S-2

- (1) Sackman H., Online planning, Prentice-Hall, 1972. (Top)
- (2) Sanders D.H., Computers and management, McGraw-Hill, 1974.
- (3) Stamper R., Information in business and administrative systems,
Wiley, 1973. (YCU)

T-1

- (1) Teichroew, D., A survey of languages for stating requirements for
computer based information systems, AFIPS, FJCC, Part II, 1972,
pp. 1203-1224.
- (2) Testa C.J., The evolution of man-computer symbiosis,

Computers & Automation, May, 1973.

- (3) Theis D.J., Communication processors, Datamation, Aug., 1972.
- (4) Tobias, The future of remote information processing systems,
AFIPS AJCC, 1972 Part II, pp. 1037-1044.
- (5) Tomasewski L.A., Decentralized development, Datamation, Nov., 1972.

W-1

- (1) Wagenhals L.W., Remote terminals systems for computer, Computers &
Automation, Sep., 1973.
- (2) Welke L., A review of file management systems, Datamation, Oct., 1972.
- (3) Whitney, V.K.M., Fourth generation data management systems,
AFIPS, NCCE, Vol. 42, 1973, pp. 239-244.

W-2

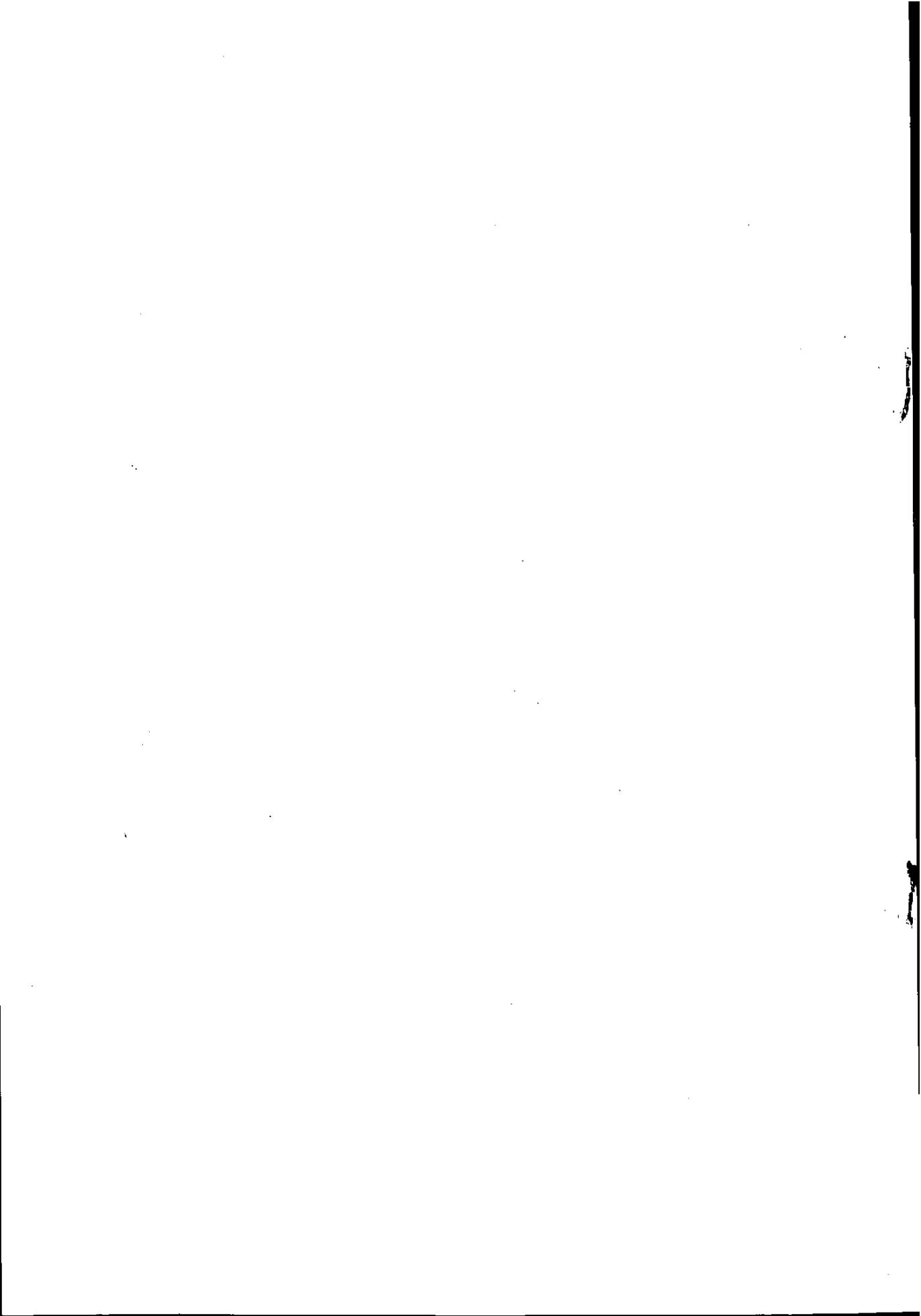
- (1) Wheelwright S.C., Computer-Aided Modeling for managers,
Addison-Wesley, 1972.

Y-1

- (1) 矢矧晴一郎, データ・マネジメント・システムの現状,
EDPスペシャルレポート⑩, 日本能率協会, 1973年6月

Y-2

- (1) Yourdon E., Design of on-line computer systems, Prentice-Hall, 1972.



————— 禁 無 断 転 載 —————

昭和 51 年 3 月 発行

発行所 財団法人 日本情報処理開発協会
(旧(財)日本情報処理開発センター)

東京都港区芝公園 3-5-8

機械振興会館内

TEL (434) 8 2 1 1 (代表)

印刷所 山 幸 印 刷 社

東京都港区東新橋 2-1-3

TEL (431) 6 5 1 0

50-R005

