

經濟經營研究叢書

經營機械化シリーズ18

經營機械化研究の展開



神戸大学經濟經營研究所

1978

経営機械化研究の展開

神戸大学経済経営研究所

目 次

日本経営機械化史補論 ……………	米 花 稔	1
—— 昭和40年代 ——		
ノルウェーにおけるマイクロデータ・ファイル……………	能 勢 信 子	21
システムの現状と問題点		
大阪市におけるEDPSの分布について ……………	都 藤 希八郎	35
会計学シソーラスの設計と編成 ……………	生 島 芳 郎	47
バザラのアルゴリズムに対する注釈 ……………	伊 藤 駒 之	63
対話型簡易汎用作表プログラム ……………	下 條 哲 司	73
—— Dialogue-CROTAB ——		
マーケティング戦略教育ゲーム……………	布 上 康 夫	95
—— SMSSの解説 ——	萩 野 典 宏	
制御シミュレーション・プログラム ……………	内 田 幸 夫	163
—— OPCON ——		
構文記述言語 FOIL V-10 ……………	杉 浦 一 平	177
対話型書類管理システム……………	民 野 庄 造	291
—— SELF ——		
情報処理システムにおける行列演算の記述法……………	定 道 宏	317

日本経営機械化史補論

——昭和40年代の特徴と課題——

米 花 稔

目 次

1. 開 題
2. 昭和40年代の推移概観
3. 情報システムのネットワーク的展開
4. M I Sのその後の推移
5. コンピュータリゼーションと日本的経営
6. その他の若干点について
7. むすび—実態と情報について

1. 開 題

この経営機械化シリーズで4回にわたって試論的に発表したものに、修正補充を行って、さきに『日本経営機械化史』（昭和50年11月、日本経営出版会刊）を不十分なが一応とりまとめた。これはわが国における PCS から EDPS までの経営機械化の推移を戦前戦後にかけて考察したのであるが、それはコンピュータ第3世代のはじまった昭和40年ごろでおえているのである。目まぐるしい経営機械化の展開のなかで、いわば歴史として一応定着したもののみられるのは、そのころまでと思われたからである。また昭和40年代からこの10年の展開が、余りにも広範にかつ多様に推移しつつあるので、これをとりまとめるのは、他により適当な人があると思われ、さらに形式的に、またそのころからマクロ的には、コンピュータ白書などの諸資料が年々まとめられていることでもあるので、その前の時期で筆をおいたのである。しかしながら、経営機械化なり、情報システムの諸問題に関心をもちつづけ、さらに今後についても何ら

かの考え方をもちたいと思っている筆者としては、第3世代に入って、さらに3.5世代ともいわれるこの10年の推移における特徴と課題についても、当然のことながらあるていど考察しておく必要があると思うのである。この10年についてここでも多少ともふれておきたいと思うのである。

また筆者の『日本経営機械化史』について、貴重な書評をいただいたことから、多少とも補論的考察をしたいと思うようになった。甲南大学経営学部の山本純一教授からの『国民経済雑誌』第134巻第3号（昭和51年9月）において、過分の紹介をいただいたあとで、「この研究成果を基礎として、経営機械化が、例えば現代的経営の理念や体質ないし行動をいかに変革してきたし、しようとしているか。経営のシステムや、機能ないし経営管理の実践的体系としての組織が、いかに革新されようとしているか等、多くの研究課題にとりくまねばならないであろう。そしてそれによってこそ立体的な経営機械化論の研究がさらに促進されるであろう」との指摘をいただいた。

さらに神戸大学経営学部の小野二郎教授からも、『経営史学』第11巻第2号（昭和51年10月）において、行き届いた紹介と特徴の指摘のあとで、「上のような特長が反面本書の弱点ともなっている。つまり各時代において機械化の対象となった個々のシステムを具体的に説明することがなされなかったために、例えば同じ造船業の原価管理システムであっても、PCS時代と現代とではどのように異なるのかが明らかにされえず、……また断片的な叙述としてはかなり重要な指摘もみられたが、経営機械化の経営活動全般に及ぼす影響について、史的考察を加えた独立の一章がなかったのが惜まれる」との指摘をいただいた。

いずれも、きわめてもっともな指摘である。今後とも関心をもちつつけるなかで、考察したいと思っているけれども、この補論において今直ちに多少ともこれらの諸点に論及する能力と余裕をもっていないことは申訳ない次第である。とはいえこのような貴重な書評を問題意識としつつ、そのためにも、この10年の推移についても多少とも考察を進めたいと思ったのである。

そのために、昭和51年中において、若干の時間をさいて、東西の電力会社、

東京、大阪に本部をもつ都市銀行の一部、証券取引所ならびに関連企業、繊維から自動車、電機にわたるきわめて一部の製造業などの情報システムの実態にふれて、ほんのアウトラインながら、この10年の推移をかい間みた。これとともに手元の若干資料を参照しつつ、考察を試みたのである。

ここでは、その結果を、情報システムのネットワーク的展開、MISのその後の推移、コンピュータリゼーションと企業経営、とりわけ日本的経営とのかかわり、その他若干の問題にわけて、概論的な考察を展開することとした。問題が広汎なのに対して、きわめて限られた時間と不十分なとりくみにとどまったのであるが、『日本経済機械化史』の補論という意味でとりあげたのである。

2. 昭和40年代の推移概観

この10年間のコンピュータをめぐる推移はよく知られているところであるが、『日本経営機械化史』の補論という意味と、後論展開の便宜から、ここで簡単に示すこととする。

- ①昭和40年代後半は、電子計算機産業の輸入の自由化、技術の自由化、そして資本の自由化の展開で特徴づけられる。それは昭和51年4月1日のソフトウェア100%資本自由化で一応終わった。
- ②それは、昭和39年(1964年)4月 IBM 360シリーズの発表からの第3世代の内外各社のこれに対応する新機種シリーズの発表、昭和45年 IBM 370シリーズの発表からの3.5世代といわれるなかで、国産6社の集約化による富士通と日立、日本電気と東京芝浦、沖電気と三菱電機のグループ化でのそれぞれMシリーズ、ACOS シリーズ、COSMO シリーズの昭和49年(1974年)代の発表に至る推移として、具体的に展開せられた。
- ③またこの時期に、データ通信回線の利用の制度化によって、コンピュータの利用の広域的展開が急速にみられるようになった。いわゆる通信回線の開放を主とする公衆電気通信法の一部改正が昭和46年5月に公布され、データ通信の従来の専用線にくらべ共同使用、他人使用の制限が相当大巾に緩和せら

れることとなった。後にもふれるように、オンライン・システムが急速に増加することとなった。

- ④1974～75年の国際比較で汎用コンピュータ設置台数は、アメリカが72,000台、わが国がこれについて32,000台で第2位、これにつぐのは西ドイツの12,000台である。これを第3世代のはじまった1965年のアメリカ26,340台、第2位西ドイツ1,980台、第3位わが国の1,790台となっており、ほぼ10年間にアメリカは2.7倍、西ドイツは6倍、そしてわが国は17.8倍の増加を示している。規模を別にすると、相当急速な普及ぶりである。このわが国の昭和50年（1975年）末の普及32,447台のうち製造業9,564台で、10年前にくらべて11.0倍であるのに対して、卸、小売、商事は9,879台で49.6倍、金融機関3,882台で31.3倍、サービス業（一般サービス、情報サービス）2,423台で23.7倍となっていて、3次産業における普及の顕著なことが目立っている。さきのデータ通信の利用ともかかわっていると思う。

- ⑤またこの間の情報処理産業の発展も著しいものがあつた。日本電子計算機株式会社調査によると、計算センターならびにソフトウェア開発企業は、昭和51年1月1日現在で842事業体、これを昭和40年4月末の108事業体と比較すると、この間の激増が知られる。この842のうち、独立計算受託企業734、公共機関55、メーカーおよび関係販売会社53となっている。

きわめて大まかなこの10年の発展の一部を示したにすぎないが、年々のコンピュータ白書（日本情報処理開発協会刊）などに詳しいので、これ以上はふれない。以下では、さきにのべたように、コンピュータ利用が、第3世代に入って、広域的展開としてのオンライン・リアルタイムの展開、集約浸透としてのMISの展開の両面に、相関連しつづみられるので、これらの側面から、筆者のささやかな見聞と手元の資料によって、この10年の推移と特徴と課題について若干の考察を試みることにする。

3. 情報システムのネットワーク的展開

わが国の情報システムにおけるオンラインのネットワークは、『日本経営機械化史』でも示したように、昭和39年（1964年）から始まっている。国鉄のみどりの窓口、日本航空、全日空の座席予約システムから、東海銀行の内国為替オンラインなど、ついで昭和40年には、三井銀行の普通預金の一部の地域でのオンライン、とつづき、やがて製造業でも自動車工業における組立ラインのオンライン管理、販売管理オンラインなどと展開したことは、まえにみたところである。

特定通信回線利用による自営のオンラインシステムは、昭和39年（当時はデータ通信専用回線）に5システムで始まったのが、昭和50年3月末で1,126システムに達し、対象業務別では、製造業、商事会社などの生産、販売、在庫などの事務管理を主とする370システム、銀行、信用金庫などの預金、為替業務が181システムなどとなっている。このほかに、電電公社自らのデータ通信サービスとして昭和43年から始めたものに、50システム（51年度末）がある。このうちユーザーのための各種データ通信サービスと、昭和45年からのDRESS（販売、在庫管理サービス）、昭和46年からのDEMOS（科学技術計算サービス）などの公衆データ通信サービスとがある。いずれにしても、さきにふれたように、コンピュータのいわゆる3.5世代化、周辺端末機器の発達と、データ通信回線利用についての昭和46年の改正が、その背景となっていることはいうまでもない。しかしながら、これらの概況は、諸資料で既に明らかなどころで、これ以上ここにふれる必要のないところである。

ここでは、この1年の筆者のささやかな見聞と手元の資料による若干の考察を試みることにしたい。

第1は、情報システムの展開の方向についてである。筆者は第3世代の情報システムは、MISの考え方にもとづく集約的浸透と、ネットワークによる広域的展開とを、相関連しつつも、2つの方向とみたのであるが、意思決定と直接にかかわる集約的浸透より、より日常業務的な広域的展開が、より先立つとみていたのであるが、ほぼその方向での展開が、この10年に顕著にみられたということである。MISを集約的浸透と位置づけた方向については次項にのべ

る。

第2は、企業におけるオンラインによるネットワークの展開は、製造業の場合には、なにより、製造、販売における管理の高度化が目指されたのであり、3次産業の場合には、金融機関における預金、為替にみられ、また電力会社の需要家情報システム、証券会社の顧客情報ファイルにみられるように、顧客サービスないし、マーケティング的視点が重視せられてのこととみられるのである。

第3は、情報システム部門とその実働部門であるコンピュータ部門ないし情報処理部門関係についてである。製造業においてはいずれも社内部門として位置づけられていることが多いようであるが、3次産業においては、金融機関、電力会社、証券関係など若干の見聞にとどまるものの、いずれも同一社内の部門とするものと、情報処理部門を別会社とするもののが、いずれの場合にも対照的にみられたことは興味深い。経営者のフィロソフィ、経営政策、管理上の考え方などの相異にもとづくもので、経営問題としての特徴点のひとつといえよう。

第4は、3次産業と1部の製造業において、情報処理ないしコンピュータ部門が、その組織上の位置づけのいかんをとわず、施設的に独立した立地と建物によるものが次第に増加しつつあることも最近の顕著な特徴といえよう。それはとりわけ昭和40年代後半から、東京、大阪でみられ、やがて昭和40年代末には、俗にコンピュータ・マンションと称せられるような不動産業の対象としての事業もみられはじめるに至った。

第5は、オンラインのネットワークは、必然的に企業の境界をこえる展開を示しつつあることも、特徴のひとつである。製造業の場合には、部品系列とか販売系列ないし、なんらかの関連企業という範囲の企業の境界をこえる情報システムを主としているようであるが、3次産業の場合は、文字通り企業の境界をこえるネットワークの展開が大規模にみられるようになって、新しい課題をもたらしつつあるように思う。

(ケ) その典型は金融機関にみられる。全国の地方銀行の内国為替オンラインが

はじめられたのは、昭和43年10月のことであった。これが全国の銀行のオンライン化となったのは昭和48年4月である。都市銀行、地方銀行をふくみ、信託銀行、長期信用銀行など88行、約7,200ヶ店からなっている。いずれも電電公社と共同であることはいうまでもない。昭和51年10月には全国信用金庫連合会（470社4,691社）が全国信用金庫データ通信システムをはじめ、やがて昭和54年にはさきの全国銀行データ通信システムへの統合が予定せられているという。

- (イ) 証券取引所を中心とする情報伝達機械化システムもまた典型例のひとつである。東京、大阪の証券取引所、主要証券会社は、コンピュータ導入の先駆の一分野であることは、『日本経営機械化史』にも詳論したところで、ここにはくりかえさない。東京証券取引所の売買立合場の市況情報を、証券会社、各地証券取引所などと即時オンラインによる伝達システムが始められたのは、昭和49年10月からである。これには、昭和46年10月設立の株式会社市況情報センターが介在して、Quickビデオ・システムとして、ボード、あるいはビデオによる表示サービスが行なわれ、また本社による各社関連情報提供が行なわれている。

以上の2例は、企業の境界をこえるオンライン・システムの典型というべく、その結果としての新しい課題をもたらしつつある。前者金融機関の場合、小切手処理のコンピュータ化から先行したアメリカのEDPS以上に、わが国の場合はかえっていわばペーパーレスの段階まで先行している部分が見られるようになり、後者証券情報システムの展開が、今日地方証券取引所、従ってまた地方所在証券会社の存立問題にも、すくなくない影響をもたらしつつあることは、現に知られる如くである。

オンラインによる情報システムのネットワーク化は、既にみたように、日常業務処理の広域的展開から始まりつつ、実態の把握の即時化、それによる管理の高度化、現場における意思決定への情報提供など、漸次高度の情報処理機能をもたらす、おのづから、いわゆるMISの展開へとかかわってきつつあることが知られる。このことが、マネジメントのプロセス、組織のあり方、意思決

定など経営のあり方にも次第に影響をもたらしていることもうかがえる。このことはさらに後の項でみることにする。

4. MIS のその後の推移

前著にものべたように、MIS の概念は、わが国の場合、第3世代のはじまる前後、昭和30年代末から一部に紹介導入せられ、専門分野で関心がもたれはじめたものの、よく知られているように、昭和42年秋トップ・マネジメントの視察団の訪米、昭和43年はじめその報告提言から、広く一般に MIS への関心をかきたてられた。それから数年いわゆる MIS ブームの観を呈したものの、この数年は MIS の用語は余り口にせられぬようになった。昭和40年代10年間のこのような推移は何を意味するかを、ここでとりあげてみたいのである。

MIS は、一応「あらゆる階層の管理者に、彼らに影響をあたえる経営活動のすべてについて、つねに完全な情報を提供することを目的とするシステム」とし、これをひとつの構想性としてフレキシブルに位置づけられるものと思う。これが、昭和43年から数年ブームのようにとりあげられ、その後は次第にこの用語のきかれなくなったという現象について、この10年の実態をかきまいた印象として、次のような諸点があげられると筆者は考えている。

第1は、なによりブームとさえいわれるように、MIS という用語にいさかかふりまわされたことからの反省があったであろう。

第2は、コンピュータがすぐ意思決定に役立つという安易な考え方への反省と、さきにみたような日常業務の情報処理の広域的展開への重点移行もあったであろう。

第3は、より実質的なもので、MIS に刺激されつつ、堅実に経営機械化を進めてきたところでは、公私、その分野、業種、業態に応じてみずからの経営体の特徴的にもっている課題に焦点をおいて、経営における情報の役割を正しく理解して、そのとりくみの方向性を、個性的にはば見定め得たからと思われるものもすくなくないようである。

第4は、高度経済成長のなかでのマネジメントへの役立てとしてとりくんだ MIS 的接近について、オイル・ショック以後の環境変化、低成長下のマネジメントのなかでの位置づけを検討する段階にきたことも、無関係とはいえないであろう。

このようにみると、MIS が一般概念としてより、歴史的な概念として登場したという意味が明らかとなり、かつての IDP とかトータル・システムという用語と同じように、今日も十分その意義をそれぞれもちつつも、経営機械化の展開過程における段階的役割を、よりになってきたと理解できるのである。

以上のような推移について、限られた事例ながら、その見聞した実態にもとづいて、今日の MIS の位置づけを考察してみよう。

コンピュータによる経営機械化に積極的であった事業者のすくなくないものは、1960年代後半（昭和40年代早々）MIS が一般的にとりあげられるようになったころには、既に自らの業態と問題意識に適合した情報システムの方向性のあるていど把握していたとみてよいと言う。

そのなかで第3世代の諸機種が登場と周辺機器、データ通信の発達によって、さきにふれたように、オンラインによるネットワークが急速に展開し、日常業務の処理から管理の高度化、やがては現場における意思決定への役立てにも部分的ながら情報提供を可能にするようになってきた。

しかしながら、本来 MIS は、情報の一元化、統合化が実現しなければ、その機能を十分果すことは期待できない。情報のいわば倉庫機能を果すべきいわゆるデータ・ベース形成が前提とならなければならない。それには、何より1970年代になってのハードウェアとしての IBM 370 シリーズなど3.5世代コンピュータの登場による記憶容量の大規模化と高速化が、これを促進することとなった。(イ)しかしながら、これを実働させるプログラムの開発、(ウ)情報体系の再編成、そのためのマネジメント上の諸問題を克服するのに時間をかけねばならなかった。このことが、さきにふれたような今日の MIS の位置づけとなっているといえよう。すなわち、

① ある代表的電機企業では、トップ経営者の積極的姿勢によって、昭和43年

早々自社独自の MIS 的構想とフレームワークをつくり、そのなかでの分権化にもとづく各事業所自らの情報システムづくりを進めた。その前提として 2 カ年間に、各事業所のトップならびにミドルマネジメントに対する 1 週間教育を順次施している。昭和48年に一応の成果を得たようであるが、それは MIS を目標としつつ、現実には、オペレーショナル・システムとしてのレベルを達成したということであった。オイル・ショック以後のマネジメント見直しの時期にあたって、情報システム化を急いだための過剰サービスの見直し、短期間における機械化にともなうアンバランスなどの検討が当面の課題になっているという。

- ② 自動車工業のある代表的企業は、全体の経営機械化のあり方が、多様化と大量化という業態のもつ特徴を焦点としているが、そのなかで製品企画から試作をへて製品化までの品質、原価、販売をふくむ情報検索システムのための、部品を中心とするデータ・ベースづくりを昭和40年代末近く達成したという。MIS がいわばミドル・マネジメントのレベルに達したという。より上位の意思決定のための MIS にも、そのとりくみのプロジェクトを発足し、その展開のプロセス自体が当面の成果になりつつあるという。
- ③ さきにもみたオンライン・システムの高度に展開している金融機関（ここでは都市銀行）の若干の事例についてみてみよう。代表的な都市銀行の若干の事例は共通して、本店の企画部門で、部門別のデータベースをほぼ構築しおわって、人事、支店、顧客などについて、ディスプレイ装置を利用する管理、企画に役立てるというレベルでの MIS に到達しているようである。これら部門別の内部情報のより統合的な活用をいかに進めるかが現段階であるといわれる。
- ④ 給配電システムから料金調定業務まで、高度にコンピュータ化の進んでいる電力会社にあっては、早くから技術設計計算、需要予測など計画分野でのコンピュータ利用も進められているが、1社はほぼ整備せられた部門情報システムから、その部門情報の総合化を目指してとりくみはじめたのが昭和50年からといわれ、また他の1社は定型的業務の機械化が完了して、総合計画

管理システムの機械化をはじめたのが昭和50年とするなどの事例がみられる。

以上、極めて一部の実態のアウトラインでも知られるように、いわゆるMISの現段階での位置づけは、その方向性なりフレームワークとしての役割を果し、現実にはミドル・マネジメントなり、部門毎の管理から、計画なり意思決定における情報機能にあるていど実現してようやく本格的とりくみに至りつつある段階といえるようである。そしてそれがまさしく昭和50年ごろとみられる。

同時に、あたかも高度成長から、低成長なり安定成長といわれる環境への変転に当面して、これまでの過渡期における促進的な役割のための情報の過剰サービスの見直し、情報システム化における生産性向上視点を重視しつつある段階ともいえると思う。

5. コンピュータリゼーションと日本の経営

高度成長のあとにきたオイル・ショックによる急激な環境変化は、これまでのわが国の企業経営のあり方についての反省がその内外において行なわれているが、そのなかで、いわゆる日本の経営の特質の位置づけについての議論も、かなりいろいろな角度からとりあげられつつあるのが、昨今の状況である。いま、そのことに一般的にふれる場ではないけれども、この成長経済展開過程でもっとも発展したもののひとつがコンピュータリゼーションであるとするならば、そのかわりあいにも多少とも考察を及ぼす必要もあるであろう。しかしながら、そのためには、今日日本の経営の特質がなんであるか、そしてまたコンピュータリゼーションが、経営活動の各側面にどのような変化をもたらしつつあるか、その両面をほりさげてから、その相互のかわりを考察しなければならないはずである。しかも前者はもちろん、後者の経営機械化の影響自体も、拙著『日本経営機械化史』のさきに示した書評にもあるように、十分な解明に至っていない。従って、この小論では、本項にかかげた主題であるコンピュータリゼ

ーションと日本の経営そのものの考察にまで及ぶことは、なお困難である。ただすくなくとも、そのような問題意識で、一步でも近づいたための手がかりにここにふれておきたいと思うのである。

コンピュータリゼーションの展開と日本の経営とのかかわりという場合、きわめて粗雑に考えて、ひとつはマネジメントのプロセスにおけるものと、もうひとつは、その担い手としての人事にかかわるものとの二面があると思われる。前者では、稟議制度とか総務部制など、後者では、終身雇用制とか年功序列などの、昨今日本の経営の典型とされるものとのかかわりということになる。しかしながら、さきにふれたような意味で、いまここで直接にこのような視点からのかかわりには論及することは差控える。すくなくともこのような問題意識のもとで、コンピュータリゼーションの展開のマネジメントのプロセス、人事とのかかわりについて、若干の考察を試みることにする。

(1) コンピュータリゼーションとマネジメントのプロセス

第1に気づかれることは、第3世代以後になって、コンピュータリゼーション担当の部門が、組織のなかで、独立の位置づけがされることが相当に多くなったこととかかわる。『日本経営機械化史』にもみたように、PCS から、EDPSの初期をへて、今日まで機械化の高度な展開ともなっていて、その担当部門の位置づけに数段階があった。当初、経理部門、営業部門、管理部門、総務部門、あるいは企画部門など、業種、業態によって、それぞれ比較的關係深い部門をよりどころとして経営機械化が進められ、今日もなお、そのような位置づけにあるものもすくなくないけれども、経営の諸機能と一般的に關係をもつことから、次第に独立の部門としての位置づけが増加し、さきにふれたように、さらにコンピュータのオペレーション部門と、企画部門とが分化されるに至っている。しかしこれらの実態についてはいくつかの調査もみられるので、一般論をかかげるにとどめる。ここでは、このような時間的推移のもたらしつつある内容的意義に若干ふれておきたいのである。

すなわち、コンピュータによる情報システムの展開が、これまでのタテわりの分担關係のなかに、ヨコつなぎ的なシステム的とりくみの展開をもたらしつつ

つある間に、これらの担当者の世代交替をふくむ思考形式の変化という点である。いいかえると、タテ思考の仕事ぶりを主体とする集団のなかに、意識的、無意識的にシステム思考を本来の業務とする世代の形成ということである。この点は、今日なおコンピュータリゼーションの過度期にあって、現に直接の関係者のなかでも、このような見解をとるものと、それを疑問視するものなど、一致するところまでは至っていない。しかしながら、昭和34年にはじめてコンピューターがわが国企業に導入せられて既に20年近く、その展開過程と、時間の推移と、担い手の世代交替を考えると、ひとつの問題意識としての留意点になるであろう。今後の課題としたい。

第2は、オンライン・リアルタイムのシステム展開、データ・ベースの整備の進展と、ミドルマネジメントにおけるあり方との関係である。ミドルマネジメントのレベルでの管理の高度化を達成しつつあることは、既に知られるとおりである。さらにディスプレイ装置などをふくむ端末機の整備は、内部情報に関する限りでの現場における意思決定、すくなくも定型的レベルのそれについて、その機能発揮を容易にしつつあることは、明らかなるところである。このことは、ライン組織における各レベルの責任権限の内容に、部分的には相当大きな変化をもたらしつつあるはずである。

またコンピュータリゼーションによる情報システムの整備が、主として内部情報に限定せられるとはいえ、ラインのミドルマネジメントに対する情報サービスが向上するにともなって、それは同時にスタッフ機能の強化となり、部分によっては、ラインとスタッフとの関係になんらかの影響を及ぼしつつあることは否定できないであろう。このことは、わが国企業の、経営組織における特徴について、若干の変化が予想せられるように思う。

もっとも、これらの点は、意思決定における人間的要素の位置づけ、組織と人間の関係におけるあり方の問題ともかかわるものがある。コンピュータリゼーションにおける今後検討せられるべき課題のひとつでもある。従ってそれらとの関連のなかで、日本的経営といわれるものとの関係も、今後の課題としたい。

(2) コンピュータリゼーションと人事

終身雇用、年功序列、従って多くの場合のローテーションによる昇進という日本的経営の特徴のひとつといわれるものと、コンピュータの企業への導入ということの間には、これまでの実態の示すところでは、若干の違和感をともなうものがあつた。

戦後 PCS による経営機械化が一般化しはじめたのが昭和20年代後半、そしてコンピュータ導入が昭和30年代なかばから、しかもその後の10数年の間のコンピュータの世代的変化とその間の機械化の幾段階かのレベルアップ、そのようななかで企業経営における情報システム化が展開したのである。それまでの事務室での事務処理と、著しく性質の異なる処理方法が、それぞれのタテ機能から、やがてシステムのな広汎なとりくみに至るプロセスは、その担い手に対する諸条件は、これまでの人事処理と相当異なる扱い方をせざるを得ないものとした。この間の若干の違和感は、経営内におけるいわゆるコンピュータ人口の漸増と、コンピュータ・マインドの侵潤によって、4半世紀をへた今日、かなり軽減せられるようになったといえよう。

しかしながら、わが国企業におけるコンピュータによる本格的な情報システムの構築は、主として昭和40年代、第3世代、3.5世代といわれるようになってからというべく、この時期に経営内におけるコンピュータ人口の増加がとりわけ目立っているのである。その意味では、人事管理なり、人事処遇に関する限りは、日本的経営の特質との具体的なかわりは、むしろこれから10年の課題として提出せられるものかも知れない。さらに、PCS 時代からコンピュータの第1世代、第2世代を通じて、経営機械化を担当した人々と、昭和40年代以後のそれを直接担当しつつある人々との世代交替もまた、いま行なわれている時期である。ここにも、同様の課題があるように思う。

このようなコンピュータリゼーションと人事の問題を通じての日本的経営の問題の考察については、より実態に即しての考察を必要とするので、今後の課題としたい。ただ、上に指摘した後者の問題、すなわち経営機械化4半世紀をへて、担当者の世代交替が現に進行中であることから、その間の実態の推移に多

少とも接触しつづけてきた筆者として、そのための資料はまだ未整備ながら、その世代交替の態様をスケッチ風にここでメモしておきたいのである。

昭和20年代後半から PCS による経営機械化、また昭和30年代半ばから、さきがけてコンピュータ導入による情報システム化にこれらの担当責任者としてとりくんできた人々の、世代交替後の方向に、いくつかのパターンのあることがうかがえるように思う。以下列挙してみよう。

- ①コンピュータ担当者としての経歴のゆえに役員に到達した人々。これは当然に当該企業におけるコンピュータ利用の位置づけとかわっている。このなかには、商法改正による監査役の機能の拡大と関連して、情報システム形成を担当したものがより業務監査に適するとして、常任監査役のポストにあてられるものも、最近目立ちはじめた。
- ②コンピュータ担当専任者のゆえに、コンピュータなりソフトウェア部門を関連部門として独立企業化して、その責任者にあてられたものも、いくつかの事例がみられる。
- ③コンピュータ担当者としてより、より経営的能力を評価して、コンピュータとは直接には関係なく、傍系企業の経営責任者に位置づけられているものもみられる。
- ④長期間コンピュータ部門の責任者としてとりくんだ専門的経歴のゆえに、定年後も顧問なり嘱託として引続き若干のかかわりを持ちながら、ひろく業務その他の分野で何らかの活動をする人々もみられる。
- ⑤コンピュータ部門担当者としての経歴から、専門家として、大学教授のポストを占めつつある人々も、関東、中部、関西、西日本にわたって、その事例がかなりみられる。
- ⑥勤務した企業とは直接に関連をもたずに、その専門的経歴によって、定年後、あるいは定年をまたずして、コンピュータ利用ないしソフトウェア開発を業務とする情報処理産業の企業、あるいはコンサルタント業務に、新たな仕事場を見出している人々もすくないようである。
- ⑦しかしながらまた、経営機械化をある期間担当したあとは、通常の昇進ル

ートにローテーション入りしていった場合も当然に見出される。これはとりわけその企業の経営機械化の特殊性とか、人事政策と関係があるように思われる。

以上は、私見による一応の列挙であるが、これらはいずれも経営機械化の初期からコンピュータ第3世代に至る間のいわばきわめて泥くさい段階からの試行錯誤のなかの展開にとりくんだ人々についての、その後のあり方である。従ってまた、第3世代以後にこの分野に入り、今後10年余のなかで予想せられる人々のあり方は、またかなり異った方向となることが推測せられる。このような実態の推移のなかで、コンピュータリゼーションと人事の諸課題、さらにその日本的経営の特質とのかかわりについて、今後もうすこしほりさげねばならないと思うのである。

本項では、問題の手がかりのいくつかについての私見を示すにとどめた。

6. その他の若干点について

この10年の推移をあとづけるために、まえにのべたように、いくつかの企業の実態にふれる機会をもったなかで、これまでにとりあげた特徴的な点のほかに、なお印象深い若干点について羅列しながらみておくこととする。

第1は、トップマネジメントのフィロソフィとコンピュータリゼーションについてである。東京電力(株)のコンピュータ部門は、総合能力開発本部のなかで担当せられている。これは昭和41年の総合能力開発部からはじまっている。それ以前は企画室での担当であった。先ごろ亡くなられた木川田一隆 前社長の、高度の機械化を人間の問題としてとりあげるという基本的考え方にもとづくもので、その運営方式のなかにもその考え方が透徹しているように見受けられた。また関西電力(株)のコンピュータリゼーションは、芦原義重元社長のパイオニア的考え方と熱心なリーダーシップにささえられて促進せられ、とりわけコスト・コンシャスな積極性が、そのあり方を特徴づけている。ここに例示するまでもなく、コンピュータリゼーションの展開とトップ・マネジメントの

リーダーシップとのかかわりの多いことは、これまで多くの事例によって知られているところであるが、その方向づけなり、特徴的なとりくみにも、すくなくならず影響していて、最近のいくつかの現場との接触において、きわめて印象深かったので、あえてその一部にここにふれたのである。

第2は、海外事業所などとのネットワークとしての情報システムの国際的展開が、ようやく顕著になろうとしている点である。より一般的にいうと TSS による国際情報処理の問題である。昭和51年4月より電算機分野の資本自由化完全実施と、昭和52年度からの連結決算制度の実施との両面から、昭和51年度の新聞報道においても、急速に課題になりはじめた分野である。

この『経営機械化シリーズ』前号No.17（昭和51年刊）のなかの拙稿「オーストラリアにみる情報システムに関する若干問題」で、オーストラリアのわが国の総合商社の情報システムについてのべたところでもこの点に閑説した。現場での事務処理の大きな部分を占める羊毛取引処理のコンピュータ化から、経理その他の一般事務処理への適用をへて、近く本国本社との連結財務諸表作成などの国際情報処理指向のきっかけをここにみようとする事例をのべたのである。

既に昭年46年（株）電通が GE と提携して、GE の MARK I によるタイムシェアリング・サービスを開始していたのが、MARK III によるサービスを、昭和51年1月から（株）電通国際情報サービスを設立して、本格的に国際的な情報ネットワーク・サービスを始めていることは知られておりである。さきにもみた金融機関の1部でも海外部門との情報システムとして現にその利用がみられる。昭和51年になって、あいついで、CDC とかタイムシェア社などアメリカで、GE とならぶ国際情報サービスを業務をする企業で、日本での企画が報ぜられており、国際回線の利用問題が当面の課題となっている。いまこれらの分野の一応の状況をこの時期の特徴的事業のひとつとしてここにのべるにとどめ、これ以上はふれない。

第3は、この10年に、コンピュータが、環境問題、社会開発など、高度成長後の地域なり社会の課題への直接のかかわりをもってきたことも、ここにあげ

ておかなければならないであろう。

テクノロジー・アセスメントが直接にコンピュータ利用とかかわり、また環境影響評価法案が今日問題となっており、また大気汚染管理システムに直接にコンピュータが利用されていることなど、一般的な展開については、ここではふれない。そのなかで、この時期に直接にコンピュータ利用のテスト的な試みを行った事例を示すにとどめる。

兵庫県が環境予測のための「兵庫ダイナミックス」を IBM サイエントフィック・センターと協力して開発発表したのが、昭和48年であり、それ以後さらに環境管理システムにとりこんでいるし、京都市が「土地利用計画」のシステムを、ユニバックの応用ソフトウェア部と協力して開発したのが昭和49年はじめであった。このころ相前後して、各府県、市など地方自治体の総合計画のみなおしが行なわれはじめた時期で、各方面でその前提作業としての、見直し反省のための予測なり、フレームワークの作業が行なわれ、そのための多少とも特徴的なシステム開発が、コンピュータ利用によって、進められた。上記の2事例は、その典型的事例の一部である。いずれもが試行錯誤のなかで、昭和50年代における一層の展開となるであろう。

7. むすび——実態と情報について

昭和40年代の10年間のわが国での主としては企業経営をめぐるコンピュータリゼーションの推移を、以上のように、きわめて粗雑ながら問題意識的にとりあげたのである。『日本経営機械化史』の視点から、今後もその推移を見つづけたいための覚書にとどまらざるを得なかった。

このなかで筆者にとって印象深い点は、この10年間にコンピュータリゼーション担当者の世代交替が顕著に行なわれつつあることである。PCS からの4半世紀を経た現段階の特徴的現象といえよう。既に本文中にふれたように、タテ割機構とその仕事の仕方に育った人々が、きわめて泥くさい、そしてきびしい諸課題にとりくみつつ、しかもコンピュータ・システムの幾段階のレベルアッ

ブをこえて、試行錯誤のなかで、そのシステム化を前進させたといえよう。そして、いま当初からコンピュータを前提にシステム思考を当然として、その活用を前進させることを本来の業務とする人々が、これら先人の業務を継承しつつあるように思う。

高度経済成長時代から、低成長なり安定成長時代への転換が論ぜられ、またそのなかでの日本の経営の特質の位置づけなり、変容のいかんが論ぜられている。そのなかにおけるコンピュータリゼーションも、すくなくないかかわりをもつものである。しかしながら、筆者はいまそれらの一般論をほりさげるまでに至っていない。たださきにのべたその担い手の世代交替のなかに介在する課題には、すくなくない関心をもつのである。次の点のみを指摘して、本小論の結びとする。

コンピュータの導入期をへて第3.5世代といわれる時代となって、情報システムの確立が、業務運営から意思決定まで、その基本的条件であるという考え方が確立するに至っている。この段階に至って思うことは、次の2点についての配慮である。

第1は、実態とその情報化とのギャップという点である。実態の把握は、なんらかのシステム思考によって、現実の混沌を捨象することから可能になるのであり、それによって、より高度の論理的展開も可能になるのであって、必然的につねに実態と情報化との間にギャップのあることの認識である。つねにこのことの原点へかえる謙虚さ、肌をとおしての把握による補いの必要性を指摘しておきたいのである。

第2は、情報システムのブラック・ボックス的部分についてである。高度の情報システム化を進めてゆくなかでは、ある程度ブラック・ボックス的な部分を介在させていかななくては、その前進が難しいように思われる。このような論理的展開の高度化をになうブラック・ボックスについても、同様の問題があるように思う。環境変化のなかでのシステム思考の精緻化に対応して、当然のことながらつねに見直しの配慮が望まれる。

ノルウェーにおけるマイクロ データ ファイル システムの現状と問題点*

能 勢 信 子

は し が き

1970年代から社会会計論における一つのトピックとして登場した研究テーマに、マイクロ データ ファイル システムの問題がある。マイクロ データ ファイル システムは、既存の個別データシステムたとえば人口統計、家計収支調査、租税統計といったマイクロレベルの統計を体系化し、また複数のデータシステムを統合してより総合的な体系をつくり、もって社会経済事象の経験的分析に、それも非集計的なレベルにおける診断に役立てようと意図するものである。現在のところ合衆国の NBER や商務省経済局の研究とノルウェーの中央統計局による研究がもっとも著名であり、その活動の一端をわれわれは IARIW の雑誌論文や NBER の年報に掲載せられる論文から知ることができる。(1) それらによると、マイクロ データ ファイル システムは、個人や個々の事業所など各種のマイクロ経済主体のデータを体系化し、データの反覆的な利用を通

* この研究の一部は昭和52年度文部省科学研究補助金試験研究(2)「経営経済データ・バンクの拡充とデータ・バンク管理システム」(代表者・能勢信子教授)によってなされたものである。

(1) O. Aukrust and S. Nordbotten, "Files of Individual Data and Their Potentials for Social Research", *the Review of Income and Wealth*, series 19, Nr. 2, June 1973, pp. 189-202. E.C. Budd, "The Creation of a Microdata File for Estimations the Size Distribution of Income", the same review, series 17. Nr. 4, Dec., 1971, pp. 317-34. R. Ruggles and N. Ruggles, "The Role of Microdata in the National Economic and Social Accounts", the same review, series 21, Nr.2. Jun., 1975, pp. 203-216. R. Ruggles and N. Ruggles, "A Strategy for Merging and Matching Microdata Sets", Progress Report on NSF Grant Project Gs-33956, *Annals of Economic and Social Measurement*, vol. 3, No. 2, April 1974, pp. 353-72.

して多面的な社会経済構造の明細を示す数値を提供し、従来の集計タイプの経済計算や伝統的手法による社会統計では把握しがたい問題の分析に役立っているようである。

ここに紹介するノルウェー中央統計局のマイクロ データ ファイル システムは、そのスムーズな発展、システムの整備の完全性、最終利用および中間利用面での効果などによって、いまでは古典的なデータ ファイル システムの一つとせられている。以下にこのシステムの構成の骨子を紹介し問題点を要約して、マイクロ データ ファイル システム一般の、またわれわれの企業データ ファイル システムの発達のための示唆を得たいとおもう。

2. ノルウェーのマイクロ データ ファイル システム における基本的な発想

まずノルウェー中央統計局のマイクロ データ ファイル システムは、つぎのような発想にもとづいている。

それはオークラストとノルドボテンによれば⁽²⁾ 個人であると企業であるとを問わず個々の経済主体を単位とする原始記録が体系的に蒐集せられ、また系統立って維持・貯蔵せられているならば、データを検索しそれを加工および再加工して種々の目的のために編集し各種の分析に充当することができるという考えである。この検索と再加工の可能なデータ蒐集を目的とするという発想は、マイクロ データ ファイル システムと既存の伝統的な統計蒐集法たとえばセンサス方式との大きい相異点をなしている。

次に注目すべきは、マイクロ データ ファイル システムがいくつかの基礎条件なくしては成立しがたいという事情である。まず第一に、このシステムにとって重要な存在であるのが個々のデータに付けられた同定装置 identifier である。個々のデータは必要な特徴ないし属性に即して分類せられ、加工され、また他の目的のために分類せられた他のマイクロ データとリンクせられる。この一連の過程において検索とデータの反覆利用が生じるが、このためにはすべて

(2) Aukrust and Nordbotten, "Files of Individual Data," *op. cit.*

のマイクロ データになんらかの識別表示表あるいは同定タグを付けていることが必要不可欠な条件である。人口に例をとると、各人の出生の記録に対してその後の個人事情の変化たとえば結婚、離婚、死亡が記入され、そしてこれがすべて同一単位の帰属するという同定標識、いわゆる背番号をつけていることが必要なのである。伝統的な人口センサスと異なるマイクロ データ ファイル システムの特徴は、個々の原始データに同定装置としての標識がつけられているかどうかにあるということができる。

第二に、このデータ ファイル システムの運営を可能にする技術的な条件が、EDP システムである。1958年ノルウェーに最初の EDP システムが導入せられ、統計の大規模生産の有利性という中央統計局の要請にこたえて、これが大量の統計情報の供給に役立って来た⁽³⁾。ちなみにノルウェーの全人口は約 400万人であり、マイクロ データ ファイル システムはその人口の全体の記録をカバーしている。これだけのマイクロ データを記入し維持保存し、また集計を行う作業は、EDP システムを活用する以外には行い得ないのである。逆にいえば、一国の統計計算機構に存在する EDP の計算能力の最大限度がマイクロ データ ファイル システムのあり方を技術的に決定するといつて差支えないであろう。

第三に、マイクロ データ ファイル システムにとって必要であるのは、個々のそして多種類の社会経済情報を包括し、これを継続的に分類・記入した蓄積し、系統別に整理されたデータ相互のリンクを保持して行く上で、システム全体を通して共通する概念群の一般的な枠組を持つことである。ノルウェーではマイクロ データ ファイル システムは、伝統的な手法による統計情報とともに人口統計、経済統計、社会統計の作成にたいして基礎資料を提供し、また各種の経済分析および社会人口分析に明細情報を与えることを意図しており、したがって当該システムの対象範囲は、多種類におよんでいる。それ故多種類の情報を秩序付けまた相互の関連を理論的に確立する枠組が必要であるこ

(3) Central Bureau of Statistics of Norway, *The Organization of the Central Bureau of Statistics of Norway*, 1977, p. 1.

とは明らかである。

ノルウェーの場合、こうした概念構造は、中央統計局の所長でありまた理論的指導者であるオークラスト博士の社会会計公理論によって与えられている。オークラスト博士の公理体系⁽⁴⁾は、その既発表分によると、エコ サークシステムを成立させる一連の基礎概念群から成立している。すなわち個々の経済対象（実物と請求権の双方）、時間、個々の取引者（部門）という基礎範疇と、範疇相互の関係（たとえば所有とその変更、生産活動、消費活動など）についての公準、対象の価値についての定義がそれで、マイクロ レベルの勘定とマクロ レベルの国民勘定を活動別、部門別にリンク付ける工夫が抽象的に与えられている。この公理系は SNA のようなマクロ レベルの計算体系とマイクロ情報を、また個別マイクロ情報を相互に論理的に接合する機能を遂行することができる。オークラストは、論文では市場的な経済対象にとどまらず、市場外的な対象の体系的取扱いについても公理化が必要であることを示唆しているにすぎない⁽⁵⁾が、後者にも何らかの一般的枠組をすでに設定していることが想像できる。概念の枠組を持つことが、マイクロ情報の蒐集を体系的に行い、その質を標準化し、分類・配列を合理的に行うために有用であることは、明らかである。

第四に、マイクロ データ ファイル システムを実際に運営して情報の生産・保蔵と利用者に対する供給を円滑に行うためには、データの蒐集・記録、表式化、計算、整理・収納についての慣行と規則を体系化しておく必要がある。たとえば、調査表をチェックレコーディングする際の規則、これをさらに EDP 用に変換する際の規則、データ利用側の要求に対応してマイクロ データを編集する編集規則がそれである。こうした一連の規則の体系は、マイクロ情報体系の円滑な運営を果すために考案される情報処理の手段体系にほかならない。

第五に、マイクロ データ ファイル システムは、ファイリング機能にかか

(4) O. Aukrust, "An Axiomatic Approach to National Accounting, An Outline", *the Review of Income and Wealth*, Series 12, Nr. 3, Sept., 1966, pp. 179-90.

ちなみに該論文はオークラスト博士が1955年に書いた博士論文 *Nasjonalregns-kap., Teoretiske prinsipper* の英文サマリーである。

(5) O. Aukrust, *ibid.*, p. 189.

わる保蔵手段を必要とする。このためノルウェーの当該システムは、データファイルのライブラリーを持ち、検索の要請に応じている。そこにファイルせられたデータは、特徴ないし属性を明示して内容を示す。マイクロデータファイルシステムの保蔵機能は、統計数値だけを対象とするのではなく、計算目的に適したプログラムの選択を可能にするプログラムドキュメンテーションライブラリー、情報を利用する場合のテーマやプロジェクトの類型を保存するプロジェクトドキュメンテーションライブラリーをも含んでいる。こうした一群のライブラリーが当該システムの保蔵機能を遂行するのである。

最後に、マイクロデータファイルシステムの存在の前提となるものに、一定の社会的条件がある。これは、以上述べた諸条件と異なり当該データファイルシステムそのものに直結してはいないが、間接にこの存在を支える社会的背景にはかならない。当該システムにあってはマイクロデータが記入されそれが継続的に維持更新せられるのであるが、このことへの協力が私的個人および私企業に対して法律的に義務付けられている。しかし私的記録を継続して追求せられることは、プライバシーの見地から問題となる可能性がある。事実、国民背番号制と併せ情報担当者の守秘義務がかつて話題となったことがある。しかしその際にもノルウェーのこの制度は影響を受けなかった。現在、ノルウェー国民は、戸籍記録の発生と同時に当該システムの調査票に記入し、事情の変化すなわち結婚、離婚、死亡がその発生の都度記入せられる。これが1964年の法律によって義務付けられているのである。企業についても同様で、その継続期間中は、収益、生産、費用、利益に関する報告を義務付けられている。この法的強制を甘受する背景には、マイクロデータの提供について社会的合意があり、とくにプライバシーの侵害という欠点にも増してマイクロデータファイルシステムのメリットが国民に評価せられていることを物語っている。ノルウェーの中央統計局は、1876年の創立が示すように、その歴史が古く、その計算と分析に対する信頼が一国の社会的合意を達成する上で、大きい要素となっていることは、想像に難くない。

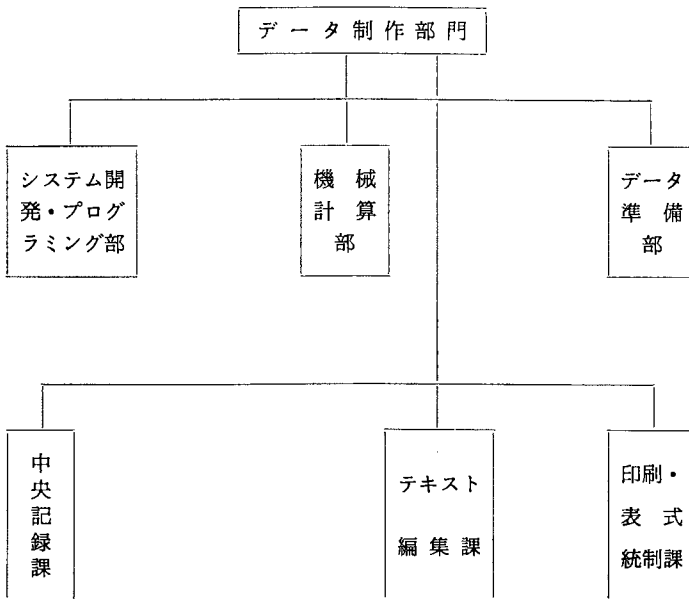
以上の諸条件が、ノルウェーのマイクロデータファイルシステムの発想

を実現せしめる契機となっているのである。

3. ミクロ データ ファイル システムの生産機構

オークラスト博士を長とするノルウェー中央統計局は、ノルウェー政府の一局である。その組織は、データ制作部門、統計部門、研究調査部門の三本立てになっており、それぞれ統計資料の生産・基礎加工、各種基礎統計の編集・刊行、国民勘定を含む経済分析と社会人口分析という機能を分業で遂行している。個々の部門内ではより細かい分業体制がしかれ、たとえばデータ制作部門においてはデータ準備部、システム開発・プログラミング部、機械計算部という三位一体的な分業にもとづく協業を可能ならしめている。中央統計局の組織全体が、各種統計の生産、その統合、用語、概念ならびに分類におけるコーディネーションを達成することを目標としており⁽⁶⁾、こうした考え方自体がミ

第1表 中央統計局データ制作部門の組織図



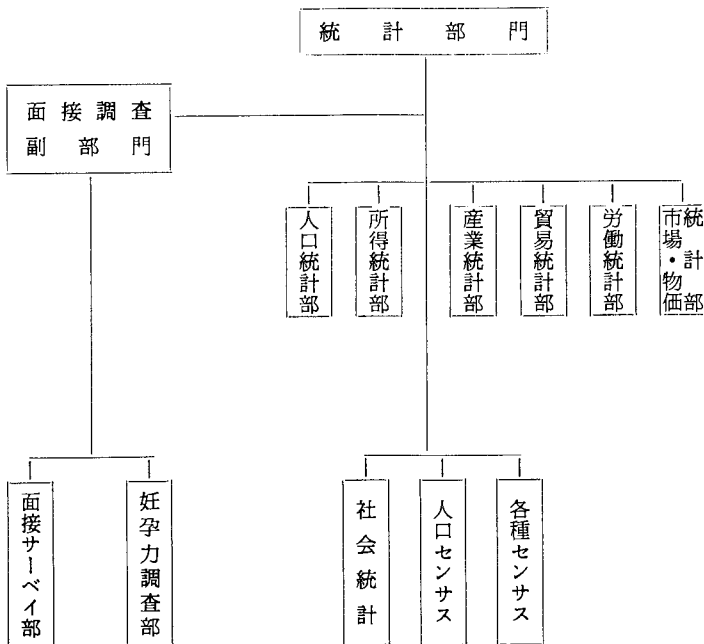
クロ データ ファイル システムの有効な発達を育成する上で役立っていることは、容易に想像することができるのである。

以下にデータ制作部門、統計部門および研究調査部の組織図⁽⁷⁾を要約し、マイクロデータの生産と保蔵を遂行する主体の位置付けをしてみよう。

前頁の第1表は、データ制作部門の組織図である。

データ制作部門は、マイクロデータファイルシステムを提供しシステムの維持・保蔵機能を遂行する。この成果はユーザーである統計部門、および研究調査部門ならびに外部に供給せられる。制作部門は、他方、統計部門から来る調査票をインプットする。

第2表 中央統計局統計部門の組織図



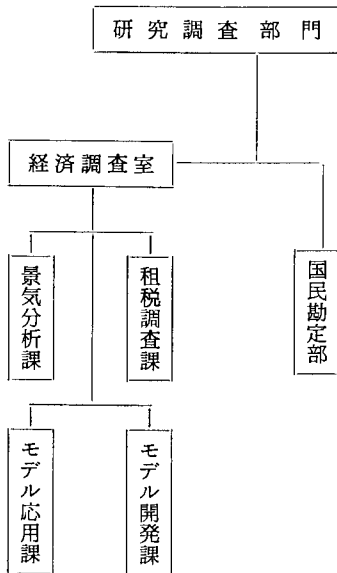
(6) Central Bureau of Statistics of Norway, *The Organisation*, op. cit., p. 1.

(7) Central Bureau of Statistics of Norway, *A Short Note on the Central Bureau of Statistics of Norway and its Research Department*, 1977, pp. 1-2.

第2表および第3表は、統計部門 および 研究調査部門の 組織図の要約である。

研究調査に関する独立の部が、いま一つある。これは、社会人口調査部で、研究調査部門と同じくマイクロ データ ファイル システムのユーザーである。ちなみに統計部門は伝統的な統計を編集・刊行するかたわら当該システムのユーザーであり、同時に調査表を当該部門に提供する供給部門という多くの機能を果している。

第3表 中央統計局研究調査部門の組織図



以上を果している。

以上の部門のほか一般管理のための部門が置かれ、そしてこれらの全体を、局長が総括している。

つぎにマイクロ データのファイルが実際に生産され、体系的に保蔵される活動を制作部門を中心として考えよう。

まず原始資料である調査表が、統計部門から供給せられる。制作部門のデータ準備部においてこれが選別ルーティンに従って選別せられ分類されて機械計算用にさらに組みかえられる。ついで作表のために、必要とする表が指定せられる。

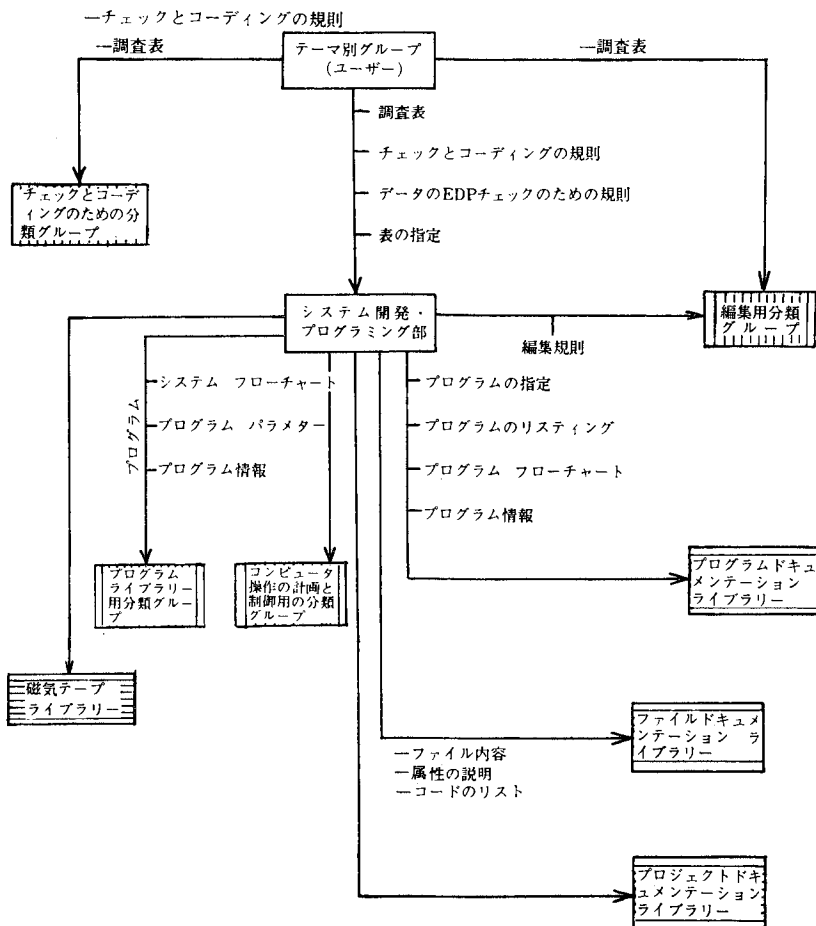
ついで作表のために、必要とする表が指定せられる。

つぎにシステム開発・プログラミング部は、利用可能なプログラムのうち必要なプログラムを選択し、特定のシステムの設計を行う。この部の機能は、マイクロ データ ファイル システム運営の中心的役割を果すところであり、多くの面にわたっている。第一に、マイクロ データ ファイル システムのユーザーの要請を充しつつ慣行的な編集規則にしたがって統計表を編集する機能、第二に機械計算部門に最も有効なシステムを設計する機能、第三に、マイクロ データ ファイルを作成・維持・保蔵する機能、第四にシステム開発機能および

びプログラム情報の研究・蒐集機能がそれである。他方、機械計算部は、データ準備部で用意した資料をシステム開発・プログラミング部の指定にしたがって機械計算の遂行にあたる。

以上の活動につけ加えるべきは、すでに指摘したデータおよび知的情報の保蔵活動である。データの検索とプログラム等知的情報の検索は、このシステム

第4表 ミクロデータファイリング活動の流れ



にとって中心的な存在の一つであるから、みぎの保蔵活動は不可欠である。前節で述べたように、情報ライブラリーが制作部門に設けられており、プロジェクトライブラリー、プログラムドキュメンテーションライブラリー、ファイルドキュメンテーションライブラリー、磁気テープライブラリー等が、それぞれの保蔵機能を遂行している。

以上、マイクロデータのファイリングという情報の生産フローとストック形成活動を、中央統計局の説明によって図表化すると、第4表のごとくである。

第4表は、該システムの生産・保蔵活動に関係する部の中でもシステム開発・プログラミング部を中心に描かれている。□印は、制作部門外部のユーザーである部門と当該システム開発・プログラミング部門である。データ準備部は、フローチャートの外側に説明しているが、機械計算部はここでは明示的に示していない。■印は、各部の特徴的な活動を示している。つぎに▣印は、情報ストック(データおよび知的情報のそれ)が保蔵されるライブラリーである。

なほ全体を結ぶ矢印は、いくつかの指定された径路をへてマクロデータが生産され保蔵される活動を、部から部へまた部からライブラリーへ流れる全体的な流れとして示している。

4. ノルウェーのマイクロデータファイルシステムの意義：

教訓と問題点

第1節において当該データファイルシステムの発想法とそれを支える諸条件を、第2節において当該システム関係部門の活動状況を要約したのであるが、これから考えることは、まずノルウェーのマイクロデータファイルシステムがきわめて体系的に作られているということである。

このシステムの体系性は、イ、オークラスト博士の公理論的発想にしたがってデータファイルとデータのリンクの概念的な枠組が与えられ、ロ、情報の生産と利用・再加工部門の間に有機的な結合があり、ハ、当該データファイルシステムの生産・維持・保蔵・利用において一連の慣行・規則の体系が設けられ、また一群の保蔵施設をもち、ハ、EDPシステムの高度利用が

行われている等の点から容易に想像しうるところである。そして、当該システムの提供するマイクロデータの集合は、データの質の標準化についての保障と、データのもつリンク可能性のゆえに、分析上有効であると考えられる。

いま一つは、ノルウェーのシステムが対象に対して直接的かつ全面的に把握する点である。ファイルせられるデータは、個々の個人と企業のレベルから直接また全面的に蒐集せられ、いわば直接性をもっている。そして個々の調査表が同定標識をつけているために、完全なファイリングを集計、再加工、リンクを可能ならしめる。マイクロレベルの数値から集計値までの直接的計算が該システムでは一貫的に保障せられているということは、みぎのシステムの利用者にとって——とくに経済構造の分析や地域分析あるいは福祉、労働力移動、犯罪などの社会人口分析を志す利用者にとって、しごく好都合である。

以上の長所から見れば、ノルウェーのこのシステムがマイクロデータファイルシステムの古典的また理想的な姿であると誰しもが考えるであろう。以下にこのノルウェー方式が、一般的に適用可能であるかどうかを考えよう。

まずイのマイクロデータの蒐集とリンク付けについて概念的な枠組が理論的に与えられるということ、および、ロのマイクロデータ制作機関の活動遂行上合目的な組織が運営せられ、またハのデータ蒐集に関する手段体系としての慣行・規制の類が制度化されることは、この有効な情報保蔵装置の利用とあいまって、情報ファイルシステムの合理的な発展を実現する上で不可欠である。その限り以上4点は、いかなる国のマイクロデータファイルシステムにとっても有用であり、システムの運営上留意すべき戦略目標といえる。それ故、ノルウェー方式のみぎの特徴は、他山の石として参考にすべきであろう。

EDPシステムと情報ファイルシステムを有効かつ高度に利用することについてもまったく同じことがいえる。

ところで問題は、ノルウェーのシステムが、マイクロデータ蒐集の直接方式をとる点にある。そしてこれと不可分に考えらるべき点が、マイクロデータにたいする同定標識の問題である。国民背番号制を導入し、もって国民に対して直接個人記録の提出を法的に義務付けるということがこの背後に存在するこ

と、そしてこれには社会的合意がないかぎり困難であることは明かである。ノルウェーの場合、中央統計局は統計生産に際して集権的な位置をもっているが、その機能および守秘義務の遂行に対する国民の信頼は大きい。しかしこれらを全面的に期待できないとき、プライバシー擁護論が、背番号方式のメリットを粉砕することは明かである。

さらに考えるべき点は、ノルウェーが人口的には小国であることである。これは EDP の技術的限界以下の計算量であることを意味している。しかし一億を越える規模の国では、このことは望み難い。

そこで、ノルウェー型マイクロ データ ファイル システムの教訓としてわれわれが受取るべきは、一般的概念構造を持つこと、データの質の標準化と、計算組織の効率的な運用とくにファイリングの手段体系である慣行と規則の整備、ライブラリー群の整備の諸点であり、これをノルウェー型ではなく間接的また分権的なマイクロ データ ファイル システムのなかで実現して行くことにあると思われる。一般的概念構造そのものではないが、マクロ的な SNA 型の部門とマイクロ単位をリンクする構想を提唱するラッグルス夫妻の試みや⁽⁸⁾分権的なデータ ファイル システムを総合した対応方式の考え⁽⁹⁾はいづれも分権的かつ間接的なマイクロ データ ファイル システムを進めて行く上で考慮すべき必要なステップと考えられる。

最後にわれわれの大学の企業データ ファイル システムからみて示唆をうける点は、データの質の標準化の問題である。勿論われわれの企業データ ファイル システムは、構想と規模のすべてにおいてノルウェー方式と余りにも異っている。しかしデータの質の維持という現点からだけ考えると、企業データ ファイル システムは、いくつかの問題がある。その一つはまず国際的標

(8) R. Ruggles and N. Ruggles, "the Role of Microdata in the National Economic and social Accounts", *op. cit.*

(9) R. Ruggles and N. Ruggles, "The Role of Microdata", *ibid.* これをラッグルス夫妻は、総合的マッチングと呼ぶ。また R. Ruggles and N. Ruggles, "A Strategy for Merging and Matching", *op. cit.* Budd, "the Creation of a Microdata File for estimating the Size Distribution of Income", *op. cit.*

準化を考慮すべきことであり、いま一つは国内企業のデータについても各企業について、またその時間比較についての標準化が必要であるということである。第一の問題は、最近、国連の UNID が提唱する多国籍企業の情報の国際化⁽¹⁰⁾ の問題であり、われわれにとってはこの慣行の確立がなされたならばこれに従ってデータの一律改訂をすることが必要である。第二の問題は、商法の改正ごとに企業情報の作成方式が変るということであり⁽¹¹⁾、昨年の財務諸表の連結問題やまた近い将来にインフレ会計が導入されることをも考えると、企業データに作成年次と測定慣行の「日付け」をする必要があると思われる。しかしこれらについては、改めて詳しく論じたい。

(10) UN. Development of an Information System Commission on Transnational Corporations, March, (E/C.10/28), 1977.

(11) わが商法は昭和25年、37年、昭和45年に大改正されその都度株式会社の財務諸表計算規則が改正せられている。

大阪市における EDPS の分布について

都 藤 希八郎

はじめに

経営・管理において、EDPSの安全性が現実の諸問題によって最近急速に強調されるようになってきている。

本シリーズにおいても前17において「EDP部門の防災」としてまとめ、その中で「分散」が安全対策の最も基本的な考え方であることを強調した。

ここでは、個々のEDPSの安全性ばかりでなく地域全体としての安全性の立場から都市におけるEDPSの分布状態について、関西の中心都市である大阪の現状の一部を明らかにし、広域災害時におけるEDPSの同時被災による企業活動の停滞防止への一提案とした。

1. 調査方法

1.1 調査地区の選定

大阪市は、東京に相対する全ゆる面での中心都市であるのは当然であるが、

- (a) 全国的また近畿地区で見ても、EDPSが各業種・規模にわたり目立って多数設置されている。
- (b) 東京に較べ近代においては地震には比較的安全とされているが、その地震を含め火災・台風・公害とEDPSの広域防災を必要とする災害を全て予定せねばならない、などの理由からまず最初に対象とした。

1.2 調査の方法

- (a) 資料のEDPSは、情報処理学会編「ユーザー調査年報、1976年版」より抽出し、それぞれの目的に応じて使用した。
- (b) EDPSの分布図は、少しでも実際に近い状態を把握できるように1/27,000の地図上に「町または丁目」を最小単位として位置を記入した。

(c) EDPS の規模は、若干の問題はあるが、識別しやすいよう次の規準によって大・中・小の3種に区分した。

大規模；大型機または中型機を複数台設置し、適用業務を巾広く実施しているとみられるもの。

中規模；中型機または小型機を複数台設置し、一部門を形成しているとみられるもの。

小規模；小型機以下を設置し、小組織か部門がないとみられるもの。

2. 広域災害の仮定

ここで検討する広域災害は前記のうち次の3つであり、その原因と直接的な防災方法の概要を表示すると第1表のとおりである。

第1表 広域災害とEDPSの直接的防災方法

種 類	原 因		防 災 方 法
	直 接	誘発原因	
地 震	震 動	火 災	防火区画，耐震・耐火構造建物
		津 波	防潮扉，EDPS の上層階設置
		停 電	複数系統受電，自家発電
火 災	熱	類 焼	防火区画，耐火構造建物
		停 電	複数系統受電，自家発電
台 風	風 圧	高 潮	防潮扉，EDPS の上層階設置
		(風 圧)	耐震構造建物

このような災害の防止策は、個々の EDPS 設置の建物では、幸いに表中にも見られるように、ほぼ同じ方法でよい場合が多い。また個々の防災方法を基本に、都市防災計画（例えば、防火地域、空地々区、緑地帯、防火建築帯、防潮堤計画、送電計画）を実施することによって、多くの広域災害はある程度防止できる。ただ火災については、対策を実施しても十分な効果を期待できない

場合も考えられる。大阪市について見ても、明治42年7月 111,365戸、明治45年1月 5,268戸、昭和20年戦災、と3回の記録がある。現在の都市は全体の耐火度が高くなっているとはいえ、昭和51年10月酒田市で約 1,100戸を焼失した例もあり、一度大火となれば1分間数百米の速度で延焼し飛火も数百米から1,000米にも達することもあるので、可燃物の多い大都市大阪としては一層の対策が望まれる。そこで以下、広域災害として火災を第一においた考察を行うことにしたい。

3. EDPS 分布の現状と考察

3.1 業種別分布について

(a) 全国的にみた分布について

前記年報による全国9,577個所のうち、大阪市内には888個所の設置個所があり約 9.3%に当る。これを業種別の設置総数から見ると、鉱業・建設16.3%、繊維・紙・パルプ15.8%、化学工業18.3%、商業15.6%、の比率が特に大きい。なお、オンラインは端末機使用の場合も含めて 112個所であり、また近隣都市に大規模を含む多数の EDPS が設置されている(第2表)。すなわち、以上の4業種は全国的に見てある程度集中していると見られ防災面からの対策を考える必要がある。

(b) 市内における分布について

市内 888個所の設置個所のうち、業種別に見て多いのは、繊維・紙・パルプ50個所5.6%、化学工業80個所9.1%、商業 298個所33.6%、ソフトウェア開発・計算センター81個所 9.1%、である。なお、これを規模別に見ると、大規模92個所10.4%、中規模313個所35.3%、小規模483個所54.4%である(第2表)。以上の4業種は、市内における数量的比率も高く、(a)からも見られるように全国的にも集中していると見られるので、何らかの防災対策を考慮すべきである。

3.2 平面的分布について

3.2.1 行政区別分布について

市内の設置個所数で目立つものを行政区別に見ると、東区 243個所27.4%、西区105個所11.8%、北区158個所17.8%である。また規模別に分けて見ると、(大規模)東区37個所40.2%、北区17個所18.5%、(中規模)東区112個所35.8%、西区34個所10.9%、北区62個所19.8%、(小規模)東区 94個所 19.5%、西区63個所13.0%、北区79個所16.4%である。すなわち「東・西・南・北」の4区に、大規模 73.9%、中規模 71.3%と主要な EDPS である大・中規模の71.9%が集中しており、小規模も56.7%が集中し、総数においても63.6%が集中している(第3表)。

これら4区に直接的・広域的な EDPS の防災対策の重点を指向すれば極めて有効なことがわかる。

3.2.2 業種別分布について

平面分布を詳細に調査するため、今回は次の4業種を対象とし地図上において検討した。

(a) 化学工業(第4表, 第1図)

化学工業は、火災発生の危険度が高く市内でも前記のとおり多数設置されているのでとりあげたが、結果としては製造工場ではなく営業関係と見られるものが多く、この点危険度は低いといえる。

先ず第4表で見ると、化学工業では、東区27個所33.8%が集中している。これを第1図によって測定すると、最も高い密度のA地区では、 $1,418m \times 675m$ の台形の地区内に、大規模2、中規模13、小規模5、計20個所のEDPSが集中しており、平均 $197m$ 平方に1個所の割合になる。このうち11個所が道修町の薬品関係である。次にやや高密度のB地区で見ても $2,200m \times 2,160m$ のほぼ正方形の地区内に、大規模3、中規模22、小規模13、計38個所があり、平均 $354m$ 平方に1箇所の割合になり、化学工業全体の47.5%が集中していることが明らかである。

これらの地区は、耐火建築群・河川・道路の面から防災度は高いと見てよいが、ある程度以上の災害なかでも火災を伴う場合においては、A地区の危険度は極めて高くB地区でも延焼の危険はあり広域災害対策が必要であろう。

(b) 金融・保険・証券, 官庁・政府機関, 地方公共団体 (第4表, 第2図)

これらの業種は, 政治・経済・情報処理の中心として社会的影響は極めて大きいのでとりあげた。

先ず第4表によると, 業種別総数から見て, (金融・保険・証券) 東・西区15個所55.6%, (官庁・政府機関) 東区5個所31.3%, (地方公共団体) 東・西区6個所60%, に集中している。これを第2図によって3業種を一括して測定すると, 最も集中しているA地区では, $1,418m \times 540m$ 内に, 大規模5, 中規模4, 小規模1, 計10個所となり $277m$ 平方に1個所の分布となる。比較的集中しているB地区では, $2,498m \times 2,727m$ 内に, 大規模11, 中規模4, 小規模3, 計18個所で $615m$ 平方に1個所の分布でやや分散しているといえる。

また, これらを業種別に検討してみると次のことがいえる。「金融・保険・証券」では13個所が大規模でオンラインのセンターとなっている。そこで第2図で大規模EDPS間の距離の概数を出したところ, $297m, 405m, 459m, 513m, 608m, 648m, 905m, 905m, 1242m$, となり, 一部距離をおいたものもあるが, 全体的には集中しているとみられ危険性が含まれている。「官庁・政府機関」では大規模EDPS12個所中10個所は電々公社関係で, そのうち6個所はデータ通信局で分散されているものが多い。ただ「官庁・政府機関」「地方公共団体」では, EDPSが各4個所ずつ大手前之町の官庁街に集中されており, 行政上の莫大な情報が蓄積される場合には分散の方法を考慮する必要がある。

お わ り に

本稿では, EDPSの広域防災を分布状態で調査し, 分散という見地から部分的に検討したが, 今後当然, (a) 都市防災計画(前記)の実施程度を考慮に入れ, (b) EDPS設置の建物の周辺, 建物自体の安全性および内部における危険分散の実態など個々の防災度を調査のうえ, 診断を行ない結論に導くようにすべきであろう。

概略的ではあるが, 広域災害に対し比較的安全度の高い地区と建物に設置さ

れることの多い大規模・中規模 EDPS の場合も、激甚な災害では企業に致命的な打撃を与えることになる危険性は本調査でも一部において伺えるし、比較的安全度の低い地区と建物に設置されることが多い小規模 EDPS の場合、個々の企業の EDPS への依存度がむしろ高いこともあり得るので、企業経営への影響は矢張大きいといえる。

しかし、EDPS の広域防災にとって有利なことは、例え建物・部門機構・EDPS 自体の分散が困難な場合でも、情報自体を MT・DISC などによって、支店間・保管業者・別建物への分散は容易に行なうことができ、各事業所がこの措置を取ることによって、EDPS の壊滅的な被害だけは防止できることである。

第2表 大阪市における EDPS の業種別・規模別分布

業 種	対 象		大 阪 市				全 国	オンライン
	規 模		大規模	中規模	小規模	計	計	
鉱業・建設	4	13	14	31	190	5		
ガラス・土石製品	1	1	7	9	143	1		
水産・食品	1	8	8	17	321	4		
繊維・紙・パルプ	8	15	27	50	316	8		
化学工業	5	38	37	80	438	11		
石油・石炭製品・ゴム製品	1	5	0	6	103	2		
鉄鋼・非鉄金属・金属製品	2	13	25	40	416	0		
機械・精密機械	4	15	21	40	440	3		
電気機器	3	4	12	19	388	4		
輸送用機器	1	6	2	9	272	0		
その他製造	1	9	31	41	290	0		
電力・ガス	1	1	1	3	50	1		
運輸・倉庫・不動産	3	10	21	34	331	2		
商業	7	91	200	298	1,910	26		
金融・保険・証券	14	8	5	27	636	14		
官庁・政府機関	12	3	1	16	355	6		
地方公共団体	2	5	3	10	386	3		
学校・研究所・病院	0	7	12	19	688	0		
組合・諸団体	1	7	4	12	421	1		
通信・出版・サービス・その他	1	12	17	30	410	7		
ソフトウェア開発・計算センター	15	36	30	81	821	9		
電子計算機メーカー関係	5	6	5	16	252	5		
計	92	313	483	888	9,577	112		

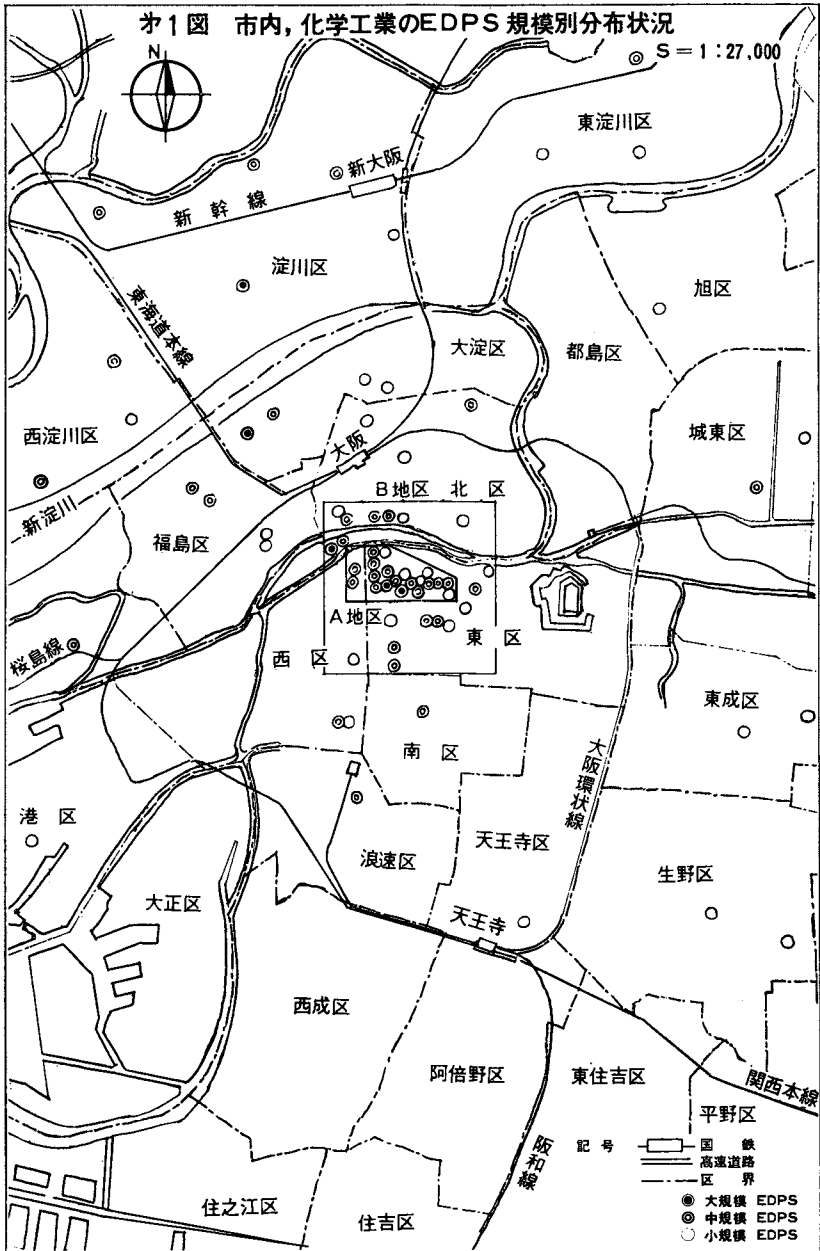
第3表 大阪市における EDPS の行政区別・規模別分布

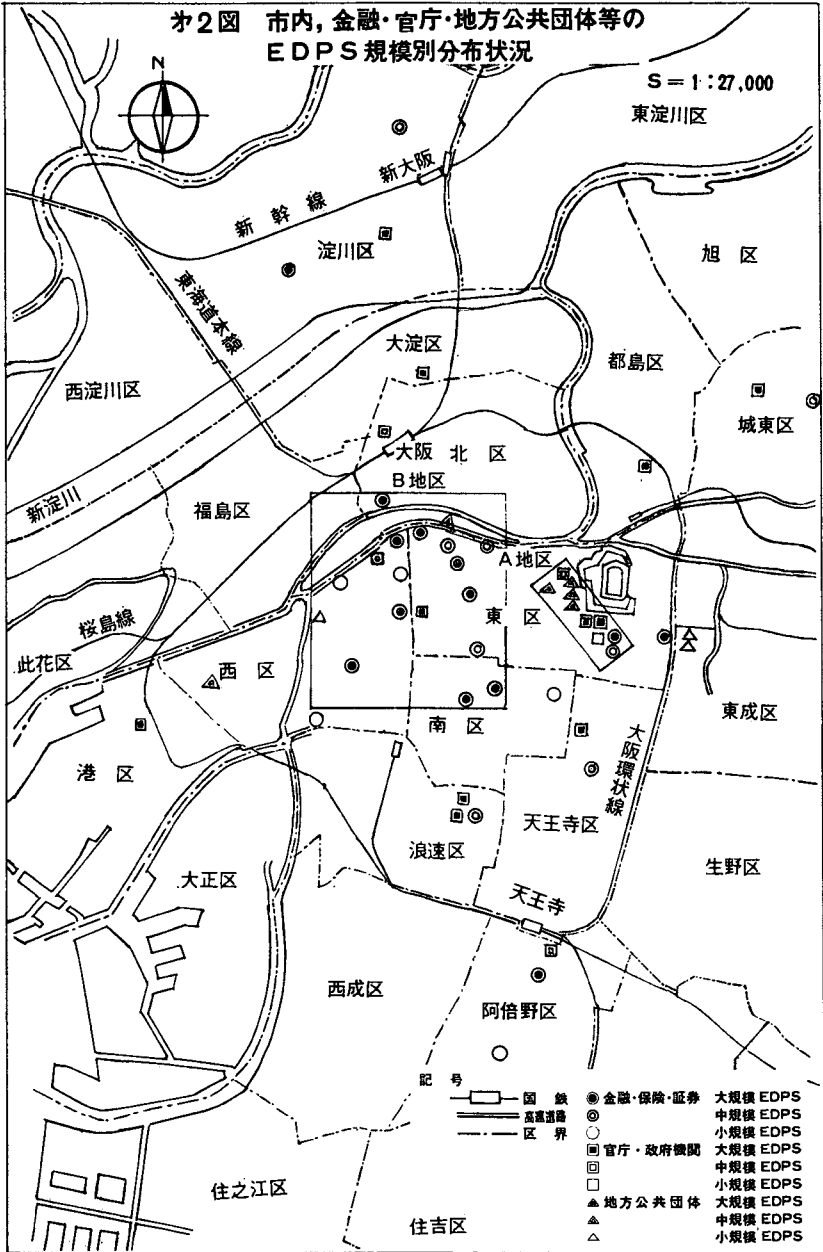
区名	大規模	中規模	小規模	計
東	37	112	94	243
西	8	34	63	105
南	6	15	38	59
北	17	62	79	158
都島	2	1	2	5
旭	0	3	2	5
城東	2	7	11	20
鶴見	0	2	5	7
東成	1	5	12	18
生野	0	2	10	12
天王寺	2	3	22	27
住之江	0	2	4	6
住吉	0	5	9	14
東住吉	0	1	10	11
平野	1	1	3	5
阿倍野	2	2	3	7
西成	1	3	10	14
大正	0	2	4	6
浪速	3	4	17	24
港	1	1	4	6
此花	1	5	6	12
福島	1	9	18	28
大淀	3	8	12	23
西淀川	0	7	16	23
淀川	4	6	10	20
東淀川	0	11	19	30

第4表 大阪市におけるEDPSの特定業種の行政区別・規模別分布

区名	規模	化学工業		金融・保険・証券		官庁・政府機関		地方公共団体	
東	大	2	27	5	9	3	5	2	4
	中	16		4		1		2	
	小	9		0		1		0	
西	大	0	5	3	6	1	1	0	2
	中	3		0		0		1	
	小	2		3		0		1	
南	大	0	1	2	3	0	0	0	0
	中	1		0		0		0	
	小	0		1		0		0	
北	大	1	11	1	1	0	1	0	1
	中	5		0		1		1	
	小	5		0		0		0	
都島	大	0	0	0	0	1	1	0	0
	中	0		0		0		0	
	小	0		0		0		0	
旭	大	0	1	0	0	0	0	0	0
	中	0		0		0		0	
	小	1		0		0		0	
城東	大	0	2	0	1	1	1	0	0
	中	1		1		0		0	
	小	1		0		0		0	
鶴見	大	0	1	0	0	0	0	0	0
	中	0		0		0		0	
	小	1		0		0		0	
東成	大	0	2	0	0	0	0	0	2
	中	0		0		0		0	
	小	2		0		0		2	
生野	大	0	2	0	0	0	0	0	0
	中	0		0		0		0	
	小	2		0		0		0	
天王寺	大	0	1	0	1	1	1	0	0
	中	0		1		0		0	
	小	1		0		0		0	
住之江	大	0	0	0	0	0	0	0	0
	中	0		0		0		0	
	小	0		0		0		0	
住吉	大	0	0	0	0	0	0	0	1
	中	0		0		0		0	
	小	0		0		0		0	
東住吉	大	0	0	0	0	0	0	0	0
	中	0		0		0		0	
	小	0		0		0		0	
平野	大	0	2	1	1	0	0	0	0
	中	0		0		0		0	
	小	2		0		0		0	

区名	規模	化学工業		金融・保険・証券		官庁・政府機関		地方公共団体	
阿倍野	大	0		1		0		0	
	中	0	0	0	2	1	1	0	0
	小	0		1		0		0	
西成	大	0		0		0		0	
	中	0	0	0	0	0	0	0	0
	小	0		0		0		0	
大正	大	0		0		0		0	
	中	0	0	0	0	0	0	0	0
	小	0		0		0		0	
浪速	大	0		0		2		0	
	中	1	1	1	1	0	2	0	0
	小	0		0		0		0	
港	大	0		0		1		0	
	中	0	1	0	0	0	1	0	0
	小	1		0		0		0	
此花	大	0		0		0		0	
	中	1	1	0	0	0	0	0	0
	小	0		0		0		0	
福島	大	0		0		0		0	
	中	2	4	0	0	0	0	0	0
	小	2		0		0		0	
大淀	大	1		0		1		0	
	中	1	4	0	0	0	1	0	0
	小	2		0		0		0	
西淀川	大	0		0		0		0	
	中	2	4	0	0	0	0	0	0
	小	2		0		0		0	
淀川	大	1		1		1		0	
	中	3	5	1	2	0	1	0	0
	小	1		0		0		0	
東淀川	大	0		0		0		0	
	中	2	5	0	0	0	0	0	0
	小	3		0		0		0	





会計学シソーラスの設計と編成

生島芳郎

I シソーラスの機能と構造

情報検索システムでは、通常、索引語の組合せによって検索を行なうので、これら個々の索引語の適・不適が、検索効率に大きな影響を与える。索引語として用いられるキーワードは、文献情報のインデクシング（索引付け）と検索時の質問語として使用されるが、いずれの場合にもそれぞれのキーワードを、意味論的に整理し相互関係を統制しておくことが、その効率をよくすることになる。用語にこのような操作を施し整備したキーワードリストがシソーラスである。

シソーラスの構造は、形式的には各語を単位とするエントリー内の表示形態とその配列、内容的にはエントリー内の各語の概念の相互関係のネットワークからなっている。そしてシソーラス内で、ある概念または複合概念を意味する語として認められた用語、または記号がディスクリプタである。それゆえ、シソーラスはディスクリプタと非ディスクリプタのエントリーで構成され、各エントリーの見出し語の概念に密接な関係をもつ用語が、その相互関係を示す記号の下に記入される形態になっている。

これら各語の相互関係は、同等関係、階層関係、関連関係の三つに大別され、通常次のように記される。

(1)同等関係

A、B両語が同義語で、Aをディスクリプタ、Bを非ディスクリプタとすると、Aのエントリーでは、UF(Used for) 又は SF (Seen from) の記号のあとにBを記してその優先関係を示し、Bのエントリーでは USE 又は SEE のあとにAを記す。

(2)階層関係

ディスクリプタAに概念上の上位語Cと下位語Dがある場合、Aのエントリーでは、BT(Broader term) 又は GT(Generic term) のあとにCを、NT(Narrower term) 又は ST(Specific term)のあとにDを記す。

(3)関連関係

明確な上下関係の概念には入らないが、非常に密接な関係にある語や、同義ではないが同義に近似の概念をもつ語などを、RT (Related term) の記号のあとに記す。

(4)注記

ディスクリプタ語の概念や意味をより明確にしたり、使用について何らかの限定をする必要のある場合、SN(Scope note) の表示の下にその説明句を記す。

このような形態のエントリーをまとめたシソーラスには、各エントリーの見出し語の語順(ABC順、50音順など)に配列した本体と、これらの用語が属する学術領域の体系に、各語を分類配列したカテゴリー索引を付けるのが通常である。カテゴリー索引は、各見出し語をその語に付記した体系表示記号によってグルーピングした後、その体系順に配列したもので、用語の概念上の位置づけと、関係語群を一覧出来る利点があり、用語の選定と管理に役立つものである。

ここでこれらの一例として、「JICST シソーラス」と「OECD マクロシソーラス」を見てみよう。

「JICST シソーラス」

日本科学技術情報センター(JICST)が1975年8月刊行した「JICST 科学技術用語シソーラス」⁽¹⁾の略称である。電気工学分野の用語から作業を始め、1973年の暫定版を基礎に、米国の TEST (Thesaurus of engineering and scientific terms, 1967.) を参考にして作成されたわが国での本格的な総合シソ

(1) 齊藤和男「JICST 科学技術用語シソーラスの編成経緯」情報管理18(5)390-399(1975)

1. J I C S Tシソーラス

- * 情報管理 (ジョウホウカンリ)
 - B A O I
 - USE ドキュメンテーション
- * 情報管理者 (ジョウホウカンリシヤ)
 - B A O I
 - USE ドキュメンタリスト
- 情報帰還 (ジョウホウキカン)
 - E F I I
 - NT 再送訂正
判定帰還
 - BT 通信操作
- 情報機関 (ジョウホウキカン)
 - B A O I
 - NT 情報センター
情報分析センター
データセンター
- 情報源 (ジョウホウゲン)
 - B A O I
 - 情報検索 (ジョウホウケンサク)
 - B A O I
 - U F I R
 - 機械検索
 - NT
 - 事項検索
 - ・データ検索
 - ・化学構造検索
 - そ及検索
 - 文献検索

ーラスである。ディスクリプタ29,173語と非ディスクリプタ 4,825語で編成された本体の一例は、第1表のようになっている。

見出し語の下にその語の属するカテゴリーコードを付し、その下部に USE, UF, BT, NT,などに続きそれに該当する語の記載がある。NT では概念が下位に降りるに従って語の頭部に・印を加え見易くしてある。また見出し語のうち左側部に*印のついているのは非ディスクリプタである。主題カテゴリーは専門領域毎に約160に分れ、見出し語に付されたカテゴリーコードは4桁(英・数字各2桁)で主題カテゴリー別索引はこのコード順に見出し語を配列してある。

「OECDマクロシソーラス」

経済協力開発機構(OECD)が1969年に発表したディスクリプター一覧表⁽²⁾の語彙を増加し、編成したもので、1972年に英-仏、仏-英両語版がまず出された。この編集には Unesco, UNCTAD, UNIDOなど27国際機関が協力し、OECD開発センターがまとめた経済・社会分野での初めての大型シソーラスである。

シソーラス本体の見出し語は 3,004語で、各国語版とも英語との二国語併記を原則としている。各見出し語には所属主題の位置を示すカテゴリーコードを付記してある。カテゴリーは経済事情、経済政策、農業、工業、商業、経営、

(2) Aligned List of Descriptors. 次の三機関のシソーラスを併合編集したもの、ILO-List of Descriptors, FAO-List of Descriptors, DSE-The thesaurus of the Deutsche Stiftung für Entwicklungsländer. 「マクロシソーラス」の正式名称は次の通り Macrothesaurus ; A basic list of economic and social development terms.

労働、環境・資源など19のブロックを最初の2桁、各ブロック内の項目（ファセット）を次の2桁で展開し、さらに必要に応じてサブファセットを次の2桁でコードする。各語の相互関係の表示には、USE, UF, BT, SN など標準的記号を各国語版に共通して使用している。このアルファベット順シソーラスを第1部とし、第2部は主題体系順、すなわちファセットコード順シソーラスという編成になっている。（第2表参照）

2. OECDマクロシソーラス

[ABC順シソーラス]

CORRESPONDENCE COURSE/COURS PAR
CORRESPONDANCE—06.07.00
CORROSION/CORROSION—08.12.03
RT BIODETERIORATION
COSMETICS/COSMETIQUE—08.12.08
COST/COUT—12.09.00
NT CAPITAL COST
DISTRIBUTION COST
EQUIPMENT COST
LABOUR COST
OVERHEAD COST
PRODUCTION COST
COST ACCOUNTING/CALCUL DES COUTS —
12.09.00
COST BENEFIT ANALYSIS/ANALYSE COUT
AVANTAGE—12.09.00
COST OF LIVING/COUT DE LA VIE—03.02.05
SN THE ACTUAL COST OF GOODS AND
SERVICES ACCEPTED AS NECESSARY
TO LIFE IN GENERAL. DO NOT
CONFUSE WITH STANDARD OF LIVING.
COSTA RICA/COSTA RICA—01.04.03
BT CENTRAL AMERICA
COTTAGE INDUSTRY/INDUSTRIE FAMILIALE—
08.02.02
COTTON/COTON—07.07.07
BT INDUSTRIAL CROP
RT TEXTILE
COUNCIL OF EUROPE/CONSEIL DE L'EUROPE—
01.03.03

[主題体系順シソーラス]

12.09 COST ACCOUNTING. PROFIT.
 12.09.00
 AUDITING/VERIFICATION COMPTABLE
 BANKRUPTCY/FAILLITE
 BUDGET/BUDGET
 SN OF AN ENTERPRISE. IN THE
 FRAMEWORK OF PUBLIC FINANCE.
 USE NATIONAL BUDGET.
 BUDGETING/ETABLISSEMENT DU BUDGET
 CASH INCOME/REVENU DISPONIBLE
 BT PROFIT
 COST/COUT
 NT CAPITAL COST
 DISTRIBUTION COST
 EQUIPMENT COST
 LABOUR COST
 OVERHEAD COST
 PRODUCTION COST
 COST ACCOUNTING/CALCUL DES COUTS
 COST BENEFIT ANALYSIS/ANALYSE COUT
 AVANTAGE
 EQUIPEMENT COST/COUT D EQUIPEMENT
 BT COST
 LABOUR COST/COUT DE LA MAIN D OEUVRE
 SN ALL COSTS INVOLVED IN PROVIDING
 LABOUR. I.E. WAGES, ALLOWANCES, SO
 CIAL SERVICES, ETC.
 BT COST

情報検索の分野に、シソーラスという用語が登場したのは、1957年米国が始めであるといわれている。以来、機械検索の発達とともに、種々の専門分野で、シソーラスが編集、使用されてきた。しかし第3表に見るように、その多くは科学・技術分野の用語を取扱ったもので、人文・社会科学分野では極めて少数である。このような既存のシソーラスは、統一的な規則によって作成されてはいないので、その構成、語の表示、配列法などに差異があり、総合的な利用に不便であるため、その規格や作成法の標準化をはかる作業が国際的に行われてきた。

3. 主要なシソーラス (刊年順)

〔自然科学分野〕

- Thesaurus of ASTIA descriptors. U. S. Dept. of Defence. 1960.
 Chemical engineering thesaurus. N. Y., American Institute of Chemical Engineers. 1961.
 Thesaurus of engineering terms. N. Y., Engineers Joint Council. 1964.
 Thesaurus of information science terminology. C. K. Schultz ed. Washington D. C., Communication Service Corp. 1964., Revised edition 1968.
 NASA thesaurus. U. S. National Aeronautics and Space Administration. 1967., Revised edition 1976.
 TEST Thesaurus of engineering and Science terms. U.S. Office of Naval Research, EJC. 1967.
 INIS: Thesaurus. Vienna, International Atomic Energy Agency. 1970.
 JICST 科学技術用語シソーラス, 日本科学技術情報センター 1975.

〔社会科学分野〕

- Information retrieval thesaurus of education terms. Cleveland, Case Western Reserve Univ. 1968.
 Thesaurus of ERIC descriptors. 2nd ed. U. S. Office of Education, Educational Resources Information Center. 1969.
 Thesaurus of ILO descriptors. Geneva, International Labor Office. 1971.
 Thesaurus of industrial development terms. N. Y., UNIDO. 1971.
 Macrothesaurus: A basic list of economic and social development terms. Paris, OECD, Development Center. 1972.
 Thesaurus du management et de l'économie; langage d'indexation pour les services de documentation et bibliothèques. Paris, Bureau Marcel van Dijk. 1975.

そしてユネスコと国際標準化機構の専門委員会 (ISO TC/46) が、1974年にまとめた「単言語シソーラスの設定と発展のための指針」(ISO 2788)⁽³⁾が、その後シソーラスの標準的規則として各国の承認を得ている。この指針は基本的に自然科学、工学、技術分野のシソーラスを対象としているので、人文・社会科学分野への適用には一考を要する点や、日本語として言語上特別な配慮を要する点などが在るが、シソーラス作成の一般基本的な準拠規定として、今後活用されるものである。

(3) Documentation-Guidelines for the establishment and development of monolingual thesauri. (ISO 2788-1974) 13p. 荒木啓介* 単言語シソーラスの設定と発展のための指針” 情報管理 19 (5) 343-349, 19 (6) 417-425 (1976)

Ⅱ 会計学シソーラスの編成*

1. 語彙源と用語構成

前稿⁽⁴⁾で報告した会計学索引語表は、1970年まで15年間の会計学論文・記事4573アイテムに使用したキーワード11,442語のうち、使用頻度3以上の1773語を五十音順に配列したもので、これを基本索引語としてシソーラス作成に利用する。さらに1971年以降の文献をも含めて広く語彙を加え、その後、各用語間の相互関係を検討してディスクリプタを選定し、シソーラスを編成することにした。

これらの作業のうち、用語の概念や相互関係の検討、主題カテゴリーへの決定などは、主題領域と用語法を熟知した専門研究者が行なうのが望ましいが、今回は前段階として既存の専門辞典類を利用することとし、神戸大学会計学研究室編「会計学辞典」(同文館)を活用した。本辞典は斯界の代表的な専門辞典で、昨年改訂第三版が出されたものである。これを用語の内容検討のほか、辞典巻末の総索引を会計用語の語彙集とみて、われわれの索引語表中の用語や追加語との照合、および用語の重要度の検討に利用し、またその体系目次の項目を主題カテゴリーの決定に準用した。

収集用語の語彙源と第一次統制を加えた語の概数は次の通りである。

(1) 索引語表の1773語のうち

(a) 辞典の総索引に出ている語 1,290

(b) 辞典の総索引に出ていない語 483

(2) 1971~75年間の文献タイトルの重要語 580語のうち、索引語表と辞典総索引ともに出ていない語 114

(3) 辞典の体系目次にある項目語のうち、索引語表に出ていない語 2,243

用語の収集源として、さきの「ISOのシソーラス指針」は(a)技術用語辞典、(b)最新の文献、(c)用語に関する文献やリスト、(d)抄録誌、(e)文献の実験的索引作業等々を挙げている。われわれの作業のうち、(1)はISOの(e)に、(2)は(b)、

(4) 「会計学用語と索引語」経営機械化シリーズ№15 (1974) 所収

* この編成作業は文献センターの関口秀子助手との共同作業で行っているものである。

(d)に、(3)は(a)に対応しているといえよう。

次に上記(1)～(3)の語彙内容と問題点をのべる。

(1) 上記(1)の索引語は使用頻度10以上の443語を含む1,773語で、この分野の重要キーワードと考えられる。JICST シソーラスの作成時の用語は、異なり語17万語、うち使用頻度3以上は約3万語、頻度10以上は約7,000語であり、分野と語数は異なるが、頻度3以上の語数対異なり語数の比率は、われわれの場合15%、JICSTは17%、また頻度10以上の比率は同じく25%と24%で、ほぼ同様な傾向を示してをり、収集語彙と使用頻度の相関関係が考えられる。

JICST の場合は、頻度10以上の語に USE-UF の優先関係の処理をして、ディスクリプタ候補としている。われわれは頻度3以上の語を辞典の総索引と照合し、上記の(a)、(b)に分けた。辞典の総索引に入っている語を、ある概念を現わす最も普遍的な語として認められた用語であるとすれば、(a)は重要度の高いディスクリプタ候補である。また(b)には次のような語が含まれている。

- (イ) 1E, OR, 証取法などの略語、俗語、ダイレクト・コストイング（直接原価計算）、インターナショナル・アカウンティング（国際会計）などの様に邦訳語と併用され、訳語のみが辞典索引にあるもの
- (ロ) 利益センター（プロフィット・センター）のように(イ)の場合と反対に外国語のみが辞典索引にあるものと総合減価償却（総合償却）のように同義関係にあるもの
- (ハ) オートメーション、目標管理、市場調査、創業利得など周辺カテゴリーに属するもの

辞典の索引には、辞典本文の見出し語はすべて入っているが、見出し語の説明句にある用語全部は索引で拾われてはいないので、索引にない語は不用語であるということにはならない。とくに(b)の中で頻度10以上の用語、例えば「貸倒準備金」「正規監査」「業績評価基準」「客体計算」などの語は、実際の使用語としての重要度において(a)の各語に劣るものではなく、学術用

語としての位置は与えられてなくてもディスクリプタ候補と考えられる。ただ用語の相互関係検討段階で、非ディスクリプタになる場合はあるが、ディスクリプタへの導入語としてシソーラスにエントリーすることが考えられよう。

なお、(b)の(i)の周辺カテゴリーに属する語は、主に経営学領域の用語である。これらは会計学本来の領域に属する用語処理が終了後、総合的に検討する予定である。

(2) 語彙源(2)の内容は次の通りである。(1)の索引語表は1956—1970年間の文献をその収集源としてをり、その後の文献にも同様な方法で、キーワードの収集を継続するのが望ましい。しかし、量的・労力的に早急には出来ないので、これを幾分でも補うため、文献内容の反映度が低いという欠点はあるが論文標題中のキーワードを収集した。これには1971—1975年間の「経済学文献季報」(60/61~79号)の分類項目「会計」に採録された3051文献の標題を用い、使用頻度2以上の用語580語を選出した。次にこれらと(1)の索引語表及び辞典総索引の両語彙とを照合して、次の結果を得た。

(i) 索引語表に出ていない語	207
(ii) (i)のうち辞典索引にも出ていない語	114

追加語彙として検討を要するのは、(ii)の114語である。この中には「会計評価基準」「会計プロセス」「引当金会計」のように、用語統制、相互関係の検討の段階で、それぞれ評価基準、会計処理、引当金などの関連処理を予想出来る語がある。また「為替換算会計」「行動会計」「税効果会計」「代用償却」「半期報告書制度」「連結会計基準」や、周辺カテゴリーに属する「環境」「公害」のような新出語が含まれている。

研究の発展につれて発生する新生語は、学術用語としての位置を認められるのに日時を要するものもあるが、個有の新概念を表す用語で、使用頻度の高い新出語は、既存用語との関連を検討して、シソーラス上の位置づけを考えねばならない。けだしシソーラスの目的は、用語の解説ではなく、語間関係の整理統制によって常に検索に役立てることにあり、新出語の検討と処置

はシソーラスのメンテナンスに必要な作業の一つである。

社会科学分野の論文タイトルには、一般に科学・技術分野のそれに比べて、上位概念の用語が多く、また一方では筆者の個人的な造語の使用が見受けられる。上記①の中にも、構成単語への分解で検索可能な複合語や個人的な造語が含まれてをり、これらの検討が必要である。

(3) 語彙源の(3)にあげた辞典の体系目次の項目語の中には、すでに前述の(1)、(2)に含まれている用語があるので、これらを除いた用語が対象となる。これら項目語は、すべて辞典本文の見出語としてそれぞれ解説文が付されてをり、会計用語中の重要語という観点からは、ディスクリプタ候補として差支えない。

ただ、これらの中には、「入金伝票」「貸金台帳」「支払手形記入帳」「通信費」「総仕入高」「前期繰越利益」などや「小口現金%」「敷金%」などの勘定科目名称のように、学習用語としては必要であるが、実際の研究分野の主題分析・検索には重要度が低く、ほとんど使用されない語がある。この種の用語のシソーラスへの収容範囲について検討の要があろう。

また本目次の会計法規・準則、会計学者・会計団体の項目には、企業会計原則、同注解、財務諸表準則、計算書類規則、税法調整意見書、APB 意見書、AAA会計原則、ASOBAT や、シュミット、シュマーレンバッハ、日本会計学会、企業会計審議会、アメリカ会計学会、AIA、NAA、のような規則名、機関名、個人名、略名、その他の固有名称が入っている。シソーラスではこれらをアイデンティファイヤ（特定語）と称し、通常はシソーラス本体に組入れず、別リストで処理しているものもある。

前述の「ISO 指針」では、これらアイデンティファイヤも一般のディスクリプタと同様に、シソーラス内に配列するよう規定している。例えばOECD マクロシソーラスでは、国際機関名、国名はディスクリプタとしてシソーラスに配列し、カテゴリー分類では国際協力・国際関係というブロック内の国際機関、国家の2ファセットに収容している。国名、人名、機関名、団体名などを含め上記の諸固有名詞は、主題分析や検索質問の場合、情報を特定化

する有力な指標、キータームとして有用であるが、各語間に相互関係がなく、ディスクリプタとして組み入れなくても、それぞれ個別に機能し得る語であり、識別の容易な語であるので、今回はアイデンティファイヤ・リストにまとめて別記することとしている。

2. 用語の統制と相互関係

今回の語彙源はそれぞれ用語構成上の特長を持ち、これに伴うディスクリプタ選定上の問題点を前節で述べた。ここでは用語の統制と相互関係について、前述以外で問題になる諸点をあげる。

(1) 複合語

ディスクリプタは、ある概念または複合概念を意味する用語で、他語と意味的差異の明瞭な普遍的なものが望ましい。複合概念を表わす複合語は、これを構成する語よりも特殊性が高いので、検索に有利な利点があり、ディスクリプタとして必要である。

しかし、今回は見当たらないが、「三語以上で構成する複合語は、構成単語に分解した事後組み合わせ検索によることにして、ディスクリプタにはしない」という考え方もあるように、ディスクリプタの増加によるシソーラスの巨大化を防ぐため、検索頻度の少いことが十分に予測される複合語は、その特殊性の度合いとの関連でディスクリプタへの採否を検討することになる。

(2) 同義語と優先関係

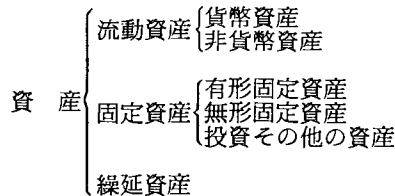
同義語のある場合は、用語として公認され、かつ最も一般に使用される語を優先語としてエントリーする。前節であげた「アカウントビリティー会計責任」「ダイレクト・コストिंग—直接原価計算」のような訳語や、「総合償却—総合減価償却」「利益性引当金—利益留保性引当金」のような簡略語の場合は、完全な同義語であるが、同義に近い準同義の語も含めて同義語処理をしている場合も多い。前にも述べた通り、シソーラスは厳密な語義の表示ではなく、索引付けと検索の用具としてその利用に目的があり、この目的に適合するレベルの準同義語は USE-UF による同義処理を許容する。しかしこの処理の難しい準同義語は関連語 (RT) として表示することにした。

今回、同義語（A、Bとする）が辞典の見出し語に両語ともある場合は、解説付きの語Aをディスクリプタに優先し、解説のない語すなわち「Aと同じ」とある語Bは、索引語表での使用頻度が多くてもUFで表示するのを原則とした。そのため、使用頻度は少ないが辞典に解説のある語はディスクリプタとなり、頻度10以上でも解説のない語は、同義語としてUF表示される場合が出た。下記がその例である。

ディスクリプタ（使用頻度）	非ディスクリプタ（使用頻度）
資本利得 (3)	キャピタル・ゲイン (10)
ピリオド・プランニング (2)	期間計画 (16)
バリュー・アナリシス (3)	価値分析 (15)
プロダクト・コスト (10)	製品原価 (17)
売価棚卸法 (1)	小売棚卸法 (8)
証拠 (3)	監査証拠 (6)
監査証明 (10)	監査報告 (13)
ピリオド・コスト (8)	期間原価 (28)

使用頻度に選定の重点をおけば、ディスクリプタと非ディスクリプタは逆になり、現在のディスクリプタはUF表示でよい。このように同義語間のディスクリプタ選定には、今回の辞典準拠に対して使用頻度と日本語の取り扱いなどの諸点で、検討と調整を要すると思われる。

(3) 階層関係



このような上下の階層関係を各語に付記(BT,NT)して、その語の意味や概念上の位置を明確にし、検索の際、上位語を使用して適合文献の再現もれを少なくする。そのため検索時、自動的に上位語転移(up-posting)をはかる検索戦略が求められている。

しかし、会計学分野のすべての用語に、直結的な上下関係を明示することは困難で、多くの場合、上位にはきわめて広域的な概念を示す語が用いられる。今回は各語にカテゴリー別体系コードを付与してあるので、一応の上位概念はこのコードの利用で得ることが出来る。階層関係の決定には専門知識が必要で、出来るだけ階層付けによる検索効率の向上を図るが、階層関係の不満足な用語、概念については、関連語との併用、活用によることになる。

(4) 関連関係

「現在原価と再調達原価」「会計監査人と公認会計士、監査法人」のように殆んど同義に近い準同義関係、「留保利益と任意積立金、利益剰余金」、「連産品と結合原価」のように意味・概念が非常に関連している関係にあるものなど、同等・階層関係以外で密接な関係のあるものを関連語 (RT) とした。

これはある主題を検索する場合、主題概念の中心となるディスクリプタの周辺に随添して多く用いられる語であるので、検索精度からみて、準同義ないし近似同義の関連語は再現性を上げることになるが、反面、関連語の増加は適合率の低下を招く要因でもある。個々のディスクリプタにおける関連範囲やレベルについては、作業上の大きな問題で十分な検討が必要であるが、その基本は、さきの「ISO 指針」にある「関連関係は索引作業および検索において、十分有効なことが明らかな二語間の関係においてのみ設定すべきである」というにつきている。

3. 表示と配列の形態

現在、上記語彙源の(1)(2)の用語の相互関係を点検作業中で、このあと(3)の追加語彙の点検を行ない、全用語の五十音順リストと体系順(カテゴリー)リストを作成する。

これら両リストには、ディスクリプタと非ディスクリプタが含まれ、各語のエントリーにはUSE,UF,BT等の記号と、これらに関係ある用語、および所属カテゴリーコードが付記される。このような編成は、今までの多くのシソーラスに見られるものであるが、今回は前記 OECD のマクロシソーラスと米国の

4. 会計学シソーラスの例

〔50音順シソーラス〕

アイドル・コスト	0 4 . 0 6 .
UF 不働費	
無効費用	
BT キャパシティ・コスト	
RT 原価性	
アウトプット法	0 4 . 0 6 .
BT 原価差異分析	
RT インプット法	
アカウントビリティ	0 1 . 0 1 . 0 1
USE 会計責任	
後入先出法	0 2 . 0 3 . 0 3
UF LIFO	
RT 棚卸資産原価配分	
安定価値会計	0 2 . 0 1 . 0 5
USE 貨幣価値変動会計	
<hr/>	
営業権	0 2 . 0 4 . 0 7
UF 買入暖簾	
BT 無形固定資産	
RT 暖簾	
営業費	0 4 . 0 5
UF 販売費・一般管理費	
<hr/>	
開業費	0 2 . 0 4 . 0 9
UF 開業準備費	
BT 創業費	

〔体系順シソーラス〕

0 1 . 0 2	会計公準
0 1 . 0 2 . 0 1	会計慣行
	USE 会計慣習
	会計慣習
	UF 会計慣行
	RT コンベンション
	RT 会計公準
	会計公準
	UF 公準
	NT 会計基準
	会計原則
<hr/>	
0 2 . 0 4	固定資産・繰延資産
0 2 . 0 4 . 0 3	
	加速償却
	RT 特別償却
	集合償却
	RT 総合償却
	総合償却
	RT 集合償却
	合成償却
<hr/>	
0 6 . 0 4	監査報告
0 6 . 0 4 . 0 1	
	監査証明
	UF 監査報告
	RT 監査報告書
	監査証明書
	RT 監査証明
	監査報告
	USE 監査証明
	助言
	RT 勧告
0 6 . 0 4 . 0 2	
	意見区分
	RT 除外事項
	意見差控
	RT 未確定事項
	意見差控監査
	報告書

教育用語シソーラス (Information retrieval thesaurus of education terms, 1968. 第3表参照) を参考にした。

体系順リストは主題カテゴリー別索引で、カテゴリーの決定には前に述べたように、辞典の体系目次を利用し、その11大項目と項目内の下位分類に従って、各用語の概念表示コード (最大6桁の数字コード) を与え、このコード順に配列する。この体系目次の利用によって、われわれは負担の軽減を計ったのであるが、次の問題点が見られる。11大項目のうち、社会会計、機械化会計、会計数理の各項目には下位への展開がなく、同じく原価計算、管理会計では次位分類だけで、その下の細分類はされていないことである。とくにこの2カテゴリーは収容語が多いので、改めて各語のグルーピングをして適切なファセットに配列し、コードを付与する必要が生じている。

以上の諸作業が終ったシソーラス完成時のサンプルを第4表に示しておく。

Ⅲ おわりに

以上がシソーラス編成過程とその問題点の概要である。今回は上記の印刷様式のシソーラスの作成をまず目標にしているが、将来、会計学を含め経営学全般の文献検索サービスを提供する場合には、より広領域の語彙を加えた総合シソーラスの編成と、同ファイルの CRT ディスプレイ表示の利用、さらにオンライン検索においては、自然語によるアクセスによって、ディスクリプタを含むエントリー・タームへ自動変換が可能なシステムが、検索効率からみて望ましい。それには、エントリー・タームに関係付けた大量の自然語ファイルの作成が必要で、これら語彙収集、用語の統制とメンテナンスが、常にシソーラスの機能に影響を持っていることに留意しておかねばならない。

バザラのアルゴリズムに対する注釈

伊藤 駒 之

I

制約条件のある最適化問題を解くに際して、非線型計画問題では、ペナルティ関数を付与したり、変数変換を行う方法が現在では広く使われている。これらの方法は、SUMT⁽¹⁾ (Sequential Unconstrained Minimization Techniques) と呼ばれていることが示すように、一連の制約条件なしの問題を系統的に解くことによって非線型の制約条件付きの問題を解こうとするものである。

Fiacco と McCormick⁽²⁾ によれば、変数変換を行う方法は interior point algorithms と呼ばれ、ペナルティ関数を付与する方法は exterior point algorithms と呼ばれている。例えば、制約条件 $x+3 > 0$ のもとで

$$\min (x-2)^5$$

という最適化問題はつぎのような形に変数変換によって代替される。

$$\text{Min } \{(x-2)^5 - \gamma \log(x+3)\},$$

ここで、 γ はラグランジェ乗数に相当する。変換された関数から判明するように、最適化行動は制約条件内の点、すなわち、実行可能 (feasible) な領域内の点をたどる。

一方、この最適化問題は exterior method では

$$\text{Min } \left\{ (x-2)^5 + \frac{\{\min [0, x+3]\}^2}{2\gamma} \right\},$$

(ここで、 γ はラグランジェ乗数に相当する。) なる形になされうる。この方法においては、 x が制約条件から、すなわち、実行可能領域から離れるに従っ

(1) Fiacco and McCormick, [2] 参照。

(2) *op. cit.*

て、ペナルティ $\{\min(0, x+3)\}^2$ がますます大きくなるように形式化がなされている。それゆえに、この方法では実行可能領域外から最適化がなされる。

このように、interior point algorithms と exterior point algorithms は最適化行動が実行可能領域内で行なわれるか、実行可能領域外からその領域内へ進むかによって区別される。

目的関数にペナルティを付与する方法に対して、M.S.Bazaraa は Friedman-Savage の方法を適用している。⁽³⁾ そして、この定式化を検討するために、確率化された Friedman-Savage の方法を我々は適用してみた。その結果は M. S. Bazaraa が指摘するほどの良いものをもたらさなかった。その原因を追求していく過程で、M. S. Bazaraa が示した algorithm に疑問点をみいだすことになった。本稿では、この疑問点と Bazaraa の興味ある議論に注釈を加えることにしよう。

II

M. S. Bazaraa に従い、つぎのような制約条件付きの非線型計画問題を考えよう。

$$g_i(x) \leq 0, \quad i = 1, 2, \dots, m,$$

$$h_i(x) = 0, \quad i = 1, 2, \dots, k$$

なる制約条件のもとで

$$f(x) \tag{1}$$

を最小にせよ。

ペナルティ関数を使ってこの問題を解くには、つぎの式を満す x をみつけないければならない。

$$\sup_{\lambda} \inf_{x \in R_n} (f(x) + \lambda I(x)) = \lim_{\lambda \rightarrow \infty} \inf_{x \in R_n} (f(x) + \lambda I(x)) \tag{2}$$

ここで、

$$g_i(x) \leq 0 \quad (i = 1, 2, \dots, m)$$

(3) Bazaraa, [1] 参照。

でかつ、

$$h_i(\mathbf{x}) = 0 \quad (i = 1, 2, \dots, k)$$

なら、

$$I(\mathbf{x}) = 0$$

さもなくば、

$$I(\mathbf{x}) > 0$$

とする。関数 $I(\mathbf{x})$ の典型的な一例は

$$I(\mathbf{x}) = \sum_{i=1}^m (\max(0, g_i(\mathbf{x})))^2 + \sum_{i=1}^k h_i^2(\mathbf{x}) \quad (3)$$

である。

そして、この問題は十分に大きい正の λ に対して、

$$f(\mathbf{x}) + \lambda I(\mathbf{x})$$

を最小化することによって解決される。しかしながら、これは、 λ が大きいため、かなり困難な問題である。Bazaraa はつぎのような叙述によってこの困難さを説明している。

いま、ノルムが1である方向ベクトル \mathbf{d} が与えられているとしよう。そのとき、実行可能な点から構成されている領域 $\{\mathbf{x} \mid I(\mathbf{x}) = 0\}$ は非常に制約された非線形であろうから、任意のスカラー μ に対して $I(\mathbf{x} + \mu\mathbf{d}) > 0$ となる可能性が大きいだろう。そして、 $I(\mathbf{x} + \mu\mathbf{d})$ は大きな正数によって加重されているので、

$$f(\mathbf{x} + \mu\mathbf{d}) + \lambda I(\mathbf{x} + \mu\mathbf{d}) > f(\mathbf{x}) + \lambda I(\mathbf{x}) = f(\mathbf{x}) \quad (4)$$

となる。すなわち、 f の値についてのなんらかの改善がなされたとしても、 $I(\mathbf{x} + \mu\mathbf{d}) > 0$ で、かつ、 λ が大きな正数であるために全体としては増加が生まれる。もし μ の値が非常に小さくて、 $I(\mathbf{x} + \mu\mathbf{d}) \doteq I(\mathbf{x})$ なら、上記の問題の最小化が進むことになるだろう。しかし、 μ の値が小さいことは最適化過程が非常に遅い速度で進むことを示す。このように、ペナルティが付与された関数を最適化することは困難な問題である。ペナルティが大きい正数 λ で加重されているがために、等高線は ill-condition に関して極めてきびしいものになる、す

なわち ridge $\{x | I(x) = 0\}$ によって非常に複雑な形になり、最適化行動を容易にさせない曲りくねった細長い形状を有することになる。

この点に関して、Bazaraa は Rosenbrock の関数⁽⁴⁾ を使って説明をしている。この関数は $x = (1, 1)$ で最小値をもち、放物線 $x_2 = x_1^2$ にそっての険しい ridge をもっている。また、これは制約条件なしの最適化問題におけるテスト用の関数としてしばしば利用されている。

いま、つぎのような制約条件付きの最適化問題を考えよう：

$$x_2 - x_1^2 = 0$$

なる制約のもとで

$$(1 - x_1)^2$$

を最小化せよ。

前記の定式化に従って、 $I(x) = h^2(x) = (x_2 - x_1^2)^2 = 0$ とし、パラミター $\lambda = 100$ とおくことによって、ペナルティ付きの関数は $(1 - x_1)^2 + 100(x_2 - x_1^2)^2$ となり Rosenbrock の関数そのものとなる。そして、探索の過程で $x_2 = x_1^2$ なる点に到達したとしよう。そのとき、 $(1 - x_1)^2$ の項における改善はペナルティである $100(x_2 - x_1^2)$ によって消され、場合によっては（それはしばしば考えられる。）全体としては改善もたらされる。

このように、パラミター λ の値が大きいたくには、アルゴリズムのステップ・サイズと考えられる、上記の μ が小さな値をもつような状態でなければ、探索成果は改善されえない。これはアルゴリズム (algorithm) 構成にさいして注意されなければならない重要事項と考えられる。

このような問題点から、パラミター λ の小さな値で探索を始め、徐々に λ の値を大きくしながら探索を続ける計画が提案されている。

III

本節では、前記に定式化されたペナルティ付きの関数を最適化する、

(4) Wilde, [3] 参照。

Bazaraa のアルゴリズム (algorithm) が述べられる。

つぎのように記号を定義しよう。

- d_i : i 番目の要素だけが1で、その他は0であるベクトル,
- y_{i+1} : ベクトル y_i からベクトル d_i の方向に沿って最適化されたベクトル,
- Δ : 初期ステップ・サイズ,
- Δ_{min} : ステップ・サイズの下限,
- β : ステップ・サイズの削減率,
- λ : ペナルティのパラメーター,
- λ_{max} : λ の上限,
- γ : ペナルティの増加率,
- ϵ : パラメーターの値の変更のための成果規準,
- t : 繰り返し回数,
- x_{t+1} : t 回の繰り返し後の到達点,
- I : パラメーター λ とステップ・サイズ Δ の変更のためのスイッチ要因,
- $|x|$: ベクトル x のユークリッドノルム,
- $p(x, \lambda) = f(x) + \lambda I(x)$.

アルゴリズム (algorithm)。

上記のパラメーターの値を定め、 x_1 を適当に選び、 $y_1 = x_1$ $i = 1$ と設定する。

段階Ⅰ : 方向 d_i にそって、ステップ・サイズ Δ で、 $P(y_i, \lambda)$ の探索を行う。そして、得られた最適値を y_{i+1} とする。もし $i \leq n$ なら、 i を $(i+1)$ と置き換え、 $i = (n+1)$ になるまで、上記の手続を繰り返す。そして次の段階に進む。

段階Ⅱ : $x_{t+1} = y_{n+1}$ と置く。 $|x_{t+1} - x_t| \geq \epsilon$ なら、 $i = 1$, $y_1 = x_{t+1}$ と置き、段階Ⅰに戻る。いま、 $|x_{t+1} - x_t| < \epsilon$ としよう。そのとき、もし $\lambda > \lambda_{max}$ で、かつ $\Delta < \Delta_{min}$ なら、 x_{t+1} を最適とみなし、アルゴリズム (algorithm) は

停止する。そのようになっていない場合には 1 と $\lambda \Delta$ を比較し、 $\lambda \Delta < 1$ なら、 λ は $\gamma \lambda$ とおきかえられる。一方、 $\lambda \Delta \geq 1$ なら、 λ は 2λ とおきかえられ、 Δ は $\text{Max}(\beta \Delta, \Delta_{\min})$ とおきかえられる。そして、 $i=1, y_1=x_{i+1}$ と置き、段階 I に戻る。

このアルゴリズム (algorithm) を適用するに際して、Bazaraa はつぎの表に示されるような値をパラミターとして選ぶことを勧めている。

表 パラミターの値

パラミター	Δ	Δ_{\min}	β	λ	λ_{\max}	γ	ε	l
勧められる 範囲	0.1から 1まで	0.00001 から 0.001 まで	0.1から 0.4まで	0.1から 1.0まで	100から 1,000,000 まで	5から 100まで	0.0001 から 0.01まで	0.1から 2まで

IV

前節でのアルゴリズム (algorithm) において、Bazaraa は、線型サーチとして、つぎのような imperfect line search と呼ぶ方法を提案している。ステップ・サイズを Δ とし、初期加速要因 $S=1$ としよう。初期値を x^* とし、 $Z^*=P(x^*)$ とする。そして、 $P(x^*+S\Delta)$ を計算する。もし $P(x^*+S\Delta) < Z^*$ なら、 Z^* は $P(x^*+S\Delta)$ とおきかえられ、 x^* は $x^*+S\Delta$ とおきかえられ、 S は αS ($\alpha > 1$) とおきかえられる。そして、最初の失敗、すなわち、 $P(x^*+S\Delta) \geq Z^*$ が起るまで、この手続が繰り返えられる。このようにして得られる $P(x)$ の imperfect minimum は x^* となる。もし最初の計算において $P(x^*+S\Delta) \geq Z^*$ なら、 $S = -1$ とおかれ、上記の手続が繰り返えられる。

この imperfect line search を Bazaraa が提案している理由として、 λ の値が小なるときに完全な最適化を行うことは、それに使用される努力量に見合う成果がでることにならないことが挙げられている。

大きなペナルティ、すなわち、 λ の値が大である場合には、前述のように最適化がかなり困難であるので、Friedman-Savage の方法よりは洗練された方

法がより適切ではないかという考えが生じる。この点に関しても、Bazaraa はつぎのような反対の見解を示している。

例えば、conjugate direction にもとづくアルゴリズム (algorithm) を使うなら、conjugate direction が正確に得られるかぎり、良い成果がもたらされるであろう。そして、conjugate direction を作るには、Hessian に関する多くの情報が必要とされる。しかしながら、計算誤差や ill-condition が存在するために、正確な情報を得ることはほぼ不可能であり、結果として生じてくる方向 (direction) は conjugate とはほど遠いものになる。

本節で述べてきた Bazaraa の見解に対して我々は強く賛意を表する⁽⁵⁾。この意味で我々はこのアルゴリズム (algorithm) に深い関心をもった。しかしながら、Bazaraa のアルゴリズム (algorithm) にはつぎに述べるような問題があった。

段階Ⅰでは、問題点は見い出さなかったが、段階Ⅱでは、パラミターの選択に関して十分な注意を払う必要がある。

彼の議論によれば、この工夫、すなわち、スイッチ要因 l は、ペナルティのパラミターが小さな初期値で、ステップ・サイズが大きな値で始められるとき、不完全な最小化を実施するのに有効で、一方最適化の終了直前では、小さなステップ・サイズによりかなり精緻な最小化を実施するのに役立つ。

しかしながら、パラミターの選択が当該の問題にとって適切でないなら、このアルゴリズム (algorithm) は有効に働かないかもしれない。Bazaraa はパラミターの選択に関してアルゴリズム (algorithm) の感受性を無視するものであると述べているが。つぎのような考慮がこの点を我々に説明するであろう。

アルゴリズム (algorithm) のパラミターが前記の表に従って定められたとしよう。そして、 $\lambda \Delta < 1$ なる場合をケースAと定義し、 $\lambda \Delta \geq 1$ なる場合をケースBと定義しよう。 t_0 回、ケースA ($\lambda \Delta < 1$) が起ったのちに $\lambda \Delta \geq 1$ が

(5) 伊藤, [4], [5]参照。

起ったとし、このケースAが最初に起ったときの λ, A を λ_0, A_0 とする。

アルゴリズム (algorithm) によれば、 $\lambda A < 1$ なら、 A はそのまま固定され、 λ は $\gamma \lambda$ とおきかえられる。それゆえに、

$$\gamma^{t_0} \lambda_0 A_0 \geq 1 \quad (5)$$

が成立しなければならない。この不等式を変形するとつぎのような不等式が導かれる。

$$\frac{\log (1/\lambda_0 A_0)}{\log \gamma} + 1 > t_0 \geq \frac{\log (1/\lambda_0 A_0)}{\log \gamma} \quad (6)$$

ここで、

$$t_0 \geq 1, \\ \log \gamma > 0$$

とする。前記の表より $\gamma \geq 5$ であるから、

$\log \gamma > 0$ は成立する。

この不等式を満たす非負整数 t_0 は確に存在する。なんとならば、 $\lambda_0 A_0 < 1$ であるから $\log (1/\lambda_0 A_0) > 0$ である。

一方、このケースAに移る前のケースBにおける最終段階では、

$$\frac{\lambda_0 A_0}{2\beta} \geq 1 \quad (7)$$

が成立していなければならない。ゆえに、

$$\frac{1}{2\beta} \geq \frac{1}{\lambda_0 A_0} > 1 \quad (8)$$

となる。 t_0 に関する以上の議論をまとめると

$$\frac{\log (1/2\beta)}{\log \gamma} + 1 \geq \frac{\log (1/\lambda_0 A_0)}{\log \gamma} + 1 > t_0 \quad (9)$$

が成立する。

同様に、 t_1 回、ケース B ($\lambda A \geq 1$) が起ったのちに $\lambda A < 1$ が起ったとしよう。そして、このケースBが最初に起ったときの λ, A を λ_1, A_1 としよう。

アルゴリズム (algorithm) に従って、もし $\lambda A \geq 1$ なら、 λ は 2λ と変換され、 A は $\max (\beta A, A_{min})$ によっておきかえられる。いま、 A_{min} は特殊な場合

として無視されよう。そのとき、

$$2^{t_1} \beta^{t_1} \lambda_1 \Delta_1 < 1 \quad (10)$$

が成立しなければならぬ。前述と同様に、この不等式を変形すると

$$\frac{\log(1/\lambda_1 \Delta_1)}{\log 2\beta} + 1 \geq t > \frac{\log(1/\lambda_1 \Delta_1)}{\log 2\beta} \quad (11)$$

が導かれる。

この場合も、 $\log(1/\lambda_1 \Delta_1) < 0$ で、 $\log 2\beta < 0$ であるゆえに、上の不等式を満たす整数 t_1 が存在する。この2つの整数 t_0 , t_1 が存在することはアルゴリズム (algorithm) がケースAとケースBの間で変動することを示している。

また、このケースBに移る前のケースAの最終段階では、

$$\frac{\lambda_1 \Delta_1}{\gamma} < 1 \quad (12)$$

が成立していなければならない。ゆえに

$$\frac{1}{\gamma} < \frac{1}{\lambda_1 \Delta_1} < 1 \quad (13)$$

となる。 t_1 に関する以上の議論をまとめると、

$$\frac{\log(1/\gamma)}{\log 2\beta} + 1 > \frac{\log(1/\lambda_1 \Delta_1)}{\log 2\beta} + 1 \geq t_1 \quad (14)$$

が成立する。

つぎに、この t_0 と t_1 の間にどのような関係をもつかみてみよう。

いま、

$$A = \frac{\log(1/\gamma)}{\log 2\beta} \quad (15)$$

とおこう、そのとき

$$\frac{1}{A} = \frac{\log(1/2\beta)}{\log \gamma} \quad (16)$$

となる。

ここで、不等式 (9) に式 (16) を代入し、不等式 (14) に式 (15) を代入すると、

$$A+1 \geq t_1 \quad (17)$$

$$1/A+1 \geq t_0$$

(18)

を得る。

この式 (17), 式 (18) から判明するように, $A=1$ のときをのぞいて, t_0 と t_1 のどちらかは 1 でなければならない。このように, Bazaraa によるスイッチ要因の工夫はその意図に反して有効な働きをなすとは考えられない。

参 考 文 献

- [1] Bazaraa, M. S., "An Efficient Cyclic Coordinate Method for Optimizing Penalty Functions," Naval Research Logistics Quarterly, Vol. 22, No. 2, June, 1975.
- [2] Fiacco, A. V. and McCormick, G. P., Nonlinear Programming : Sequential Unconstrained Minimization Techniques, Wiley, 1968.
- [3] Wilde, D. J., Optimum Seeking Methods, Prentice-Hall, 1964.
- [4] 伊藤駒之, 「パターン・サーチについて」, 経営機械化シリーズ, No.14, 1973.
- [5] 伊藤駒之, 「Hooke-Jeeves の方法の一変種」, 国民経済雑誌, 第133巻, 第5号, 昭和51年5月。

A Manual-free Program

——対話型簡易汎用作表プログラム
Dialogue-CROTAB——

下 條 哲 司

1. は し が き

新しいプログラムが完成すると、それがちょっとしたサブルーチンであっても、それを使用するために必要な予備知識—プログラム名したがってそれを呼び出すための Calling sequence, 引数1つ1つの型, 長さ, 意味など, サブルーチン内部での処理のしかた, 帰ってきたときの変数や配列の内容, 精度などについての詳細, 等々——を詳しく記述しておく必要がある。これなくしてはごく簡単な処理しか行なわないようなサブルーチンでも, 満足には動いてくれない。サブルーチンの目的だけを知っていても, 引数の順序や数はつねに固定的であって, 前回使用したときと全く同じ使い方であったとしても, 引数の1つすら省略は許されない。

まして汎用プログラムともなれば, これを使用するために, 利用者が提供しなければならない情報は非常に多いだけではなく, それを与えるしかた=文法は, その汎用プログラムがより多くの機能をもつ, 便利なものであればあるほど, 複雑で覚えにくいものである。そこで一定の意味をもたされている用語や記号=語彙もまた, それが沢山あればあるほど全部を完全に使いこなすのに, 相当の熟練を必要とする。

ただ1つの機能をもつ, 単能プログラムであれば, 処理を受けようとする材

料としての情報だけを与えるだけで用は足りる。

$$X = \text{SIN} (A)$$

という関数は、ただ1つの情報Aを与えるだけで充分であるが、この場合でも例えば、Aはラジアンで表現された角度であって、4バイトの大きさの実数でなければならないといった限定が行なわれねばならないし、結果もまた4バイトの実数となるといった、いわずもがなの説明も一般的には必要になる。

もしこの関数SINに、引数として与える値が、度であってもラジアンであっても、どちらでもうまく計算できるような機能を与えるとすれば、Aの値が度であるかラジアンであるかという情報をもうひとつ追加して、

$$X = \text{SIN} (A, 1)$$

または

$$X = \text{SIN} (A, 2)$$

と書かせ、第2の引数が1であればラジアン、2であれば度として計算を行なうといった約束を作らねばならない。そしてそのために引数が1つ増加するとともに、約束ごとを利用者に知らせるための記述が、マニュアルにおいて必要となる。Aが8バイトの実数であったり、4バイトまたは8バイトずつの実部虚部をもつ複素数であったりする場合にも、この関数の機能を拡張すれば、さらに引数はふえ、その引数の意味を説明すべきマニュアルも、それだけ内容の多いものとなる。

このような関連はごく当然のことであり、便利なプログラム、すなわちより多くの機能を備えたプログラムであればあるほど、それを利用しようとする者に多くの情報を要求し、その情報の1つ1つについての詳しい説明が必要となってくる。情報を与える順序=文法と、各情報の内容と意味との対応=語彙とが約束として定められ、その約束が詳細に記述されねばならないことになる。

文法と語彙とは1つの言語を形成する。いずれの言語もそれぞれ1つの約束ごとの体系であり、約束ごとは複数の人間——もっと一般的にはその言語を用いて意思の伝達を行なうもの——相互間に共通の知識としてもたれねばならないものである。したがってこの約束ごとを将来の不特定多数の利用者に伝達す

るためには——ここでも1つのもっと広く使用されている言語＝定義ずみの言語を用いねばならないが——詳しいマニュアルが必要となる。プログラムは一般に、ほんのちょっとしたサブルーチンでも、あるひとつの言語体系を定義してもっており、そのプログラムによって所期の目的を果たすためには、それが要求する言語による情報の提供を行なわねばならない。

以上を要約して、マニュアルの不要なプログラムを作成するために、解決しなければならない問題点を整理すると、次のようなことになるであろう。

- (1) マニュアルがもはや必要のない言語は自然言語であるから、汎用プログラムが要求する「利用上必要な情報」を、自然言語または、それから容易に連想できる言語を用いて伝達させることが望ましい。(簡略自然言語)
- (2) 汎用プログラムが要求する「利用上必要な情報」の順序は、汎用プログラム自身が質問を発する形で要求するならば、予めこれをマニュアルなどで規定しておく必要はない。(質問方式)
- (3) 利用上必要な情報の内容と意味との対応は、辞書といわれる対応表によって記述されるが、こうした対応は必ずしも必要ではなく、その段階で表現しうるすべての意味を列挙して、そこから適当なものを選ばせるといった方法で、十分用は足せるはずである。(列挙方式)
- (4) プログラムの要求する言語において備えられていない機能の1つに、抽象的な表現を指摘することができる。FORTRANのような言語にはサブルーチンや関数、あるいは配列などという形での抽象が若干用いられているけれども、汎用プログラムの言語にはこのような抽象を用いることのできる例は稀である。抽象は同じことのくり返しを避ける方法であり、これによって繁雑で退屈な作業から利用者を解放することができる。(再帰使用)

これら4つの機能を汎用プログラムのコンパイル部分にもたせることにすれば、その汎用プログラムを使用するにあたっての予備知識、したがってこれを得させるためのマニュアルは大幅に省略することができ、それらの機能をうまく活用するならば、マニュアルなどは全く不要ということにもなるであろう。

そこでこれらの機能を、すでに完成している簡易汎用作表プログラム **CROTAB** にもたせた場合の効用について、以下に解説する。

CROTAB は本経営機械化シリーズ17に紹介した通り、作表のための種々な情報をコントロールカードから読み込んだ上、予めこれをコンパイルし、これを作業実行のときに参照できる内部表現に改めるような方式を採用している。仕事の性質上この方式の方がやりやすいからであることはいうまでもないが、上に述べたような4つの機能を通して、直接内部表現を作り出すことのできる新しいコンパイラを、古いコンパイラにおきかえるだけでこのことは実現する。

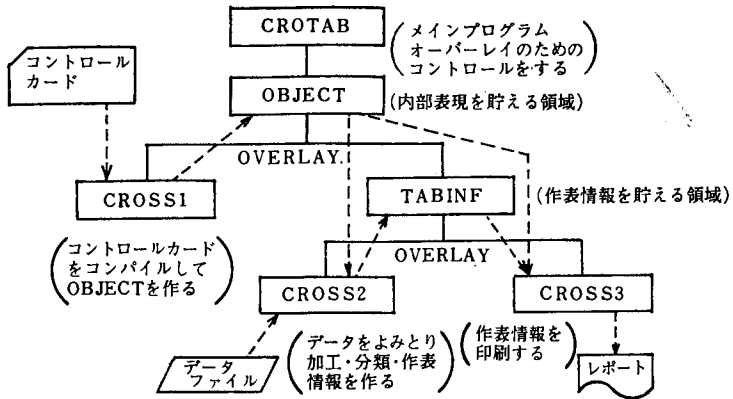
なおこの4つの機能をもつコンパイラを、コントロールカードの逐次実行方式の汎用プログラムに応用することもできる。この場合にはコントロールカードのコンパイル部分は実行時に行なうことが多いので、対話方式によってコントロールカードそのものを作り出す作業を行なうようなプログラムを作ることが、そこでのやり方になるであろう。しかし、こうすることによって逐次実行方式は、プレコンパイル方式に改められることになるが、そのおかげで4つ目の機能＝再帰使用が容易になることは強調されるべきである。

2. Dialogue-CROTAB の概要

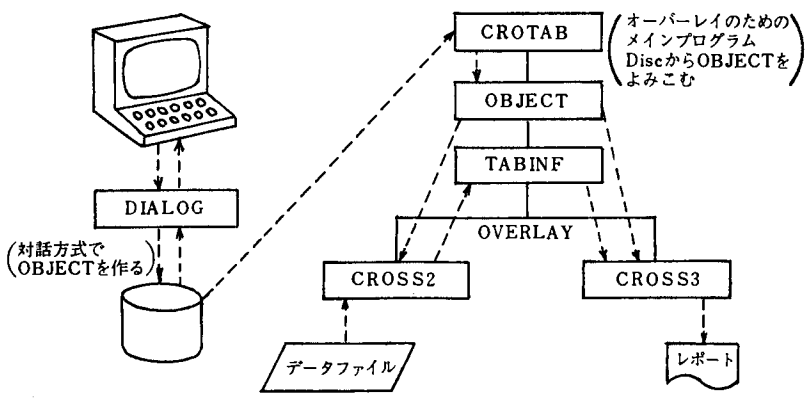
Dialogue-CROTAB は **CROTAB** のもっているサブルーチンの1つ、**CROSS1** を新しいサブルーチン **DIALOG** に置きかえるだけで実現する。ただし **CROTAB** を実行する場合には、かなり大きい記憶領域を占有することと、対話方式によって情報をインプットするにはかなりの時間を要するのが普通であることのため、**DIALOG** は別の Job として単独に動かせることにした。

すなわち **CROTAB** は第1図のような構成をもっており、**CROSS1** はこのうちでコントロールカードを内部表現に翻訳する機能をもっていたが、Dialogue-CROTAB では、別 Job である **DIALOG** が対話をしながら作

り出した同じ形の内部表現を， Disc から取り出すだけで作業に入れるように改めただけである。このときの構成は第2図に見るとおりとなる。



第1図 CROTABの構成



第2図 Dialogue CROTABの構成

ここで忘れてならないことは， D I A L O Gは対話方式で内部表現としての O B J E C Tを作り出すというばかりでなく，すでに作られた内部表現をとり出して，さらに対話方式によってこれを修正することができ，作成された内部

表現のひとまとまりの組に適当な名まえをつけてカタログしておき、いつでもこれを参照することができる機能をもつ。ただし **CROTAB** 本体のメインプログラムでは、いつでも一定の場所に収納された内部表現のみをとり出すだけで、カタログされたものの中から名まえによって適当なものを選択する機能をもたせていないので、**DIALOG**がその都度まず働いて、必要な内部表現 (=object code) を所定の場所に収納する作業を行なうことが要求される。だから**DIALOG**は**CROTAB**の常設の窓口としても機能することになる。したがって**DIALOG**と新しい形の**CROTAB**とは2つのJobではあるが、つねにセットになって作業する。そしてこのセットにつけられた名称が Dialogue-**CROTAB**にほかならない。

3. 簡略自然言語

一般に人間は機械に対して詳細な情報を要求する一方では、人間が機械に与える情報を極限にまで簡略化しようとする傾向がある。もともと機械に与えられている機能は有限であるために、その機能のうちの1つを指摘するためには、ごく簡単な情報だけで十分である。機械のもっている機能の数を N とすれば、これを区別するためには $\log_2 N$ ビットだけの情報があれば事足りる。機械に比べれば人間の判断・動作は非常に緩慢であるから、必要もないのに多くの情報を人間に要求するのは無駄である。したがってここでも機械にはできるだけ詳しい=冗長度の高い情報を出させるけれども、人間にはより簡単な情報しか要求しないという方針をもたねばならない。

以下の記述においては Dialogue-**CROTAB**が人間との対話の相手方になるが、人間から見るとこれはキカイと呼んでも決しておかしくはないと思われるので、Dialogue-**CROTAB**と機械との混成チームのことを「キカイ」と呼ぶことにしたい。

プログラム**DIALOG**がランを始めるとまず Display 装置に、

HELLO! DIALOGUE-CROTAB IS READY;

という文字が現われる。これはただプログラム **DIALOG** がランを開始したというわずか1ビットの情報にすぎない。あるいはランをはじめるべく操作した人間にとっては、このことはむしろ当然のことであって、もはや何の情報も与えないかも知れない。しかし一方では機械は正常に動いており、また **DIALOG** が必要とする機器が他のプログラムによって **assign** されてはいない、というほどの情報はあるかも知れない。

Display 装置にはその次の行に、直ちに、

YOUR NAME = ?

と出る。これは **CROTAB** 使用者の名まえを尋ねているのであるが、この名まえは後になって、出来あがった object code につけられた名まえとともにカタログされて、他の使用者の同じ名まえの object code と区別するために用いられる。

キカイはその名まえがすでに登録済みの使用者名であれば次の質問を行なう。

NEW OR OLD ?

すでに登録済みの名まえであれば、過去に登録した object code があるはずであるから、それを参照する準備をしなければならない。もちろん登録済みのない名まえならば **NEW** にきまっているから、この質問は skip される。この質問に対する人間の答えは **N** または **O** でよい。たとえ **NEW** とか **OLD** と答えても第1字だけ見れば十分である。

上の質問に対する答えが **O** であった場合については後に述べるとして、答えが **N** であった場合のその後の対話をたどってみよう。

キカイはここでまず

TASK NAME = ?

と尋ねてくる。これは以下で input される指示によって作られる object code の参照名である。たとえ二度と使われないうようなものであっても一旦は **DIALOG** のもっている file に貯えられねばならないから、何らかの名まえを与えておかねばならない。これらの名まえは利用者名の場合も含めて、別段の記述

がない限り8文字以内であると考えていただきたい。

ここでキカイはCROTABの作業のための情報をとり入れる段階に入る。最初に data file がどこにあるかを聞いてくる。

FROM CARD OR DISC OR TAPE?

これに対してはCまたはDまたはTを与えればよい。CROTABはinputの媒体として data set をCARDなら5, DISCなら21, TAPEなら11を用意している。次いで

NUMBER OF RECORDS=?

と尋ねてきたならば、実際のレコード数またはそれ以上の数値を整数で与えてやればよい。もしこれを無視して何も与えなければ、キカイは自分で20,000という数値を読みこんだのと同じ処理をする。さらにキカイは

FORMAT=?

と尋ねてくるから、これに対してはデータの形式をFORTRANの表示法で与えてやらねばならない。FORMATは左カッコではじまって、右カッコで終るが、もしそれが欠けていればキカイは自動的にこれを追加する。次いでキカイはFORMATの中のデータ個数を数えて、

NUMBER OF ITEMS IS △△, OK?

と尋ねてくる。もちろん△△の部分に数字が入っているわけだが、これに間違いなければYESまたはY, 間違っておればNOまたはNと入れてやる。Yなら次へゆくが、Nならば人間の入れたFORMATが間違っているはずであるから、もういちど

FORMAT=?

ときいてくる。

このような対話は人間にとっては決して不案内なものではないであろう。質問の意味は少なくともCROTABを使用して何かをやろうという目的をもって、いるほどの人間ならば容易に理解できるはずであるし、その答え方もごく簡単な説明を聞くだけで十分であろう。

ここでキカイが人間に対していろいろな質問をする場合に、DIALOGが

用いているサブルーチンについて紹介しておこう。このサブルーチンは **ACCEPT** という名まえで、次のような機能をもたせてある。

その Calling Sequence は

CALL ACCEPT ('literal string', **K**)

という形で、display 画面の次の行の先頭から literal string を表示し、**K** の値が 1 なら文字列を、2 ならば数字を読みこんで帰ってくる。文字列は Common array **NAME** (1000バイト) に、数字はこれを実数になおして Common real variable **VALUE** に入れる。ただし **CALL** 文中の literal string の中に ? がなければ、何も読み込まずに帰る。literal string の中の ? または ; までを表示し、? があればそこにカーソルをもどして人間からの情報を受けとるので、人間がある情報を与えたあとの画面は、たとえば

YOUR NAME=SHIMOJO

のような形で残ることになる。**SHIMOJO** の **S** のところに ? があったのであるが、それは人間の input によって消えてしまっている。

4. 質問方式による順序の制御

次いでキカイは **CROTAB** という **N** カードの情報に要求する段階に入る。**N** カードは上で与えられた **FORMAT** で表現されている item の 1 つ 1 つに、印刷用の名まえを与えるものである。Dialogue-**CROTAB** ではここで

GIVE ME NAMES OF ITEMS ;

と表示した後

NAME OF ITEM NO.1=?

で人間の input を待つ。これにやはり 8 文字以内の名まえを与えると、さらに

NAME OF ITEM NO.2=?

ときいてくる。これが item のコ数だけくり返される。この数を以下の説明のために n_1 としておこう。こうした質問を n_1 回くり返すと **N** カードに関する

限り、キカイは満足して次にゆく。

次はCカードである。Cカードは item の加工を行なうものであった。ここでは

DO YOU WANT TO MAKE NEW ITEM?

という質問が出る。これにYESまたはYと答えると、キカイはさらに

NEW ITEM NO.=?

ときいてくる。これは加工によって作られる item の番号であり、加工算術式の左辺にくるべき番号である。したがってこの番号 (n_2 としよう) が わかると、次にキカイは

$n_2=?$

ときいて人間の input を待つ。もちろん人間は必要な算術式をこれにつづけて書き込んでやればよい。item の最初の n_1 コにはすでに名まえが与えられているが、 $n_1 + 1$ 番目以降には名まえがない。また n_1 以下の番号のものでも名まえを変更したいことがあるので、ここで

NAME=?

とキカイは尋ねる。すでに名まえがあるから不要なときはこれを無視してENDキーを押してやればよいし、名まえを与えたければそこで8文字以内の文字列を与えてやればよい。

こうして加工方法が指定されると、キカイは再び

DO YOU WANT TO MAKE NEW ITEM?

ときいてくるから、さらに別の加工を指定するためにはYを、それがもう必要でないならNOまたはNを与えてやる。Nが与えられたときキカイはCカードを終って次にゆく。

次にゆくとキカイは

WHAT DO YOU WANT TO DISPLAY?

ときくので、実行時に加工された item をレコードごとに印刷してほしいときにはY、それが必要なければNを input する。Yのときキカイはさらに

NUMBERS=?

ときいてくるから、これに対しては印刷したい item 番号をカンマで区切って並べ、最後にセミコロンをつけて与えてやればよい。ここで与えることのできるのは10コマまでであること、CROTABと同じであるので、それよりも多く与えたときには、

```
NUMBER OF DATA MUST BE LESS
THAN 11;
```

と表示し、もういちど

```
NUMBERS=?
```

ときいてくることになっている。

このようにプログラムに与える情報の順序あるいはそれぞれの規格などについての約束=文法は、質問方式を用いることによって簡単に制御することができる。人間は質問に答えるだけで文法を逸脱することはできないようになっているし、もしそうした違反があればその都度注意され、改めなければ次へは進めないことになる。したがって文法についての知識がなくても十分にやってゆける。これがマニュアルを不要にする1つの重要な要件であるといえよう。

5. 列挙方式による辞書の代用

データの加工がすむと分類の段階である。そこでキカイがいきなり

```
WHAT TYPE OF CLASSIFICATION DO
YOU WANT?
```

ときいてきても、何の予備知識もなければ答えようはない。こういうとき一般的にはどう答えればよいのかを相手に問い返すことになるが、そこは相手はキカイだから、ただ?と打ってやればよいことになっている。つまり質問に対して質問が返ると、答え方がわからない、つまりメニューをよこせという意味だとキカイは考えねばならない。そこでキカイは分類方法の一欄表と、それぞれを指摘するときの記号との対応表、すなわち辞書を表示する。

```
A: CLASSIFY BY ITEM VALUE AS IT IS;
```

E : CLASSIFY INTO EQUAL RANGES ;
B : CLASSIFY BY GIVEN BORDERS ;
S : CLASSIFY BY SPECIAL VALUES ;
G : CLASSIFY INTO GROUPES OF ITEM
VALUES ;
P : CLASSIFY INTO GROUPES OF
CLASS ;

これ以上のことは画面には表示できないので、とりあえずはこれだけにとどめておくが、このうちどれかを選択し、実際に一回使ってみるとこれらの意味はすぐに理解できるはずである。

キカイはこの6行を表示したあと再び

WHAT TYPE OF CLASSIFICATION DO
YOU WANT ?

ときいてくる。これにそれぞれの行の頭の1字を返事してやればよいことはいうまでもない。6つのうちどれかを指摘してやると、その後の対話で、利用者が望んでいる分類方法の詳細がわかってくる。ただしこれをすべての場合について説明するのは繁雑にすぎるので、以下ではほんの数例のみを示して、あとは想像にまかせることにしたい。

いまEをえらんで指摘したとしよう。Eは等間隔分類である。まずキカイは

ITEM NO.=?

と尋ねる。これは上の6つのうち、P以外についてはすべて同じである。Pの場合だけは分類すべき対象がitemではなく、すでに分類された区分値についてなされるので、

CLASSIFICATION NO.=?

ときくことになる。CLASSIFICATION NO.については上の質問に答えると、分類指定の回数を通し番号としてつけた上で、キカイが自動的に追加してくれる。すなわちE分類を指摘したあと、ITEM NO.=? という質問に5と答えたとする、そのあとの画面は

ITEM NO.=5:CLASSIFICATION NO.=1;

となって画面に残ることになる。

E分類で必要な情報は item 5の値の最小値, 最大値および区画数である。もちろん最小値, 最大値は item 5 の実際のそれである必要はなく, いま分類しようとしている目的によって, それは任意に定めることができる。最小値以下, 最大値以上の値をもつデータはここでは結局採用されないことになる。したがってキカイは,

MINIMUM=?

MAXIMUM=?

NUMBER OF CLASSES=?

とたてつけに尋ねてくるから, それぞれに逐一答えてやることによって目的は達せられる。さらに各**CLASS**に名まえを与えてやらねばならないので, キカイは

**DO YOU GIVE ME LABELS OF
CLASSES?**

としたあと, Yなら

NO.1=?

などと, 上で指定した**CLASS**の数だけの**LABEL**を要求してくる。このとき, もし名まえは適当につけてくれという意味でNと答えると, キカイは適当に名まえをつけてくれる。たとえば上の**E**分類の場合は, 各区分の最低値を計算し, これに波形~をつけたものを各区分の名まえとする。名まえをつけるとキカイは先の

**WHAT TYPE OF CLASSIFICATION DO
YOU WANT?**

に帰り, 再びそのあとをくり返すことになる。もちろんここでNの応答が出ると次に進む。

いまひとつの例として**B**の場合について述べておこう。**B**は境界値を与えてデータを区分するしかたである。上の質問に対して**B**と答えたとき, キカイは

ITEM NO.=?

ときいてくる。そしてその次に、

NUMBER OF CLASSES=?

という問いに n_3 という答えを与えると、

BORDER NO.0=?

BORDER NO.1=?

から順に

BORDER NO. n_3 =?

まで、計 $n_3 + 1$ 回の質問をくり返す。これらに答えるたびに、同じ行の右側に、たとえば

BORDER NO.0=3000 LABEL=?

という質問が出る。これらに答えてゆくと、画面には境界値とそれにつけられた LABEL 名とが残ってゆくことになる。もし LABEL の質問を無視して、いきなり END を押してやると、BORDER として与えた数値に波形をつけてこれを LABEL 名とする。したがってそのときの画面は

BORDER NO.0=3000 LABEL=3000~

という形になる。もちろん最後の

BORDER NO. n_3 =?

のあとでは LABEL を尋ねる必要はない。

分類につづいて最後の作表指示の段階がくる。ここでも同じような質問が発せられる。

WHAT TYPE OF TABLE DO YOU WANT?

もちろんここでも表の種類に不案内な向きに対しては、? という応答に対して、次のようなメッセージを表示する。

DEPTH VERTICAL HORIZONTAL

- 1. CLASS BY CLASS BY CLASS**
- 2. CLASS BY CLASS BY ITEM**
- 3. CLASS BY ITEM BY ITEM**

4. MULTIDIMENSIONAL TABLE

以下ではキカイが発する質問のみを列挙すれば十分であろう。

作表指定が1の場合は

CLASS NO. FOR DEPTH=?
 CLASS NO. FOR VERTICAL=?
 CLASS NO. FOR HORIZONTAL=?
 WHAT IS CONTENT?

ここで?と問い返すと次のリストがでる。

N: NUMBER OF CASES;
 ITEM NO. AND A: TOTAL OF THE ITEM;
 ITEM NO. AND M: MEAN OF THE ITEM;

さらにMの場合は

TABLE NO. OF NUMBER OF CASES=?

ときいて平均値の計算に必要な N-TABLE の番号を知る。

なおTABLE NO. は

WHAT TYPE OF TABLE DO YOU WANT?

に何らかの答を出すたびに、通し番号で与えられ、いちいち表示され画面に残るから、あとで参照することは容易である。

表の種類2を指定したときも上と同じように、

CLASS NO. FOR DEPTH=?
 CLASS NO. FOR VERTICAL=?

ときいた後、ヨコ項目について、

DO YOU WANT NUMBER OF CASES?

さらに

ITEM NO. AND A OR M OR S FOR
 COLUMN2=?

と尋ねる。ここでも答え方がわからないと?を与えれば、

A: TOTAL;

M : MEAN ;

S : STANDARD DIVIATION ;

とメニューを出した後、もういちど

**ITEM NO. AND A OR M OR S FOR
COLUMN 2 = ?**

ときいてくる。もちろん最初の**NUMBER OF CASES**に**NO**の答を出していると**COLUMN 2**は**COLUMN 1**である。上の質問で**NO**であるのに**M**が出てきたときは、計算ができないのでそのことを知らせた上、

DO YOU WANT NUMBER OF CASES ?の質問からくり返すし、**S**がでてきてその直前が同じ**ITEM NO.**の**M**でなければ、そのことを知らせた上同じ質問をくり返す。これらのことは**CROTAB**の機能を見ていただければ明らかであろう。

分類および作表の指定方法についてこれ以上詳しく述べる必要はあるまい。要はこのような対話の重なりによって、次第に**CROTAB**のコントロールカードに代る情報が誤りなく伝えられてゆく。そしてその途中で文法チェックが行なわれ、キカイが満足するまで完全な指示ができあがってゆく。初めての利用者は簡単すぎる質問やメニューに当惑するかも知れないが、一度実行まで試みればあとは全くスラスラと事が運ぶはずである。

6. 名まえによる旧指示の再利用

スラスラと事が運ぶとはいっても、以上で述べてきたような対話を、その都度延々とくり返すことは、利用者にとって非常に退屈なことである。すでに与えた指示に何かの論理的な誤まりがあって思っていた通りの結果が出なかった場合とか、あるいは思い通りの結果は出たけれども、いまひとつ別の表を作りたいと思う場合など、同じ指示が少々の修正だけで再び三たび使われることは常である。したがってすでに与えた指示、したがってそれから作られている内部表現をもういちど引っ張り出してきて修正した上再利用するということ

が可能でなくてはならない。このために Dialogue-CROTAB では、以上で見てきたような一群の指示に対して名まえをつけ、名まえが異なる限りすべて別のものとしてカタログし、いつでも参照できるような機能をもたせている。

Dialogue-CROTAB がランを開始したとき、

```
HELLO! DIALOGUE-CROTAB IS READY;
```

という文字が表われ、そのあと直ちに

```
YOUR NAME=?
```

と尋ねてくる。これに答えてやるとさらに、

```
NEW OR OLD?
```

ときいてくるから、すでに与えたものがあれば、OLD または O とのみ答えてやればよい。

するとキカイは、

```
TASK NAME=?
```

ときいてくるので、すでに与えた名まえを与えればよい。YOUR NAME と TASK NAME とは2つ1組で1つのユニークな指示群= TASK を代表するので、両方ともすでにカタログされたものでなければ、キカイにはどうすることもできないことはいうまでもない。YOUR NAME で与えた名まえがカタログになれば、新しくこれを登録するが、OLD といいながら TASK NAME が記憶になれば、キカイはここで

```
SORRY, I DON'T KNOW XXXXX;
```

とそこで与えた名まえを伴って応答する。しかしそれがミスタイプであるかも知れないので、

```
TASK NAME=?
```

と再び尋ねることになる。

過去に与えた TASK NAME を利用者が忘れたときは、これに? の応答をしてやると、同じ利用者が過去にカタログした TASK NAME をメニューとして出してくる。これを使って上の問いに答えることはもはや簡単であろう。

YOUR NAMEも**TASK NAME**も一致するものがあつたとき、キカイはその名まえでカタログされている内部表現をとり出してくる。内部表現はいわばプログラムのオブジェクトであるから、これを利用者が与えたままのメッセージに戻して表示することはできないが、そのうちのどの部分を修正する場合でも、その場所をみつけることは決してむずかしいことではない。

ここでキカイが

NEW TASK NAME = ?

ときくのは、すでに与えた指示群 = object code に、若干の修正を加えたものに別の名まえをつけるかどうかを尋ねているのである。これによって利用者がすでに与えたものを後に再び小修正して使う意思があるかどうかを知り、もしその意思があるならそれを残しておかねばならないので、今作ろうとしているものは大部分が前のもののコピーであっても、新しい名まえのもとにカタログしておかねばならない。もちろんもしここで前の名まえと同じ名まえを与えると、古いものは消されて新しいものに入れかわってしまう。

次にキカイは

WHAT STATEMENT DO YOU MODIFY ?

と尋ねるが、これに答えるしかたがわからなければ？を入れてやると、

**F : FORMAT AND SPECIFICATION OF
DATA ;**

N : NAMES OF ITEMS ;

D : DISPLAYING ITEMS ;

C : COMPUTATION OF ITEMS ;

R : CLASSIFICATION DESCRIPTION ;

T : TABLE DESCRIPTION ;

と表示する。もちろん旧指示にDカードが含まれていなければ、Dの行は出ない。これらは旧指示の object code を見ながら表示されるからである。

たとえばこれにCと答えたときには、旧指示 object code に含まれている左辺の item の番号を順番に表示する。たとえば

5 = ?

と表示されたとき、これを変更したければこれに続けて新しい算術式を書いてやればよいし、修正が必要なければそのまま **END** キーを押せばよい。またこの式を削除したければ空白を押したのち **END** キーを押してやればよい。ここで旧指示の算術式を表示するのはもっと親切ではないかとも考えられるが、旧指示はすべて **object code** になっているので、これを元に戻す作業を行なうか、元の式をそのまま記憶しておくか、いずれも厄介なため現段階では採用していない。**object code** は逆ポーランド表示になっているため、元に戻すとはいっても、必ずしも元の通りにはならないかも知れない。

旧指示に含まれる計算式がすべて終了すると、キカイは

ANY OTHER COMPUTATION ?

ときくから、Yと答えてやると、

ITEM NO. = ?

と尋ね、これに新しく定義しようとする **item** の番号 (n_2 とする) を与えると

$n_2 = ?$

と尋ね、さらに

NAME = ?

ときく。これらは前に見たのと同じである。

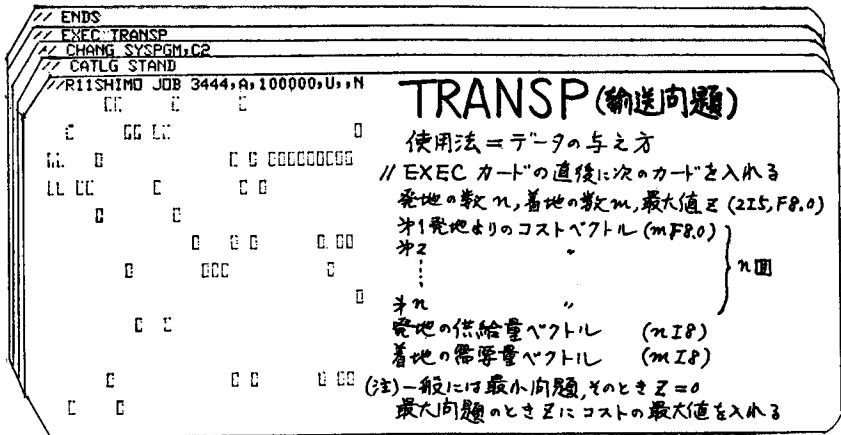
多くの場合、旧指示を変更する必要があるのは、**item** の計算や分類ではなく、**Table** の定義である。そこで分類方法の指示でも、上のようにごく簡単に修正ならびに削除、追加が行なわれたあと、**Table** に関する指示については新しい指示を受けつけるときと同じプログラムを用いてよいと考えられる。

したがって以上で新旧の指示の与え方の大要はすべて説明されたことになる。ひとつの **TASK** についてのすべての指示が終了すると、これを利用者名および **TASK NAME** との下にカタログすれば、プログラム **DIALOG** の仕事は終了する。**DIALOG** が **CROTAB** と共用している **File** にこれを収納し、今定義された **TASK** の利用者名と **TASK** 名とをその中の一定の

場所書きこんでその Job を終了すれば、次にランを開始した CROTAB のメインプログラムで、その名まえを知り、その object code をとり出して直ちに作業を開始することができる。

7. あとがき

プログラムを作った人間にとって、そのプログラムを使用することはとても簡単なことである。しかし人の作ったプログラムを使用する人間は、そのプログラムの目的は十分理解していても、それを使用するときに、処理を受けねばならない原データの順序や形式、さらにそのプログラムに備えられているいくつかの機能を利用するために必要とされる情報など、多くの知識をもたねばならない。そしてそうした知識を与えるために用意されるのがマニュアルである。



するデータや、プログラムの機能を使いわけのための情報の与え方を指示するマニュアルは、上とは全く別の形で保管されているのが普通である。

簡単なプログラムならば、登録されている **object code** を呼び出すための少数枚のコントロールカードの表面に、そのプログラムを使う場合の必要最小限の注意事項を書き込んでおけば、プログラムとマニュアルとの連絡を確実にとることができる。たとえば第3図のごとくである。

ところがこれ以上に大きいプログラム、またはもっと複雑なマニュアルを用意しなければならないプログラムにあっては、このような芸当はとてできなくなる。そこでマニュアルもいっしょにプログラムに入れてしまえばよいということになり、このような仕儀となったわけである。その結果 **Dialogue-CROTAB** のコントロールカードには、ただ

Dialogue-CROTAB (対話式簡易作表プログラム)
使用法—作表指示の与え方
このカードデッキを **READER** からよませ、端末 **DISPLAY** または **TYPEWRITER** の前に坐り、質問に応答すること。

と書き込んでおくだけでよい。

このような考え方はすでに **SPEED** (視覚プログラミング・システム) において使用されているものであるが、**CROTAB** のように予めコントロール・メッセージをコンパイルした後に実行するタイプのプログラムでは、上のようなやり方が容易になる。**SPEED** に見られる考え方をもっと徹底的に追究し、さらに対話の効果を生かすように試みたのがこれである。**TS-LSMPACK** や **TS-STRAP** などの開発にあたっての、杉浦一平和歌山大学教授や定道宏神戸大学助教授らのアイデアをここに利用させていただいたことを深謝する。

参 考 文 献

- (1) 「TSS総合アプリケーションシステム」関西情報センター，昭和49年3月。
- (2) 定道宏・正井樹「視覚プログラミング・システム——SPEED——」神戸大学経済経営研究所『経営機械化シリーズ』16，1975。
- (3) 下條哲司「汎用簡易作表プログラムCROTABについて」神戸大学経済経営研究所『経営機械化シリーズ』17，1976。

マーケティング戦略教育ゲーム*

— SMS Sの解説 —

布 上 康 夫
萩 野 典 宏

目 次

- I まえがき
- II マーケティング戦略教育ゲームの構造
 - [1] 経営構造
 - [2] 市場構造
 - (1) 外生的市場構造要因—市場投入要因—
 - ① 企業数
 - ② 1社平均市場潜在需要量
 - ③ 景気指数
 - ④ 単位変動費
 - (2) 外生的市場構造要因—市場既設要因—
 - ① 市場数
 - ② 有効意思決定の上限値・下限値
 - ③ マーケティング効果モデルの係数
 - ④ その他のモデルの係数
 - ⑤ 有効需要繰越量の配分比重
 - (3) 内生的市場構造要因
 - ① 変動的市場構造要因
 - ② 期首市場構造要因

* この研究の一部は昭和52年度文部省科学研究補助金試験研究(2)「経営経済データ・バンクの拡充とデータ・バンク管理システム」(代表者・能勢信子教授)によってなされたものである。

Strategic Marketing Simulation System の開発に関しては、神戸大学経営学部・荒川祐吉教授、田村正紀助教授より御指導を賜った。そのコンピュータ・システムの開発においては、神戸大学経済経営研究所・定道宏助教授より有益な示唆を受け、甲南大学電子計算センターより電子計算機(I BM370-125)の利用に関して多大の協力を得た。ここに記して厚く感謝の意を表したい。なお、このシステムの不十分な点はわれわれ筆者の未熟さによるものである。建設的な批判を賜り改善していきたい。

- [3] 意思決定システム—企業投入要因—
 - (1) 経営方針
 - (2) 市場調査
 - (3) 価格
 - (4) 販売促進
 - (5) 研究開発
 - (6) 生産量
 - (7) 流通
 - (8) 資金運用
- [4] 市場行動システム
 - (1) 企業有効販売促進と企業有効研究開発
 - (2) マーケティング効果
 - (3) 産業需要創造効果と市場有効需要量
 - (4) 企業需要創造効果と企業有効需要量
 - (5) 小売交渉効果と有効市場在庫量
 - (6) 需給適合
 - ① 超過需要量
 - ② 超過需要量の再配分
 - ③ 企業実際販売量
- [5] 経営成果システム
 - (1) 期末経営構造要因
 - (2) 期末財務構造要因
 - (3) 損益計算書
 - (4) 貸借対照表
 - (5) 経営分析
 - ① マーケティング管理分析
 - ② 在庫管理分析
 - ③ 財務管理分析
 - (6) 経営総合評価
- [6] 市場成果システム
 - (1) 期末市場構造要因
 - (2) 市場構造分析

Ⅲ むすび

付 録

I マーケティング戦略教育ゲームのフローチャート

Ⅱ 意思決定表

Ⅲ プログラム・リスト

Ⅳ 実行結果

I ま え が き

ビジネス・ゲームは、Andlingerが「**Business Game-Play One**」(Harvard Business Review, 1958年)を発表して以来、数多くのものが各国のビジネス・スクールや企業で開発されている。日本でも、昭和33年に慶応大学の慶応ハーバード高等経営学講座で紹介されてから、数多くのセミナーや企業で開発され、教育訓練や社内研修などに利用されている。これらのビジネス・ゲームは、主として意思決定のシミュレーションや経営管理に必要なテクニックの習得を目的としたものであり、ケース・メソッドの形で経営学の教育に利用すれば、**現実**に即した**事例の提起**が可能であること、**複雑な企業の意思決定を行う機会**をゲームの参加者である学生に提供できること、**意思決定および結果の評価に適切な経営科学手法を教える機会**の提供が可能になることから、教育上での利用の効果はきわめて大きい。さらに、ゲームの参加者を産業別に十分な経験を踏んだ意思決定者に限定した場合に、経営構造、市場構造や需給動向、景気変動などの変化に伴う**意思決定パターンの産業別参加者の比較分析を通じて意思決定プロセスの研究**にも十分貢献しうる。このように、ビジネス・ゲームをケース・メソッドの一環として教育に、あるいは、意思決定プロセスの研究として捉えるとき、そこに展開される経営学上の問題提起、企業の経営構造、市場行動のメカニズムなどは十分に現実を反映したものでなければならない。しかし、現実の企業経営構造は複雑で、数個の変数では十分に説明できないし、意思決定のプロセスやその方法も多種多様である上に、産業界では、不況時に価格や生産量の協定を結んだり、企業内部では、合併・増資などの非定期的な意思決定を行う。これらの一部でもルーチン化してビジネス・ゲームに組込むとすれば、企業経営構造は、きわめて複雑なものになる。さらに、市場行動のメカニズムは、現実の流通過程に整合したものであり、その諸要素の相互作用は、現実の市場行動や、マーケティングの理論と矛盾したものであってはならない。これらの複雑な企業経営構造と市場行動メカニズムを持ったビジネス・

ゲームは、電子計算機の利用を前提とすれば可能ではあるが、逆に、複雑なビジネス・ゲームになればなる程、設定せねばならない入力や結果の出力が多くなりすぎ、ビジネス・ゲームのゲーム性が損なわれる。

これらのことを考慮して、今回新たに開発した**マーケティング戦略教育ゲーム**（以下、**マーケティング・ゲーム**）の特徴は次のとおりである。

1. 可能な限り入力と出力を少なくし、また、ビジネス・ゲームのゲーム性を損なわないために「**マーケティング志向ゲーム**」とした。そこで意思決定要因をできるだけ単純化し、マーケティング行動要因に焦点をあてた。
2. マーケティング行動を製品戦略を中核戦略とするマーケティング諸戦略の有機的体系、すなわち「**マーケティング戦略空間**」として捉えている。この認識を基礎として、マーケティング効果、需要創造効果、小売交渉効果などを導出する水平的競争過程、垂直的流通過程を表わす「**計量的市場行動モデル**」を構築しているところが他に類を看ない特色である。
3. 市場行動を現実に一層即したものにするために、「**景気指数**」を導入した。この景気指数は、マーケティング効果の算定モデルの係数（マーケティング反応係数）などの内生的市場構造要因の算定に利用され、重要な役割をはたしている。
4. マーケティング行動を中心としたものであるが、「経営方針」・「経営計画」に基いて決定される各々の意思決定要因間の相互関連体系を一つの「**意思決定システム**」（図Ⅱ-3参照）として認識した。さらに、「意思決定システム」は相互関連のある7つのサブシステムから構成されており、このサブシステムの一つ一つが単独のビジネス・ゲーム（価格決定・ゲーム、販売促進・ゲーム、在庫管理・ゲームなど）として活用できる。

Ⅱ マーケティング戦略教育ゲームの構造

このマーケティング・ゲーム・システムの作成過程においては、次のような論理体系を基礎とした。すなわち、**期首構造**を所与として**行動過程**が展開されその結果として**期末構造**が生成する。これは企業レベル（マイクロ）と市場レベル（マクロ）という2つの階層で扱えられる。つまり、**期首経営構造**と**期首市場構造**を所与として、**企業行動**（意思決定過程）、**市場行動**（市場組織内相互作用過程）が展開され、その展開過程の結果として**期末経営構造**、**期末市場構造**が生成する。期末構造は次期・期首構造となり、次期意思決定の前提条件となる⁽¹⁾。この論理体系は時間を含む因果サイクルとして認識されるものである。**企業統制者**は**企業目的**ないし**経営方針**（たとえば、利潤率極大化方針、あるいは、市場占有率極大化方針など）に従って、意思決定パターンの方向づけを行なう。企業統制者はある時は構造維持的に、またある時は構造革新的に意思決定パターンの方向づけを行なう⁽²⁾。この意思決定パターンの方向づけは経営構造の推移、および、市場構造の推移との関連において行なわれる。

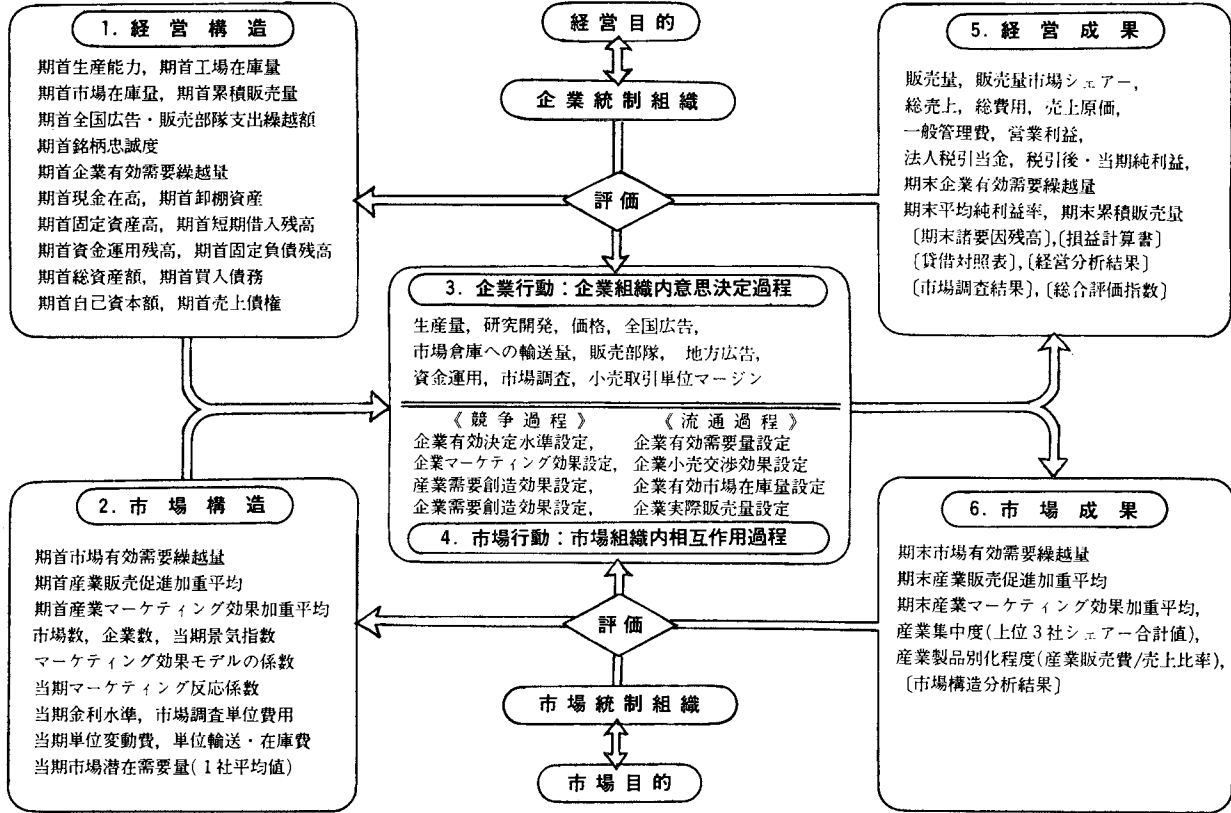
市場統制者は産業組織内の競争が促進され、かつ、安定的価格において需要充足が促進されるように市場行動パターンを方向づける。しかしながら、このゲームにおいては、市場統制者はゲーム開始点においてゲーム終了点までの市場構造を設定しておく、ゲーム中間点においては市場組織への介入（ゲーム・システムの変更；市場構造条件の変更）の必要はない。

このマーケティング・ゲームは以上のような基礎システムのもとに作成された。第Ⅱ－1図はその概要を示したものである。

(1) このゲームでの「1期」とは「四半期」を指す。

(2) 参加者の許容基準 (criteria of participant acceptance) (Greenlaw, P. S., et al. [2], p. 122) に合わせうように、一方ではインプット要因（意思決定要因、市場構造外生要因）の個数を必要最少限にとどめ、「意思決定システム」の単純化を図り、他方では、システムの現実性を高めるために、内部システムの拡充をはかり、「市場システム」、および、「経営成果システム」においては、理論・現実・シミュレーションの相互関係に留意しつつモデルの現実性の向上に努力した。

図Ⅱ-1 マーケティング戦略教育ゲームの概要



[1] 経営構造⁽³⁾

経営構造は期首経営構造諸要因の体系的集合である。第Ⅱ-1図に示しているように、期首経営構造要因は、生産・流通関係の期首要因、マーケティング関係の期首要因、および、貸借対照表関係の期首要因から成る。

期首経営構造要因の値はすべて前期の期末経営構造要因の値を繰り越すことによって与えられる。

ゲーミング開始期（第1期）の期首経営構造要因の値は審判団（市場統制組織）によって設定される。

第1期の期首経営構造要因の値は、新しい企業の創設ではなく、**既存企業の継続的経営**という考え方で設定される⁽⁴⁾。

第1期の期首経営構造要因の値⁽⁵⁾は次の通りである（全チーム共通）。

（単位=1000）

期首総資産 = 11780	期首生産能力量 = 200
期首現金在高 = 2000	過去の最大生産量 = 200
期首売上債権 = 1500	前期の生産量 = 150
期首資金運用残高 = 1000	期首総在庫量 = 300
期首棚卸資産 = 2280	期首工場在庫量 = 150
期首固定資産 = 5000 ⁽⁶⁾	期首市場在庫量 市場1 = 85
期首買入債務 = 1280	期首市場在庫量 市場2 = 65
期首短期借入金 = 1000	
期首固定負債 = 3000	全国広告（1期前） = 100
期首引当金（固定） = 500	同（2期前） = 100

(3) 「経営構造」の詳細については「経営成果システム」を参照せよ。

(4) Greenlaw, P. S., et. al. [2], p. 130.

(5) 金額はすべて「ドル」で表示される。

(6) 「期首・固定資産」の10%が「当期・総固定費」となる。

「総固定費」は次のように配分される。

一般管理費 = 総固定費 × 0.1

販売固定費 = 総固定費 × 0.1

生産固定費 = 総固定費 × 0.8

期首資本金（固定）	=6000	同（3期前）	=100
期首剰余金	=0	同（4期前）	=100
期首累積利潤率	=0	販売部隊（1期前）	市場1=60
期首累積販売量	=0	同（同）	市場2=50
期首銘柄忠誠度	市場1=1	期首企業有効需要	繰越市場1=0
同 上	市場2=1	同 上	市場2=0
期首累積研究開発効果	=0		

（忠誠度，開発効果の単位は1である）。

[2] 市場構造 (7)

市場構造は外生的要因と内生的要因とから成り，外生的要因は投入要因と既設要因とから構成されている。すなわち，市場構造は

外生的市場構造要因（市場投入要因）

// （市場既設要因）

内生的市場構造要因

という3つのカテゴリーの諸要因の相互関係において認識される。

(1) 外生的市場構造要因—市場投入要因—

市場投入要因は企業数，市場潜在需要量（1社平均値），景気指数から構成される。

① 企業数（チーム数）

このマーケティング・ゲームに参加しうるのは10企業（10チーム）以内であるが，筆者の経験から5企業～6企業が適当である。

② 1社平均市場潜在需要量

市場潜在需要量はゲーム・コントローラーによって設定され，各企業には知らされない。しかし，各企業が正確な予測を得たものと仮定して，ゲーム中間点までの市場潜在需要量を公開するのも1つの方法である。

(7) このマーケティング・ゲーム・システムでは「市場地域数」を2地域に，「産業数」を1産業に限定している。

市場潜在需要量は趨勢として増大的であり、第1期～第5期の平均値に対して、第21期～第26期の平均値は約2倍となる。勿論、短期的には大小の変動が組み込まれる。(第Ⅱ-2図参照)。

この**市場潜在需要量**は、そのまま企業潜在需要量に分配されるのではなく、**産業需要創造効果**により**市場有効需要量**が設定され、これに、**期首有効需要量繰越残**が加えられ、**マーケティング効果相対値**を乗じて各企業に分配される。これは、さらに、**企業需要創造効果**に基き**企業有効需要量**が決定される。この過程については市場行動システムで説明する。

1 社平均市場潜在需要量は2つの市場の各々に設定される。

③ 景気指数

景気指数は需要量の推移とはほぼ相関的であるが、必ずしも同調的ではない。景気指数が正で大きい時ほど**好況**であり、負で絶対値が大きい時ほど**不況**である。好況期には、価格よりも、広告などの販売促進支出の方が**マーケティング効果**への貢献度が大きい。不況期には、販売促進支出よりも、価格の方がその貢献度が大きく、少しの値下げでもマーケティング効果が上昇する。

この**景気指数**の2分の1が倉庫貸借料、金利、売掛・買掛率などの**内生的市場構造要因**の算定に利用されている。これは両市場に共通に作用する。

(2) 外生的市場構造要因—市場既設要因—

このマーケティング・ゲーム・システムでは、次にあげる市場構造要因に最初から特定の数値を固定的に設定している。これらは両市場に共通である。

① 有効意思決定値の上限値、下限値

価格下限=20ドル

流通マージン上限=価格/3

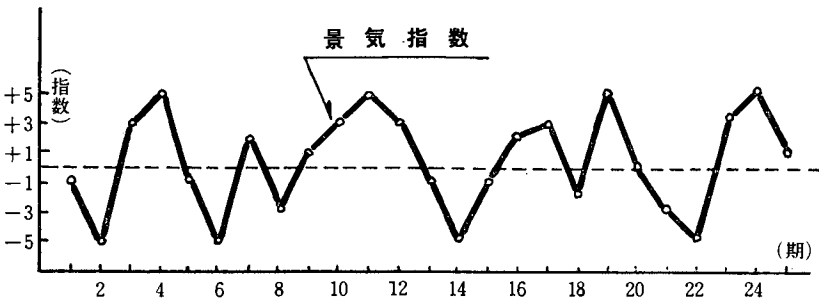
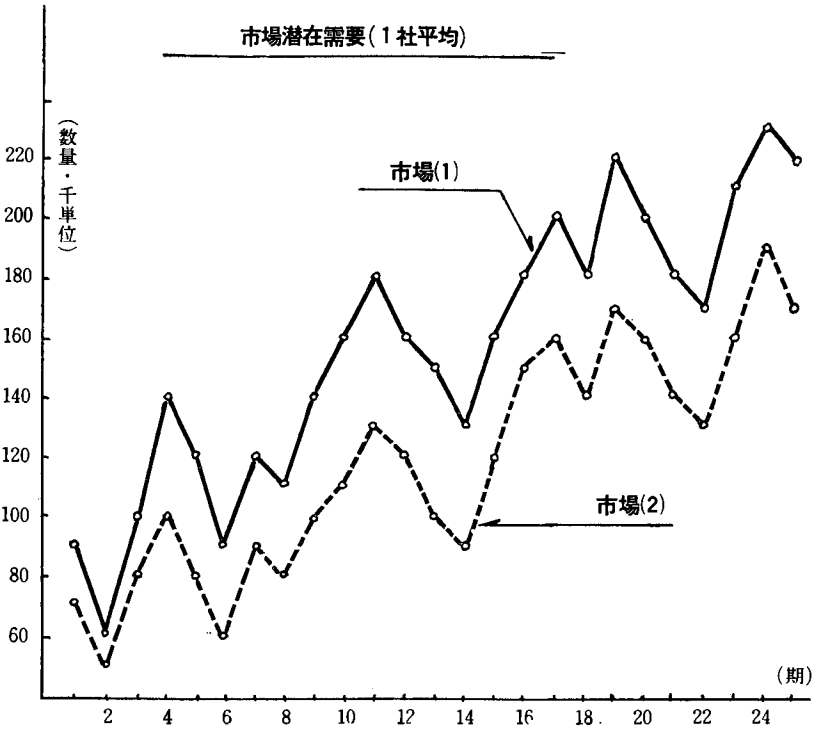
流通マージン下限=1ドル

研究開発費上限=期首産業研究開発加重平均の2倍

研究開発費下限=50千ドル

販売促進費上限⁽⁸⁾=期首産業販売促進加重平均の2倍

(8) 販売促進とは全国広告、地方広告、販売部隊である。



図Ⅱ-2 市場潜在需要、景気指数の推移

販売促進効果通減率＝期首産業販売促進加重平均を越える分は60%

生産量上限＝前期比＋20%

生産量下限＝前期比－20%

生産能力追加量＝期首生産能力の20%

② マーケティング・モデル⁽⁹⁾の係数

全国広告係数(α)＝0.375 地方広告係数(β)＝0.225

販売部隊係数(γ)＝0.300 価格係数(ϵ)＝2.0

研究開発係数(ζ)＝0.225 銘柄忠誠度係数(η)＝0.275

繰越販売促進係数(γ)＝0.250

③ その他のモデルの係数⁽¹⁰⁾

販売促進支出の繰越効果係数

全国広告；1期前係数(A1)＝0.8 2期前係数(A2)＝0.5

3期前係数(A3)＝0.2 4期前係数(A4)＝0.1

販売部隊；1期前係数(A5)＝0.6

研究開発効果定数＝1.8

企業需要創造効果係数＝0.25

企業小売交渉効果係数＝0.22

市場調査費＝6千ドル

輸送費単価＝0.5ドル

市場在庫管理費単価⁽¹¹⁾＝0.5ドル

法人税率＝0.4

(3) 内生的市場構造要因

内生的市場構造要因は、景気指数の変化に伴って変動する変動的市場要因、および、各企業の活動水準の集計から算出される期首市場構造要因からなる。

(9) 「マーケティング・モデル」については、「市場行動システム」の「マーケティング効果」を参照せよ。

(10) 各々のモデルについては「市場行動システム」を参照せよ。

(11) ただし、市場在庫量合計>生産能力の場合には倉庫貸借料が加算される。これに關しては内生的市場構造要因における超過市場在庫管理費単価の項をみよ。

① 変動市場構造要因

市場総潜在需要量 = 1 社平均市場潜在需要量 × 企業数

市場在庫総量 = 期首市場在庫量の市場合計

マーケティング反応係数 = 景気指数 × 1.5

単位変動費 = 20 + 景気指数 / 5

超過市場在庫管理費単価⁽¹²⁾ = 2 + 景気指数 / 10

固定資本単価 (生産能力千単位当り) = (35 + 景気指数 / 2) × 1000

金利係数 = (1 - 景気指数) × 0.5 / 100

資金回収費率 = 金利係数 + 0.01

固定負債利率 = 金利係数 + 0.08

短期借入利率 = 金利係数 + 0.01

資金運用利率 = 金利係数

買掛・売掛率 = 1 / (2 + 景気指数 / 10)

② 期首市場構造要因

期首市場構造要因の値は前期の期末市場構造要因の値を転記したものである。第1期の期首市場の構造要因の値は審判団によって設定される。その数値例をつぎに示す。(単位=千ドル)

全国広告費 = 100

研究開発費 = 150

地方広告費市場 1 = 40

販売部隊市場 1 = 60

地方広告費市場 2 = 30

販売部隊市場 2 = 50

期首産業研究開発費累積額 = 0

期首市場有効需要繰越残市場 1 = 0

// 市場 2 = 0

期首市場潜在需要累積量市場 1 = 0

// 市場 2 = 0

(12) これは、市場在庫総量 > 生産能力のときのみ適用される。それ以外は 0 である。

[3] 意思決定システム—企業投入要因—

マーケティング行動は、より良質の製品をより安定的価格で、より大量に供給することによって**需要家の効用**を最大化させるという**市場目的**と、品質・価格・販売促進・流通経路管理などの諸活動において競争企業よりも相対的な優位性を確立し利潤率の極大化あるいは市場占有率の極大化を達成することによって**企業組織の効用**を最大化させるという**企業目的**の達成とを志向するものである。

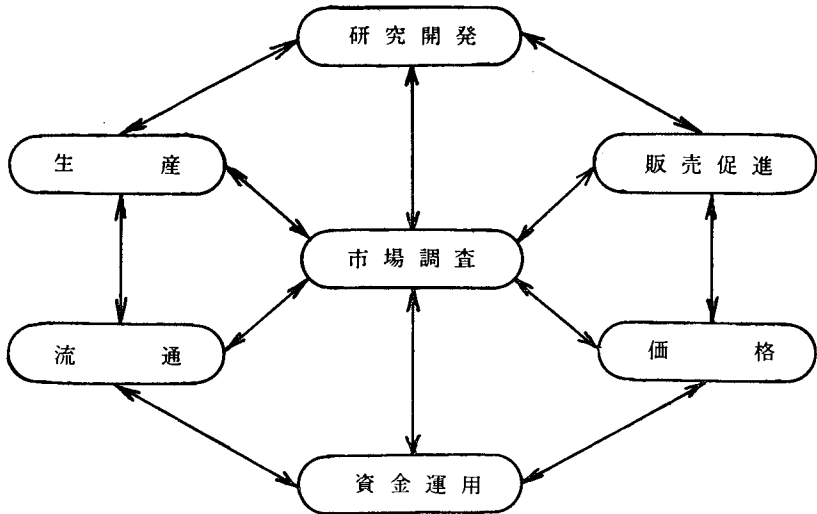
これらの市場目的と企業目的との達成を志向するマーケティング行動は諸々の**マーケティング諸活動の体系的集合**として認識される。これを製品戦略を中核とする体系として認識する時、現代における**マーケティング戦略空間**の概念が提起される⁽¹³⁾。

このマーケティング・ゲームの意思決定要因、つまり、経営方針、市場調査、研究開発、販売促進、価格、生産、流通、財務という各要因は有機的な相互関連性をもって**意思決定システム**を形成している。つまり、ある意思決定要因は、市場構造・経営構造を考慮するとともに、他の意思決定要因との関連を考慮して決定されねばならない。例えば、価格と全国広告のみに焦点をあてると、マーケティング効果を維持するには、価格を引き上げると全国広告を増加させねばならない。(図Ⅱ-3参照)。

これらの意思決定要因には、様々な次元において特徴をもつ。**全国市場レベルでの意思決定要因**は研究開発、生産、全国広告、価格、流通マージン、資金運用であり、**2つの地方市場別の意思決定要因**は、地方広告、販売部隊、工場倉庫から市場倉庫への輸送である。**繰り越し効果をもつ要因**は研究開発、全国広告、販売部隊であり、**繰り越し効果をもたない要因**は地方広告、価格、流通マージン、資金運用である⁽¹⁴⁾。

(13) 田村正紀[7], Mickwitz, G. [4] では、マーケティング・ミックスの構造の変化がプロダクト・ライフ・サイクルとの関連において把握されている。

(14) 参加チームは、各々、経営組織体である。従って、チームの構成メンバーは各々、意思決定要因を分担して予備決定を行ない、協議を経て最終決定をする、という組織的行動をとることが望ましい。



図Ⅰ-3 意思決定諸要因の相互関連（各意思決定は「経営方針」・「経営計画」に基づいて行なわれる）

なお、このマーケティング・ゲーム・システムでは、特定の要因のみを意思決定するという、「サブ・ゲーム・システム」を行なうことも可能である。例えば、次のような組み合わせが考えられる。

サブ・ゲーム名	意思決定要因
価格決定ゲーム	価格
販売促進ゲーム	全国広告、地方広告、販売部隊、流通マージン
在庫管理ゲーム	生産量、輸送量
流通管理ゲーム	生産量、輸送量、流通マージン
小売交渉ゲーム	価格、販売部隊、流通マージン

「サブ・ゲーム・システム」では必要な意思決定要因のみをデータとして与え、不要な意思決定要因についてはデータとして与えなくてよい（つまり、0でよい）。不要な意思決定要因は次のようにして、システム内部で自動的に、各社共通の値が設定される。

価格=35ドル

各販売促進＝期首産業販売促進費加重平均値

研究開発＝期首産業研究開発費加重平均値

生産量＝次々期の1社平均市場潜在需要量（両市場合計）

輸送量＝次期の1社平均市場潜在需要量（各市場とも）

流通マージン＝1ドル

資金運用＝0ドル（ただし期首現金，期首借入の状況に応じて自動的に調整される）。

（1）経営方針

企業統制者は経営成果および市場成果の状況に応じ経営方針を設定する。経営方針としては、基本的には**市場占有率極大化**あるいは**利潤率極大化**のいずれかが設定される。**市場占有率極大化方針**では、「低価格」・「高流通マージン」・産業平均以上の「販売促進支出」・「研究開発支出」・「生産量」が決定される傾向がある。**利潤率極大化方針**では「高価格」・「低流通マージン」・産業平均以下の「販売促進支出」・「研究開発支出」・「生産量」が決定される傾向がある。いずれの両方針においても、どの戦略に重点がおかれるかは意思決定者の判断に依存している。

このマーケティング・ゲームにおける**総合評価**⁽¹⁵⁾は、経営方針を明確に設定し経営方針に基いて長期経営計画を立て実行したチームが有利な結果を得るように設計されている。

（2）市場調査

市場調査は戦略的意思決定のための重要な基礎情報を与える。このマーケティング・ゲームにおいては「市場潜在需要量の推移」，および，「期首産業販売促進・研究開発加重平均値」・「価格」の情報が与えられる。調査項目は下記のとおりである。（「産業」は1社平均値で与えられる）。（*印のものには6,000ドルの調査費が必要である。

(15) このマーケティング・ゲームでは、マーケティング行動の市場目的と企業目的とを考慮して、各チームの業績を

総合評価基数＝（累積販売量シェア²＋累積純利益率シェア²）^{0.5}
のシェアによって順位をつける。

[調査項目]

価格、産業全国広告支出(*), 産業地方広告支出(*), 産業販売部隊支出(*), 産業潜在需要量(*), 企業潜在需要量(*), 企業市場占有率(*), 産業平均販売量, 企業販売可能量/潜在需要量比率。その他の市場構造分析結果。

「市場調査」情報は「経営分析」情報と結合されることによって意思決定をより有効化させる。

「市場調査」を意思決定項目からはずすこともできる。この場合, 「市場調査」情報のすべてが公平に全チームに提供される。

(3) 価格

「価格」は「マーケティング効果」と「売上額」を規定する重要な要因であり, 相対的に高い「価格」は「マーケティング効果」を小さくさせ, 「企業潜在需要量」を小さくさせる。しかしながら, 前述のように, このマーケティング・ゲームにはマーケティング反応係数が設定されており, この係数が正(好況)の時には価格効果は相対的に小さく販売促進効果・研究開発効果が相対的に大きくなる。反面, この係数が負(不況)の時には相対的に価格効果が大きく, 販売促進・研究開発効果が小さくなる。従って, 好況時には価格低下を試みるよりも販売促進を活発化させる方がよく, 不況時には販売促進を抑えて価格を低下させる方が「マーケティング効果」を向上させる。

また, 価格低下は需要量・販売量の増加には貢献するが, しかし, 「売上額 = 価格 × 販売量」であり, 価格低下は売上額増大には必ずしも有効な作用を及ぼすとは限らない。

(4) 販売促進

販売促進支出は「全国広告支出」, 「地方広告支出」, 「販売部隊(セールスマン)支出」により構成されている。これらの支出の増加は各々の特性に応じて企業のマーケティング効果を向上させ「企業潜在需要量」を増加させる。

全国広告は第1・第2市場に共通に効果を生じる。これに対して, **地方広告**と**販売部隊**は各々, 第1・第2市場に個別に支出され, 個別に効果を生じる。従って, 例えば, 第1市場において在庫量が過剰であれば, 第1市場への「地

方広告支出」と「販売部隊支出」を特に増大させることによって販売量を増大させうる。

また、**全国広告**は次期より4期間にわたって、**販売部隊**は次期のみ1期間、**繰り越し効果**をもつ。換言すれば、今期の販売促進効果に対しては、今期および過去4期間の「全国広告」、今期および前期の「販売部隊」、そして、今期の「地方広告」が作用する。

「全国広告」と「販売部隊」の支出の**繰り越し効果**は繰り越し期間が長くなるほど小さくなる。

販売促進繰り越し効果⁽¹⁶⁾は次のようにして求められる。

- 全国広告繰り越し効果 $t = 0.8 \times \text{有効全国広告 } t_{-1} + 0.5 \times \text{有効全国広告 } t_{-2} + 0.2 \times \text{有効全国広告 } t_{-3} + 0.1 \times \text{有効全国広告 } t_{-4}$
- 販売部隊繰り越し効果 $t = 0.6 \times \text{有効販売部隊 } t_{-1}$
- 販売促進繰り越し効果 $t = \text{全国広告繰り越し効果 } t + \text{販売部隊繰り越し効果 } t$

地方広告は意思決定期においてのみ効果をもち、繰り越し効果をもたない。

以上の3つの**販売促進支出**はいずれも2つの理由から「上限値」「下限値」が設定されている⁽¹⁷⁾。第1に、マーケティング戦略は競争企業間の競争過程における相互学習を通じて均等化する傾向をもつ⁽¹⁸⁾。第2に、需要家に対する販売促進活動は、産業内の平均的水準を著しく越えると説得的圧力が増大し需要家の反発を受ける可能性が大きくなる。従って、産業内の販売促進支出は競争企業間において均等化する傾向をもつ。

(16) Kotler は、「繰り越し効果」(carryover effect) を次のように定義している。すなわち、「繰り越し効果」は、今期のマーケティング支出が今期以降において販売量に対して作用する効果である。これは、マーケティング戦略の購買者の購買行動への「効果のおくれ」(タイム・ラグ)が存在すること、および、マーケティング戦略の刺激に因り高められた購買率が将来の販売量に「累積的效果」を及ぼすことに起因する。(Kotler, P. [3], pp. 121~129)。ここでは、「効果のおくれ」(タイム・ラグ)を考慮せず、「効果の累積」のみを考慮している。

(17) 「産業有効上限値」は「加重平均値」の2倍である。

(18) 田村正紀[7], p. 89。現実的には、マーケティング戦略の量的均等化が進行したとしても、その質的差別化が一層追求されることとなる。このマーケティング・ゲームではその質的差別化をモデルに組み込むことができない点に限界をもっている。

そこで、このマーケティング・ゲームにおいては、**産業販売促進加重平均**を均等化水準とし、その30%を「下限値」とし、その2倍を「上限値」として、「上限値」以上の支出は費用に計上されるが効果をもたないものとしている。また、**販売促進加重平均**を越える支出については60%のみが効果をもつと仮定している。このように、「産業販売促進加重平均」を基準として、**有効販売促進支出**が「市場システム」において算出され、その値を用いて、各チームのマーケティング効果が算出される。〔算出の方法については、[4]市場システム：(1) 企業有効意思決定値をみよ〕。

(5) 研究開発

研究開発は、「市場行動システム」においては、**マーケティング効果と企業需要創造効果**とを通じて「企業有効需要量」の増加に貢献し、「経営成果システム」においては、**単位生産費減少効果**を通じて売上原価の減少に貢献する。ただし、企業は新製品開発技術にしる、生産設備改善技術にしる、他社から技術購入することも可能である。従って「研究開発」の下限を支出すれば「研究開発効果は」は1未満にはならない。（「研究開発の下限」=50千ドル、「研究開発効果」が1未満の場合には1とする。）⁽¹⁹⁾

研究開発効果は次のようにして算出される。

$$\circ \text{企業累積研究開発費 } i_T = \sum_{i=1}^T \text{有効研究開発費 } i_i$$

$$\circ \text{1社平均累積研究開発費 } \tau = \sum_{i=1}^N (\text{企業累積研究開発費 } i_T) / N$$

$$\circ \text{1社平均累積潜在需要量 } \tau = \sum_{i=1}^T \text{1社平均市場潜在需要量 } i_i$$

(19) 以下 i = 企業 N = 企業数
 j = 市場
 t = 期 T = 到達期をあらわす。

- 研究開発効果 $i_r = \left[\frac{\text{企業累積研究開発費 } i_r}{(1 \text{ 社平均累積研究開発費 } r \times 1 \text{ 社平均累積潜在需要量 } r)} \right]$
 \times 研究開発効果定数

(ただし、研究開発効果 $i_r < 1$ のとき、研究開発効果 $i_r = 1$)

「研究開発支出」は、以上のように、必ず1以上の「研究開発効果」を生じ、有効需要量増加と売上原価減少に貢献する。しかし、「研究開発支出」には研究開発用の設備投資が必要であるという理由により、意思決定された**研究開発支出の50%が固定資本投資**として自動的に追加される。従って、「研究開発支出」がコスト増大の方向に作用する側面をも考慮せねばならない。

(6) 生産量

「生産量」の決定において特に注意すべきことは、第1に、**単位変動費・単位固定費への適合**、第2に、**前期生産量と生産能力への適合**、第3に、**流通期間と潜在需要への適合**である。

① 単位変動費と単位固定費

まず、第1に、企業は、生産能力と需要量を一定とすると、「**単位変動費**」の低い期間により多く生産することが望ましい。「単位変動費」は景気状況に依存して変化し、不況期においてより低くなる傾向にある。また、一般に、生産能力以内において「生産量」を増大させると**単位固定費**が減少する。しかし、後述するように、生産能力以上の「生産量」の決定は固定資本投資を伴い、単位固定費を引き上げることに注意すべきである。

② 前期生産量と生産能力

第2に、**生産量の変動**は1期につき**前期生産量の±20%以内**でなければならない。現実企業は特に急激に雇用数を変動させたり、原材料調達量を変動させたりすることは不可能であり、必然的に、企業が1期に「生産量」を変動させる範囲は限定される。そこで、「生産量」の変動巾を前期比±20%以内に設定した。さらに、「生産量」が過去の最高水準を越える場合には、設備投資が必要であり、期首生産能力の20%の生産能力追加が行なわれる。すなわち

- $(\text{前期生産量} \times 0.8) \leq \text{今期生産量} \leq (\text{前期生産量} \times 1.2)$

今期生産量>期首生産能力ならば

- 今期生産能力追加量=期首生産能力×0.2

だけの生産能力の追加が行なわれ、それのみあうだけの生産量が可能となる
なお、この場合、

- 期末生産能力=期首生産能力+今期生産能力追加量

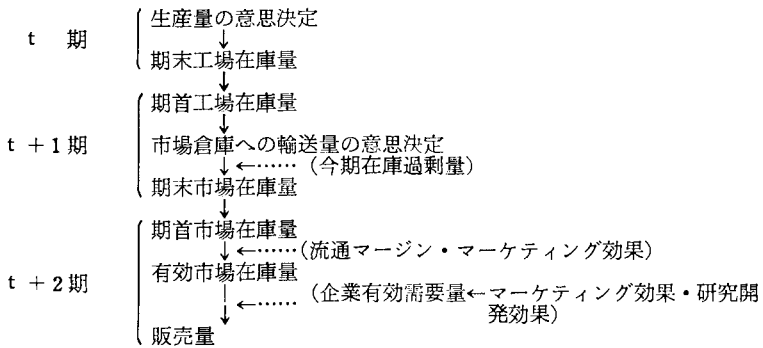
となり、生産能力追加のための固定資本投資が行なわれる。

固定資本投資は減価償却費の増加を伴うので、生産量の増大は必ずしも生産費の低下を伴うとはかぎらない。この生産能力追加（固定資本投資）は、本部

- ・ 営業・倉庫などの設備拡充を含んでいる。

③ 流通時間と需要変動

第3に、生産物の流通のために、生産から販売までにタイム・ラグがあり、



図Ⅱ-4 生産・流過程の必要期間

「生産→工場在庫」を t 期とすると「工場在庫→市場在庫」は t + 1 期、「市場在庫→小売販売」は t + 2 期となる。（流通に 2 期間必要である）。例えば第 5 期の生産量は、第 7 期に「期首市場在庫量」となるので第 7 期の需要量に適合させねばならない。（図Ⅱ-4 参照）。

(7) 流通

流通に関する意思決定は、**輸送量**と**流通マージン**の 2 つの要因から成る。

輸送量は「期首工場在庫量」から市場 1, 2 のそれぞれの「市場倉庫」への

配分・輸送に関する決定である。

流通マージンは各チーム（製造企業）が小売業者に支払う「製品1単位当りの小売取引マージン」〔両市場共通〕（価格の $\frac{1}{3}$ 以内）の決定である。

第1に、**輸送量**は、「期首工場在庫量」の市場倉庫(1)(2)への配分量である。第1市場倉庫と第2市場倉庫への「輸送量」の合計は「期首工場在庫量」に等しくなければならない。

輸送量の決定において注意すべきことは、まず、**市場潜在需要**への適合の問題である。流通期間として、「輸送量」の決定から「有効市場在庫量」になるまでに1期間が必要である。従って、「輸送量」の決定には次期の「市場潜在需要量」に適合させねばならない。

つぎに、需要適合のためには**生産量**、**輸送量**、**流通マージン**の意思決定が適格に行なわれ、「在庫過剰量」または「在庫不足量」を最小にせねばならない。

在庫過剰量が大きいほど、「在庫管理費」が増大する。特に、

期首市場在庫量 > 期首生産能力

の場合には、倉庫賃借料（1単位当り約2ドル）がその在庫超過分に課せられる。また、**在庫不足量**が大きいほど、在庫があれば獲得しえたであろう利潤を失う、つまり、品切れコスト（機会費用）を伴うことになる。

第2に、**流通マージン**は製造企業が小売企業との取引交渉において提示する「誘因」である。小売業者が製造企業に提示する「誘因」つまり、「販売能力」は十分であると仮定する。

「流通マージン」は、ゼロではあり得ないし、極端に高くもなりえないという理由から

$$0 \text{ ドル} \leq \text{流通マージン} \leq \left(\frac{\text{価格}}{3} \right) \text{ ドル}$$

という制約条件を設定している。**流通マージン**は**マーケティング効果**との相乗効果によって**小売交渉効果**を生成する。「マーケティング効果」は製造業者から小売業者への販売促進的な**プッシュ効果**としての意味と、さらに、消費者へ

の販売促進を通じての消費者のブランド忠誠度向上による**プル効果**としての意味とを含んでいる。

「流通マージン」の決定においては、前期の「期末市場在庫量」の状況と今期の「マーケティング効果」の予測に留意せねばならない。流通マージンによる「小売交渉効果」については「市場行動システム」をみよ。

(8) 資金運用

資金運用の意思決定は、蓄積された資金を運用し、「営業外収入」という形で経営成果を向上させようとするものである。「資金運用」は短期運用の形態に限定されており、いつでも回収可能である⁽²⁰⁾。

「資金運用」の意思決定には次のような制約条件が付けられている。

① 資金運用の可能化条件

期首短期借入金=1,000千ドル⁽²¹⁾，かつ，期首現金在高> 1,000千ドルならば，

◦ 資金運用額 ≤ (期首現金在高 - 1,000千ドル)

② 資金運用の必要化条件

期首短期借入金=1,000千ドル，かつ，期首現金在高 > 期首買入債務 × 2. であり，資金運用の決定がなされていない時には，自動的に，資金運用額が次式により設定される。

◦ 資金運用額 = (期首現金在高 - 期首買入債務 × 2)

③ 資金回収の必要化条件

期首現金在高=1,000千ドル⁽²²⁾，かつ，期首短期借入金 > 1,000千ドル，かつ，期首資金運用残高 > 0 ならば，

◦ 資金回収額 (負の資金運用額)

= -min (期首資金運用残高，期首短期借入金 - 1,000千ドル)

(20) 資金運用の方法を「長期」，「短期」などに分ければ，ポート・フォリオ・セレクションの意思決定を組み込みうる。

(21) 「期首短期借入金」の最低水準を1,000千ドルに設定している。

(22) 「期首現金在高」の最低水準を1,000千ドルに設定している。

④ 利子収入と資金回収費用

以上のような条件のもとで行なわれる資金運用には、次のような利子収入と資金回収費用を伴う。

- 利子収入（営業外収益）

$$= (\text{期首} \cdot \text{期末資金運用残高} / 2) \times \text{金利係数}$$

- 資金回収費用（営業外費用）

$$= | \text{資金回収額} \times (0.01 + \text{金利係数}) |$$

資金回収費用を考慮すると、少なくとも、2期間以上の運用が可能でなければ、利子収入を得ることができない。

[4] 市場行動システム

「市場行場システム」は市場組織内における「水平的（競争的）相互作用過程」と「垂直的（流通上の）相互作用過程」との交錯において認識される⁽²³⁾。この市場組織内相互作用過程としての「市場行動システム」はいくつかの下位システムから構成されている。個々の下位システムは各企業の「意思決定システム」のアウト・プット、つまり、「意思決定値」、および、審判団が与える「外生的市場要因の値」とゲーム・モデル内で設定される「内生的市場要因の値」の相互関係を通じてそのアウトプットを生成する。ある下位システムのアウトプットは次の下位システムのインプットとなる。ここにシステム連鎖関係が認識される。従って、「市場行動システム」は複数の下位システム間の相互関連体系として認識される。

市場行動システムは次の6つの下位システムから成る。（いずれも、第1市場と第2市場とに共通である）。

(1) 企業有効販売促進と企業有効研究開発

企業のマーケティング行動は他社への相対的優位性の向上を意図する行動で

(23) このシステムは、主として、「競争的相互作用過程」を取り扱っており、「流通上の相互作用過程」における流通業者の行動および購買者の行動については、システム内部で自動的に処理している。

あり、マーケティング意思決定値を他社よりも拡大させ販売量を増加させることを意図している。しかし特色ある製品差別化のパターンも競争過程を通じて産業組織内に普及し、独創的なマーケティング戦略も競争企業間での相互学習過程を通じて均等化する傾向をもつ。

そこで、「販売促進」・「研究開発」の意思決定値は時系列の「産業加重平均」の2倍以上にならないものと仮定する。

また、販売促進支出のマーケティング効果は、産業平均を越えると通減し、意思決定値の60%のみが有効となる。なぜなら、産業平均以上に強度の「販売促進活動」に対しては、需要家は「説得的」と感じ製品に対する選好を低下させるからである。

そこで、有効販売促進支出、有効研究開発支出は産業加重平均支出を基準として次のようにして算出される。

産業加重平均 i を(産業加重平均 $_{i-1}$ + 産業単純平均 i) / 2として

産業加重平均 i \geq 企業販売促進費 $_{ii}$ ならば、

- 有効企業販売促進費 $_{ii}$ = 企業販売促進費 $_{ii}$

産業加重平均 i $<$ 企業販売促進費 $_{ii}$ $<$ 産業加重平均 i \times 2 ならば、

- 有効企業販売促進費 $_{ii}$ = 産業加重平均 i + (企業販売促進費 $_{ii}$ - 産業加重平均 i) \times 販売促進係数

産業加重平均 i \times 2 \leq 企業販売促進費ならば、

- 有効企業販売促進費 $_{ii}$ = 産業加重平均 i \times 2

以上の有効企業販売促進費の算出方法は、全国広告、地方広告、販売部隊、研究開発に適用される。ただし、研究開発の場合には販売促進係数を掛けない。

(2) マーケティング効果

マーケティング効果の算定モデルは、企業間の競争過程の中心を表わすものである。このモデルは、Kotlerによって示されたマーケット・シェア・モデルを基礎としており⁽²⁴⁾、マーケティング意思決定値(有効値)の産業組織内での相対的優位性により「マーケティング効果」を算出する。

マーケティング効果の算出モデル⁽²⁵⁾は次式のとおりである。

$$\begin{aligned} \text{。実質マーケティング効果}_i &= \text{販売促進繰越し効果}^{\delta} \\ &\quad \times \text{有効全国広告支出}^{\alpha} \times \text{有効地方広告支出}^{\beta} \\ &\quad \times \text{有効販売部隊支出}^{\gamma} \times \text{有効研究開発効果}^{\zeta} \\ &\quad \times \text{需要家の銘柄忠誠度}^{\eta} \\ &\quad / (\text{価格} + \text{マーケティング反応係数}) \varepsilon \end{aligned}$$

。相対的マーケティング効果_i = 実質マーケティング効果_i

$$/ \sum_{i=1}^N \text{実質マーケティング効果}_i$$

この相対的マーケティング効果に基いて、市場有効需要量、企業潜在需要量が算定される。なお、「販売促進繰越し効果」については、[3]・(4)を参照せよ。「有効研究開発効果」については、[3]・(5)をみよ。

(3) 産業需要創造効果と市場有効需要量

産業全体のマーケティング活動が過去の傾向よりも積極的〔または、消極的〕となると、「市場有効需要量」は、審判団が設定した「市場潜在需要量」よりも多くなる〔または、少くなる〕。つまり、「産業需要創造効果」が作用する。

(24) Kotler, P. [3], p. 676.

Kotler のモデル。Rit = t 期における i 企業の製品に対する選好比率、Pit=価格、Ait = 広告費および販売促進費、Dit = 流通費および販売部隊費、ej=行動変数 j の弾力性、Sit = i 企業の市場シェア、ait = 広告効果指数、dit = 流通効果指数、とすると、

$$\text{Sit} = \frac{R_{it}^{e_1} \cdot P_{it}^{-e_2} \cdot (a_{it} A_{it})^{e_3} \cdot (d_{it} D_{it})^{e_4}}{\sum_i \left[R_{it}^{e_1} \cdot P_{it}^{-e_2} \cdot (a_{it} A_{it})^{e_3} \cdot (d_{it} D_{it})^{e_4} \right]}$$

(25) 「銘柄忠誠度」(EYは)「相対的マーケティング効果」(RE)の時系列加重平均値であり、次式により算定される。(N=企業数)

t = 1 のとき EYit = 1.0

t > 1 のとき EYit = (REit · N + EYi.t-1)/2.0

通常、EYは、1.0以上、1.0以下のいずれにしても、1.0に近い値を示す。

産業需要創造効果の算定には、マーケティング反応係数を除外して計算した名目マーケティング効果を用いる。すなわち、

◦名目マーケティング効果＝

$$\text{実質マーケティング効果} \times \left(\frac{\text{価格} + \text{マーケティング反応係数}}{\text{価格}} \right)^{\varepsilon}$$

◦産業総名目マーケティング効果＝ $\sum_{i=1}^N$ 名目マーケティング効果

◦産業総名目マーケティング効果加重平均_t＝

$$= (\text{同} \cdot \text{効果}_t + \text{同} \cdot \text{効果}_{t-1}) / 2$$

(ただし、 $t = 1$ のとき、同・効果加重平均_t＝同・効果_t、とする。)

◦産業需要創造効果_t＝ $\left(\frac{\text{産業総名目マーケティング効果}_t}{\text{産業総名目マーケティング効果加重平均}_t} \right)^{0.5}$

◦市場総有効需要量_t＝市場総潜在需要量_t × 産業需要創造効果_t
+ 期首産業総有効需要繰越量

産業需要創造効果_t ≧ 1、ならば、各々、

◦市場総有効需要量_t ≧ 市場総潜在需要量_t、

このようにして算出された「市場総有効需要量」は各企業の「相対的マーケティング効果」に応じて各企業に配分される。すなわち、

◦企業潜在需要量_{it}＝市場総有効需要量_t × 相対的マーケティング効果_{it}
である。

以上の算定式は両市場に適用される。

(4) 企業需要創造効果と企業有効需要量

企業の研究開発が、相対的に産業加重平均の研究開発よりも大きく、1社平均市場潜在需要量よりも大きい場合には、「企業有効需要量」は「相対的マーケティング効果」によって配分された「企業潜在需要量」よりも大きくなる。すなわち、「研究開発」の企業需要創造効果が作用する。企業需要創造効果は次式により算出される。すなわち、

◦企業需要創造効果_{it}＝(研究開発効果_{it})^η

(η＝0.225、ηは研究開発のマーケティング効果モデル係数である)。

(「研究開発効果」については「意思決定⑤」をみよ)。

企業研究開発効果 $i_i \geq 1$, であるので,

- 企業需要創造効果 $i_i \geq 1$ 。

そこで、「企業有効需要量」は次式より算出される。ただし、需要不足の時
には産業超過有効需要量が再配分されうる。

- 企業有効需要量 $i_i = \text{企業需要創造効果 } i_i \times \text{企業潜在需要量 } i_i$
+ 期首企業有効需要繰越量 i_i

(5) 小売交渉効果と有効市場在庫量

「意思決定」⑦「流通」で述べたように、**流通マージン**は**相対的マーケティング効果**との**相乗効果**によって**小売交渉効果**を生成する。

- 相対的流通マージン効果 $i_i = \frac{\text{流通マージン } i_i}{\sum_{i=1}^N \text{流通マージン } i_i / N}$
- 相対的マーケティング効果指数 $i_i = \text{相対的マーケティング効果 } i_i \times N$
- 小売交渉効果 $i_i = \min(\text{相対的流通マージン効果 } i_i$
 $\times \text{相対的マーケティング効果指数 } i_i, 1)$

この**小売交渉効果**は**流通マージン**の小売業者に対する相対的な「誘因効果」
および、マーケティング諸決定の**プッシュ効果**と**プル効果**を包含する効果である。

また、**小売交渉効果**は小売業者の製品取扱い量の大きさを規定する。その規定の仕方は次の二様である。

小売交渉効果 $i_i = 1$, ならば

小売業者は i 企業の「期首市場在庫量 i_i 」の全量の取扱いを引き受ける。

すなわち,

- 有効市場在庫量 $i_i = \text{期首市場在庫量 } i_i$

この場合「小売交渉効果」が1より大きくても同じ結果となる。従って企業は「小売交渉効果」がちょうど1になるように「流通マージン」を決定することが望ましい。

小売交渉効果 $\alpha_i < 1$, ならば, 小売業者は i 企業の「期首市場在庫量 i_t 」に「小売交渉効果 α_i 」をかけた量だけの取扱いを行なう。すなわち,

- 有効市場在庫量 $i_t =$ 期首市場在庫量 $i_t \times$ 小売交渉効果 α_i

この場合は小売業者の製造企業に対する同調性が低いことを示す。

(6) 需給適合

① 超過需要量

需給適合は, 購買者の企業の製品に対する需要量 (企業有効需要量) と企業の市場における品揃え量 (有効市場在庫量) との適合を意味する。需要が品揃えより多い場合, 超過需要が発生する。

- 超過需要量 $i_t = \max [(\text{企業有効需要量 } i_t^* - \text{有効市場在庫量 } i_t), 0]$

超過需要量は, 1 回だけ有効需要不足企業に再配分される。これによって, 企業有効需要量の修正が行われる。

② 超過需要量の再配分 (購買者の購買銘柄変更, および, 購買時期繰延べ)

企業の有効市場在庫量が企業有効需要量よりも小さく, 在庫不足が生じると, 消費者は購買銘柄の変更か購買時期の繰延べを行なう。

① 超過需要の他社への流出 (購買者の購買銘柄変更)

- 産業超過需要量 $= \sum_{i=1}^N [\max (\text{企業超過需要量 } i_t \times 0.6, 0)]$

- 企業有効需要量 $i_t =$ 企業有効需要量 $i_t +$ 産業超過需要量 \times 相対的マーケティング効果 i_t

なお, 「超過需要への他社への流出」の調整は, 1 回だけ行なわれ, 1 回の調整後に発生した超過需要は自然に消滅するものとする。

② 超過需要の次期への繰越し (購買者の購買時期繰越)

- 期末企業有効需要繰越量 $i_t = \max (\text{企業超過需要量 } i_t \times 0.2, 0)$

- 期末産業有効需要繰越量 $= \sum_{i=1}^N [\max (\text{企業超過需要量 } i_t \times 0.2, 0)]$

③ 企業実際販売量

「企業実際販売量」は, 需給適合の結果である。従って, 需要要因としての

「企業有効需要量」と供給要因としての「有効市場在庫量」とを比較し、いずれか小さい方が「企業実際販売量」となる。この関係は次のように表わされる。

$$\circ \text{企業実際販売量 } i = \min(\text{企業有効需要量 } i, \text{有効市場在庫量 } i)$$

[5] 経営成果システム

経営成果システムは、各企業が企業行動・市場行動を通じて達成する経営成果諸要因の相互関連を示すものである。経営成果システムは次の6つのサブ・システムから成る。

すなわち、「期末経営構造要因」、「期末財務構造要因（損益計算書と貸借対照表の基礎要因）」、「損益計算書」、「貸借対照表」、「経営分析」、および、「経営総合評価」である。

(1) 期末経営構造要因

期末諸要因は次期の期首諸要因へ繰越されるものであり、次の諸要因から成る。

期末生産能力、期末工場在庫量、期末市場在庫量、期末累積販売量、期末企業有効需要繰越量、期末全国広告費繰越額、期末販売部隊繰越額、期末銘柄忠誠度。

各々の期末諸要因は次のように算出される。

- 期末生産能力 = 期首生産能力 + 当期生産能力追加量
- 期末工場在庫量 = 当期生産量
- 期末市場在庫量 j = 期首市場在庫量 j - 当期販売量 j + 当期輸送量 j
- 期末在庫総量 = 期末工場在庫量 + $\sum_{j=1}^2$ 期末市場在庫量 j
- 期末累積販売量 = 期首累積販売量 + $\sum_{j=1}^2$ 当期販売量 j
- 期末企業有効需要繰越量 j = $\max(\text{企業超過需要量 } j \times 0.2, 0)$

◦ 次期首全国広告費繰越額

次期首繰越額（1期前）＝当期有効全国広告費

次期首繰越額（ $k + 1$ 期前）＝当期首繰越額（ k 期前）（ $k = 1, 2, 3$ ）

◦ 次期首販売部隊繰越額＝当期有効販売部隊費

◦ 期末銘柄忠誠度＝（期首銘柄忠誠度＋当期相対的マーケティング効果
×企業数⁽²⁶⁾）／2

以上の期末諸要因は次期期首諸要因に繰越される。次項の貸借対照表の期末諸要因も次期期首に繰越される。

（2）期末財務構造要因

このマーケティング・ゲームは単純化されたシステムであるので、現実の損益計算書、貸借対照表より単純化されている。

損益計算書と貸借対照表の基礎となる「期末財務構造要因」は次のようにして算出される⁽²⁷⁾。

- 総売上＝〔価格×販売総量〕×1000⁽²⁸⁾
- 流通取引費＝〔流通マージン×販売総量〕×1000
- 減価償却費＝期首固定投資残高／10
- 一般管理費＝減価償却費×0.10
- 販売固定費＝減価償却費×0.10
- 生産固定費＝減価償却費×0.80
- 単位生産費＝生産固定費／〔生産量×1000〕＋単位変動費
- 生産費＝〔生産量×単位生産費〕×1000
- 期末棚卸資産＝期末在庫総量×単位生産費×1000
- 売上原価＝生産費＋期首棚卸資産－期末棚卸資産
- 市場調査費＝調査項目数×6000

(26) 「相対的マーケティング効果」に「企業数」をかけているのは、対平均値比重にするためである。

(27) マーケティング・ゲームの特殊性により、各要因の算出方法には現実的でない点も含まれている。

(28) 同式で「1000倍」しているのは、意思決定変数、および、関連変数が1000単位であるからである。

- 販売促進費 = $\left[\text{全国広告費} + \sum_{j=1}^2 \text{地方広告費}_j + \sum_{j=1}^2 \text{販売部隊費}_j \right] \times 1000$
- 輸送費 = $\sum_{j=1}^2 \text{輸送量}_j \times 0.5 \times 1000$
- 在庫管理費 = $\left[\sum_{j=1}^2 \text{期首市場在庫量}_j \times 0.5 + \right.$

$$\max \left(\sum_{j=1}^2 \text{市場在庫量}_j - \text{生産能力}, 0 \right)$$

$$\left. \times \text{超過市場在庫管理費単価} \right] \times 1000$$
- 超過市場在庫管理費単価 = $2 + \text{景気指数} / 10$
- 販売管理費 = 販売促進費 + 市場調査費 + 販売固定費
- 販売費・一般管理費 = 販売管理費 + 研究開発費 + 輸送費 + 在庫管理費
+ 流通取引費
- 総費用 = 売上原価 + 販売費・一般管理費
- 資金回収額⁽²⁹⁾ = $|\min(\text{資金運用}, 0)|$
- 資金回収費 = $[\text{資金回収額} \times \text{資金回収率}] \times 1000$
- 固定負債利子 = 期首固定負債残高 \times 固定負債利率
- 短期借入利子⁽³⁰⁾ = 期首短期借入金 \times 短期借入利率
- 営業外費用 = 資金回収費 + 固定負債利子 + 短期借入利子
- 営業外収益 = 期首資金運用残高 \times 資金運用利率
- 営業利益 = 総利益 - 総費用
- 期末売上債権 = 総売上 \times 買掛・売掛率
- 期末買入債務* = $(\text{生産費} + \text{販売費} \cdot \text{一般管理費}) \times \text{買掛} \cdot \text{売掛率}$

(29) 「資金運用」の意思決定値が「負」の時には「資金回収」を意味する。この場合には、証券売却手数料などの「資金回収費」を伴う。

(30) 現金支払の順序

- ① 期首：利子支払
- ② 期首：借入金返済
- ③ 期中：経常支出

- 売上債権受取額 = 期首売上債権 + (総売上 - 期末売上債権)
- 買入債務支払額

$$= \text{期首買入債務} + \text{期末買入債務} \times (1 - \text{買掛} \cdot \text{売掛率}) / \text{買掛} \cdot \text{売掛率}$$
- 固定資本投資額

$$= [(\text{生産能力追加量} \times \text{固定資本単価}) + (\text{研究開発} / 2)] \times 1000$$
- 期末固定資産 = 期首固定資産 + 固定資本投資額
- 固定負債返済額 = $\min(\text{期首固定負債} / 5, 300 \text{千ドル})$
- 期末固定負債 = 期首固定負債 - 固定負債返済額

$$+ \text{固定資本投資額}$$
- 期末現金在高* = 期首現金在高 - 資金運用額 + 営業外収益

$$- \text{営業外費用} + \text{売上債権受取額} - \text{買入債務支払額}$$

$$- \text{法人税引当金} - \text{固定負債返済額}$$
- 短期借入金返済額

$$= \min(\text{期首短期借入金} - 1,000 \text{千ドル}, \text{期末現金在高現} 1,000 \text{千ドル})$$
- 期末短期借入金 = 期首短期借入金 - 短期借入金返済額

(負の「短期借入金返済額」は「借入増加額」を意味する)。
- 期末現金在高 = 期末現金在高* - 短期借入金返済額
- 期末流動資産 = 期末現金在高 + 期末売上債権 + 期末資金運用残高
- 期末流動負債 = 期末短期借入金 + 期末買入債務
- 期末棚卸資産 = 期末在庫総量 \times 単位生産費
- 期末当座資産 = 期末流動資産 + 期末棚卸資産
- 総資産 = 期末当座資産 + 期末固定資産
- 負債・資本計 = 総資産
- 負債計 = 期末固定負債 + 期末流動負債
- 引当金 = 5,000千ドル
- 資本金 = 6,000千ドル
- 資本計 = 負債・資本計 - 負債計 - 引当金
- 当期純利益 = 営業利益 + 営業外収益 - 営業外費用

- ・法人税引当金 = $\max(\text{当期純利益} \times \text{法人税率}, 0)$
- ・税引後純利益 = 当期純利益 - 法人税引当金
- ・剰余金 = 資本計 - 資本金 - 税引後純利益

以上の基礎要因を用いて、次に示すような「損益計算書」と「貸借対照表」が作成される。

(3) 損益計算書

<u>損益計算書</u>	
総売上	
売上原価	
販売費・一般管理費	
総費用	
営業利益	
営業外収益	
営業外費用	
当期純利益	
法人税引当金	
税引後純利益	

(4) 貸借対照表

<u>貸借対照表</u>	
総資産	負債・資本合計
流動資産	負債計
当座資産	流動負債
現金在高	買入債務
売上債権	短期借入金
資金運用残高	固定負債
棚卸資産	引当金
固定資産	資本計
	資本金
	税引後純利益
	剰余金

(5) 経営分析

「期首・期末諸要因」, 「損益計算書」, 「貸借対照表」, および, 「市場成果諸要因」等のデータを用いて様々な経営分析が行なわれ, 経営計画の確立のための基礎資料が提供される。ここでの経営分析は主として3つの部分から構成されている。「マーケティング管理分析」, 「在庫管理分析」, 「財務管理分析」である。

① マーケティング管理分析

「マーケティング管理分析」は, 主として, マーケティングに関する「意思決定値」と「産業加重平均値」との比較, および「マーケティング効果」等の産業平均との比較による情報を提供する。この情報は, 需要動向・経営計画に

対するマーケティング意思決定の適合性を判断するのに用いる。(研究開発を含む)。

「マーケティング管理分析」により提供される情報は次のとおりである。

意思決定の対産業平均比較

$$\bullet \text{ 意思決定/産業平均比率} = \frac{\text{有効意思決定値}}{\text{期首産業加重平均値}} \times 100$$

「有効意思決定値」は、前述のように、「期首産業加重平均値」を越える値については60%だけ「有効」とし、また、その2倍までを「有効」として算出した値である。(研究開発では100%「有効」)。

「期首産業加重平均値」は前期の「期末産業加重平均値」、つまり、

$$\bullet \text{ 前期末産業加重平均値} = (\text{前期産業加重平均値} + \text{前期産業単純平均値}) / 2$$

に等しい。

「意思決定/産業平均比率」は、「価格」、「全国広告」、「地方広告」、「販売部隊」、「流通マージン」、「研究開発」について算出される。

需要増加が予測される場合、市場在庫が過大である場合には、「意思決定/産業平均比率」が100%よりかなり多くなるように(価格の場合は100%以下になるように)意思決定方針を立てることが望ましい。

販売量成長率の対産業平均比較

$$\bullet \text{ 産業販売量} = \sum_{i=1}^N \text{ 企業販売量 } i$$

$$\bullet \text{ 企業販売量平均増加量 } \tau_{ij} = \sum_{i=2}^T (\text{企業販売量}_{ij} - \text{企業販売量}_{i-1, ij}) / (T-1)$$

$$\bullet \text{ 産業販売量平均増加量 } \tau_j = \sum_{i=1}^N \text{ 企業販売量平均増加量 } \tau_{ij}$$

$$\bullet \text{ 企業販売量平均成長率 } \tau_{ij} = \frac{\text{企業販売量平均増加量 } \tau_{ij}}{\sum_{i=2}^T \text{ 企業販売量 }_{ij}} \times 100$$

$$\bullet \text{ 産業販売量平均成長率 } \tau_j = \frac{\text{産業販売量平均増加量 } \tau_j}{\sum_{i=2}^T \text{ 産業販売量 }_{ij}} \times 100$$

$$\text{企業／産業販売量成長性相対比率 } \tau_{ij} = \frac{\text{企業販売量平均成長率 } \tau_{ij}}{\text{産業販売量平均成長率 } \tau_j} \times 100$$

市場行動効果の対産業平均比較

- ・研究開発効果：（「市場行動システム」より）
- ・銘柄忠誠度：（ 〃 ）
- ・マーケティング効果指数：（ 〃 ）
- ・小売交渉効果：（ 〃 ）

これらの指標は「産業平均効果」との比率で表わされる。従って、「1.0以上」であれば「産業平均効果」以上を示し、「1.0以下」であれば「産業平均効果」以下を示す。（「マーケティング効果指数＝相対的マーケティング効果×企業数」）。

② 在庫管理分析

「在庫管理分析」は「生産量」，「輸送量」，および，「流通マージン」に関する意思決定が，「期首市場在庫量」と「市場有効需要量」との推移に適合しているかどうかについての情報を提供する。この情報は，過剰在庫管理費，または，過小在庫（品切れ）による機会費用（在庫があれば獲得しえた収益）を最小にする意思決定値を導出するために用いる。

在庫量適合度

① 小売交渉効果を考慮しない場合

$$\text{期首市場在庫量適合度 } i = \frac{\text{期首市場在庫量 } i}{\text{当期企業有効需要量 } i} \times 100$$

この指標は生産量・輸送量の意思決定が企業の有効需要量に適合していた程度を表わす。

② 「小売交渉効果」を考慮する場合

$$\text{有効市場在庫量適合度 } i = \frac{\text{当期有効市場在庫量 } i}{\text{当期企業有効需要量 } i} \times 100$$

この指標は，「市場在庫量適合度」にさらに「小売交渉効果」を加味して，実質的な在庫過不足を比率で表わしたものである。100%を最適値とし，100%以上は「在庫過剰率」，100%以下は「在庫不足率」を表わす。（以上の比率は

市場別に算出される)。

在庫過・不足コストの対売上比率

④ 期首市場在庫量適合度 >100 の場合には「在庫過剰コスト」が発生する。

$$\bullet B = \text{在庫過剰量} = \sum_{j=1}^2 \{\max[(\text{期首市場在庫量}_j - \text{企業実際販売量}_j), 0]\}$$

$$\bullet A = [\sum_{j=1}^2 \text{期首市場在庫量}_j]$$

$$\bullet \text{在庫管理費} = A \times 0.5 + \max\{[(A - \text{期首生産能力}) \times \text{超過市場在庫管理費単価}, 0]\} \times 1000$$

$$\bullet \text{在庫過剰コスト} = \{B \times 0.5 + \max\{[(A - \text{期首生産能力}) \times \text{超過市場在庫管理費単価}, 0]\} \} \times 1000$$

$$\bullet \text{在庫過剰コスト率} = \frac{\text{在庫過剰コスト}}{\text{総売上}} \times 100$$

⑤ 有効市場在庫量適合度 <100 の場合、「在庫不足コスト(品切れ機会コスト)」が発生する。つまり、

$$\bullet \text{在庫不足量} = \sum_{j=1}^2 \{\max(\text{企業有効需要量}_j - \text{有効市場在庫量}_j, 0)\}$$

$$\bullet \text{在庫不足コスト} = \text{在庫不足量} \times \text{価格} \times 0.08 \times 1000$$

(売上純利益率を8%とした。)

$$\bullet \text{在庫不足コスト} = \frac{\text{在庫不足コスト}}{\text{総売上}} \times 100$$

③ 財務管理分析

「財務管理分析」は、「経営方針」,「市場調査」をはじめとする「意思決定」全体が「企業構造」,「市場構造」の動向に合理的に適応していたかどうかを包括的に示す情報を提供する。各チームの意思決定者はこれらの情報をもとに将来の経営成果を最適化しうるように次期の意思決定を行なう。

「財務管理分析」により提供される情報は次のとおりである。この「財務管理分析」,において利用されるデータは、「損益計算書」,「貸借対照表」に

おける各要因のデータである。

収益性比率

- 総資本純利益率（税引後） = $\frac{\text{税引後純利益} \times 8}{\text{期首} \cdot \text{期末総資産計}} \times 100$
- 自己資本純利益率（〃） = $\frac{\text{税引後純利益} \times 8}{\text{期首} \cdot \text{期末自己資本}^{(31)}} \times 100$
- 売上高純利益率（〃） = $\frac{\text{税引後純利益}}{\text{総売上}} \times 100$
- 売上高営業利益率 = $\frac{\text{営業利益}}{\text{総売上}} \times 100$

回転率

- 総資本回転率 = $\frac{\text{総売上} \times 8}{\text{期首} \cdot \text{期末総資本計}} \times 100$
- 自己資本回転率 = $\frac{\text{総売上} \times 8}{\text{期首} \cdot \text{期末自己資本}} \times 100$
- 棚卸資産回転率 = $\frac{\text{総売上} \times 8}{\text{期首} \cdot \text{期末棚卸資産計}} \times 100$
- 固定資産回転率 = $\frac{\text{総売上} \times 8}{\text{期首} \cdot \text{期末固定資産計}} \times 100$
- 売上債権回転率 = $\frac{\text{総売上} \times 8}{\text{期首} \cdot \text{期末売上債権計}} \times 100$
- 買入債務回転率 = $\frac{\text{総売上} \times 8}{\text{期首} \cdot \text{期末買入債務計}} \times 100$

安全性比率

- 流動比率 = $\frac{\text{流動資産}}{\text{流動負債}} \times 100$
- 当座比率 = $\frac{\text{当座資産}}{\text{流動負債}} \times 100$
- 固定比率 = $\frac{\text{固定資産}}{\text{自己資本}} \times 100$
- 固定長期適合比率 = $\frac{\text{固定資産}}{\text{固定負債} + \text{自己資本}} \times 100$

(31) ここでの「自己資本」は貸借対照表の「資本計」に等しい。
自己資本=資本金+税引後純利益+剰余金

$$\text{自己資本比率} = \frac{\text{自己資本}}{\text{総資産}} \times 100$$

企業間信用比率

$$\text{売上債権対買入債務比率} = \frac{\text{売掛金}}{\text{買掛金}} \times 100$$

費用負担比率

$$\text{売上原価率} = \frac{\text{売上原価}}{\text{総売上}} \times 100$$

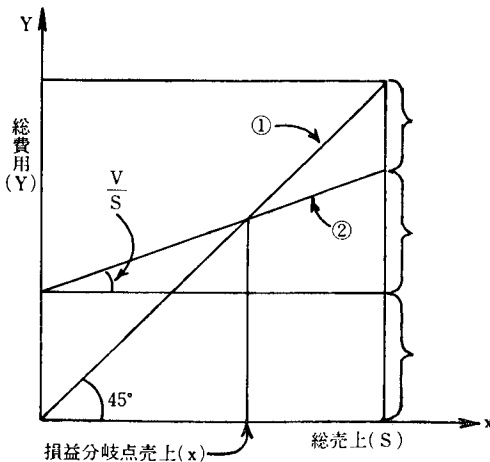
$$\text{利子負担率} = \frac{\text{短期借入利子} + \text{長期借入利子}}{\text{総売上}} \times 100$$

$$\text{減価償却費率} = \frac{\text{総固定費}}{\text{総売上}} \times 100$$

損益分岐点比率

$$\text{損益分岐点売上}^{(32)} = \frac{\text{総固定費}}{1 - \frac{\text{総変動費}}{\text{総売上}}}$$

(32) 損益分岐点売上は、周知のように、次のようにして求められる。①式は、「総売上 = 「総費用」を表わす。S = 総売上額、F = 総固定費、V = 総変動費



$$\textcircled{1} : Y = X$$

$$\textcircled{2} : Y = F + \frac{V}{S} X$$

①を②へ代入すると、

$$X = F + \frac{V}{S} X$$

$$F = X - \frac{V}{S} X$$

$$F = X \left(1 - \frac{V}{S}\right)$$

$$X = \frac{F}{\left(1 - \frac{V}{S}\right)}$$

$$\circ \text{損益分岐点比率} = \frac{\text{損益分岐点売上}}{\text{総売上}} \times 100$$

◦ 損益分岐点比率 ≥ 100 ならば、各々、営業利益 ≥ 0

従って、企業は損益分岐点比率が 100 以上になるように諸々の意思決定を行なわねばならない。

収益率時系列安定性比率 (第 5 期から算定する)

$$\circ \text{収益率平均値 } i_T = \sum_{i=1}^T [\text{収益率 } i_t] \cdot \frac{1}{T} \quad (T \geq 5)$$

$$\circ \text{収益率標準偏差 } i_T = \sqrt{\sum_{i=1}^T [\text{収益率 } i_t^2] \cdot \frac{1}{T} - \text{収益率平均値 } i_t}$$

$$\circ \text{収益率時系列安定性比率 } i_T = 100 - \frac{\text{収益率標準偏差 } i_t}{\text{収益率平均値 } i_t} \times 100$$

ただし、平均値 ≤ 0 ならば、安定性比率 = 0。

「収益率」の指標は「収益性比率」において示したとおりである。「収益率安定性」の分析には「自己資本純利益率」が適当であろう。

(6) 経営総合評価

各チームの総合評価は、第 1 期から到達期 (当期) までの長期的な **累積販売量シェア** と **累積利潤率シェア** との合成値によって行なわれる。すなわち、

$$\circ \text{累積販売量 } i_T = \sum_{i=1}^T \left[\sum_{j=1}^2 \text{販売量 } j_{it} \right]$$

$$\circ \text{累積販売量シェア } i_T (S_{iT}) = \frac{\text{累積販売量 } i_T}{\sum_{i=1}^N \text{累積販売量 } i_T} \times 100$$

$$\circ \text{累積利潤率}^* i_T = \sum_{i=1}^T [\text{総資産純利益率(税引後)} i_t]$$

◦ 累積利潤率 $i_T = \max(\text{累積利潤率}^* i_T, 0)$ (負の累積利潤率はゼロとみなす)

$$\bullet \text{ 累積利潤率シェア }_{i,T} (R_{i,T}) = \frac{\text{累積利潤率 }_{i,T}}{\sum_{i=1}^N [\text{累積利潤率 }_{i,T}]} \times 100$$

$$\bullet \text{ 総合評価シェア }_{i,T} (G_{i,T}) = \frac{\sqrt{S_{i,T}^2 + R_{i,T}^2}}{\sum_{i=1}^N \sqrt{S_{i,T}^2 + R_{i,T}^2}} \times 100$$

• 総合評価順位 $_{i,T}$ は「総合評価シェア $_{i,T}$ 」の順位である。

算出した総合評価シェア $(G_{i,T})$ は、累積販売量シェア $(S_{i,T})$ という市場目的達成度（需要充足度），および、累積利潤率シェア $(R_{i,T})$ という企業目的達成度との合成指標である。

2つのシェアを二乗しているのは、どちらか大きい方をより高く評価するためである。経営方針を明確に設定し長期経営計画を立て着実に実行しているチームには、このモデルが有利に作用する。「総合評価順位」は、「総合評価シェア」 $(G_{i,T})$ の大きさの順位により示される。

〔6〕 市場成果システム

市場成果システムは、各企業の市場行動過程を通じて達成される市場成果諸要因の相互関連を示すものである。市場成果システムは次の2つのサブ・システムから成る。

(1) 期末市場構造要因

① 期末市場有効需要量繰越量

「企業有効需要量」が「有効市場在庫量」よりも多かった場合には「品切れ」が発生する。この「品切れ量」の20%は「期末市場有効需要繰越量」となり、次期へ繰越される。（「市場システム」参照）

• 期末市場有効需要繰越量 $_{i,T}$ =

$$\sum_{i=1}^N [\max(\text{企業有効需要量} - \text{有効市場在庫量}) \times 0.2, 0]$$

として算出される。

② 期末産業販売促進加重平均

これは、販売促進および研究開発の各チームの支出額が「戦略の均等化傾向」によって産業平均と著しい相違を示さないという考え方のもとに設定された。その算出方法は次のとおりである。（研究開発も同様）。

$$\text{期末産業販売促進加重平均} = (\text{期首産業販売促進加重平均} + \text{当期産業販売促進単純平均}) \div 2.0$$

(2) 市場構造分析

① 販売量集中度

集中度は寡占の市場構造における上位企業群（寡占核）の集团的市場占有率を示す一般的な指標である。ここでは、販売量占有率を用いて、「当期販売量集中度」と「累積販売量集中度」とを算出する。これらの集中度には上位1社上位2社合計、上位3社合計が示される。

② 販売促進費集中度

販売促進費（全国広告費、地方広告費、販売部隊費）は製品差別化行動への支出を表わしている。ここでは、他産業と比較する方法がとれないので、販売促進費（両市場合計）の集中度を計算し、販売量集中度と比較する。販売量集中度と同様に、「当期販売促進費集中度」および「累積販売促進費集中度」が算出される。

Ⅲ む す び

このマーケティング戦略教育ゲーム(SMSS)は以上のような構造をもつものである。このシステムは、本来手で計算できるとく単純化されたシステム⁽³³⁾をもとに新たにコンピューター・システムとして開発したものである。開発の意図はより多くの現実性を持たせること、および、より多くの理論的意味を包含させることであった。しかし、シミュレーションの一般的限界である、要因の特定化、質的要因の除外、市場領域の限定、行動形態の限定などを含み現実

(33) Greenlaw, P. S., et. al. [2].

性・理論性の改善は十分にはなされていない。今後、さらに可能なかぎりの改善に努力したい。

このシステムは、そのような様々な限界を含んでいるが、このシステムの利用者に対して、意思決定諸要因間の相互関連性、企業間の競争的相互作用過程、経営成果の分析と評価などについての理解を深めさせるというビジネス・ゲームの教育目的の一端をはたしているといえよう。また、このシステムはコンピューター・システム⁽³⁴⁾として作成されたものであって、コンピューターを利用した経営計画、意思決定パターン、経営分析等々の比較分析的研究の目的もはたしうる。

以下、「付録」として、マーケティング戦略教育ゲームの「コンピューター・システム」のフローチャート、意思決定表、プログラム・リスト、および実行結果を載せる。

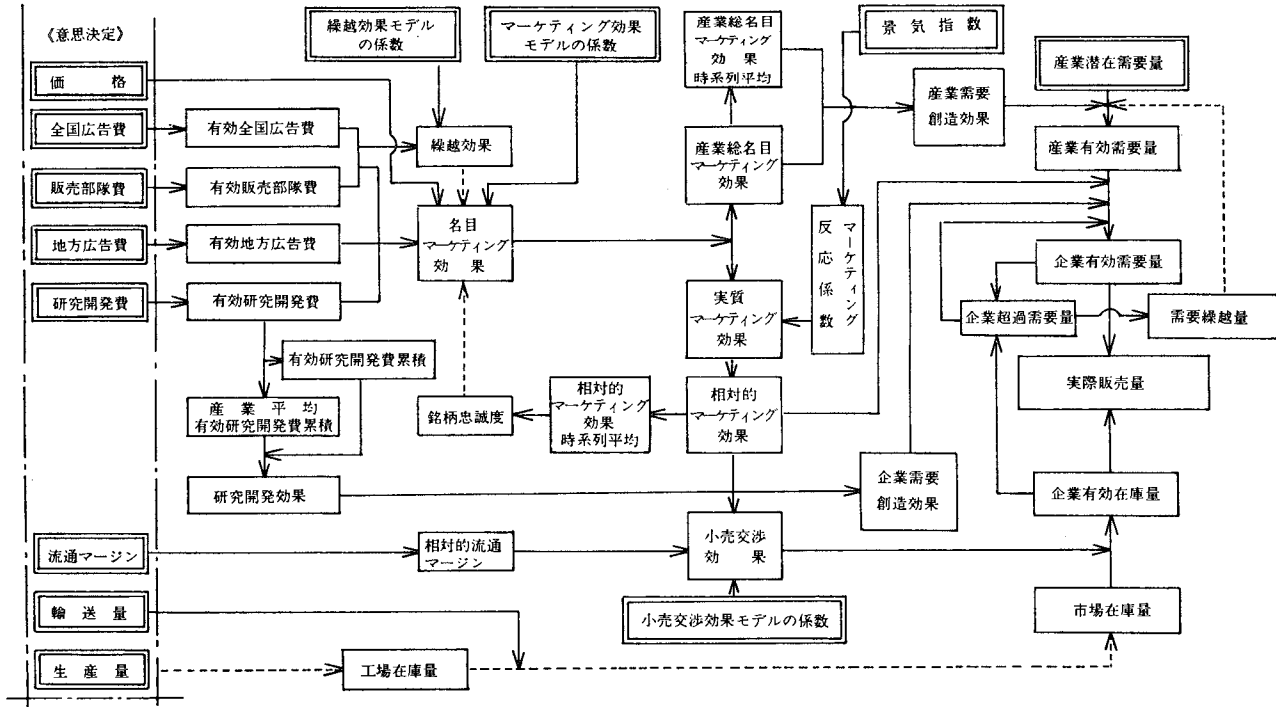
参 考 文 献

- [1] Bucklin, L. P., *A Theory of Distribution Channel Structure*, 1966。
(田村正紀訳、『流通経路構造論』, 1977)。
- [2] Greenlaw, P. S., L. W. Herron and R. H. Rawdon, *Business Simulation*, 1962。
- [3] Kotler, P., *Marketing Decision Making: A Model Building Approach*, 1971。
- [4] Mickwitz, G., *Marketing and Competition*, 1959。
- [5] 森下二次也・荒川祐吉編、『体系マーケティング・マネジメント』, 1966。
- [6] 日本経済新聞社、『経済分析のためのデータ解説』, 1972年。
- [7] 田村正紀、『マーケティング行動体系論』1971。
- [8] 通商産業省、『わが国企業の経営分析』(年鑑)。
- [9] 通商産業省、『世界の企業の経営分析』(年鑑)。

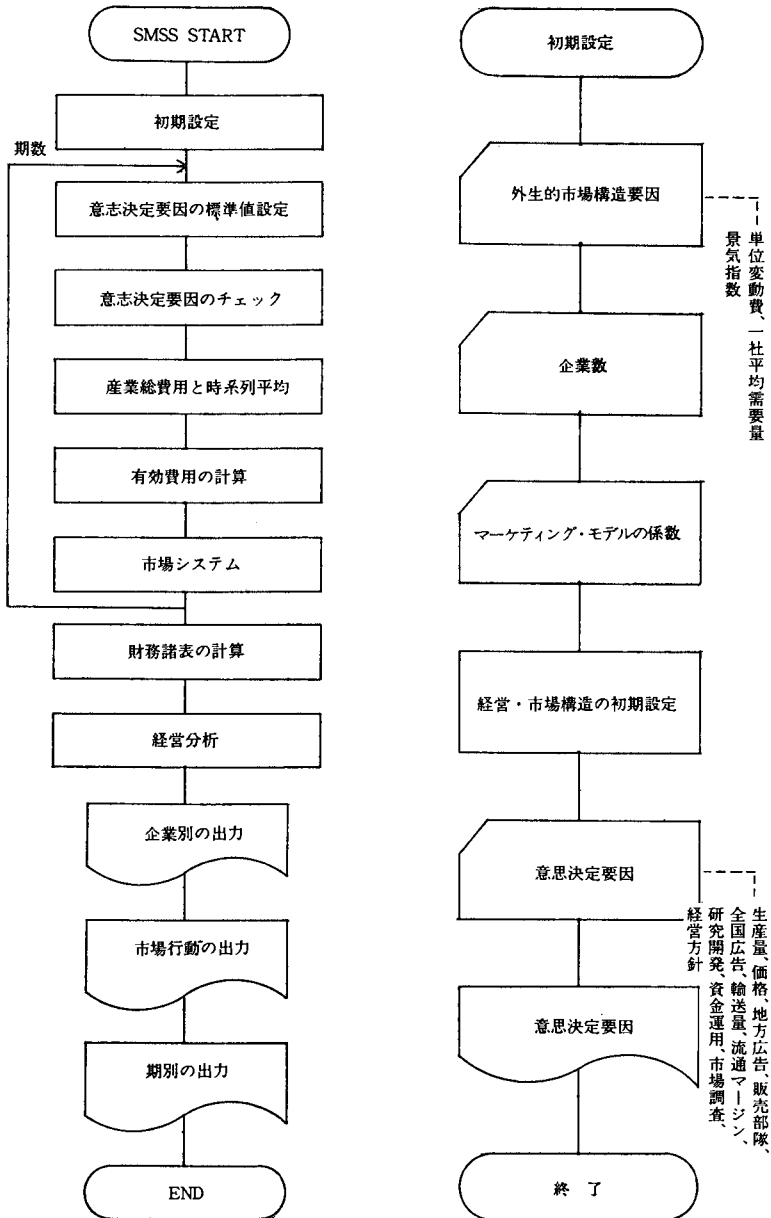
(34) この他に、意思決定要因の入力、経営成果と市場成果の出力をTSSターミナルに変換したTSS・SMSSも開発済である。

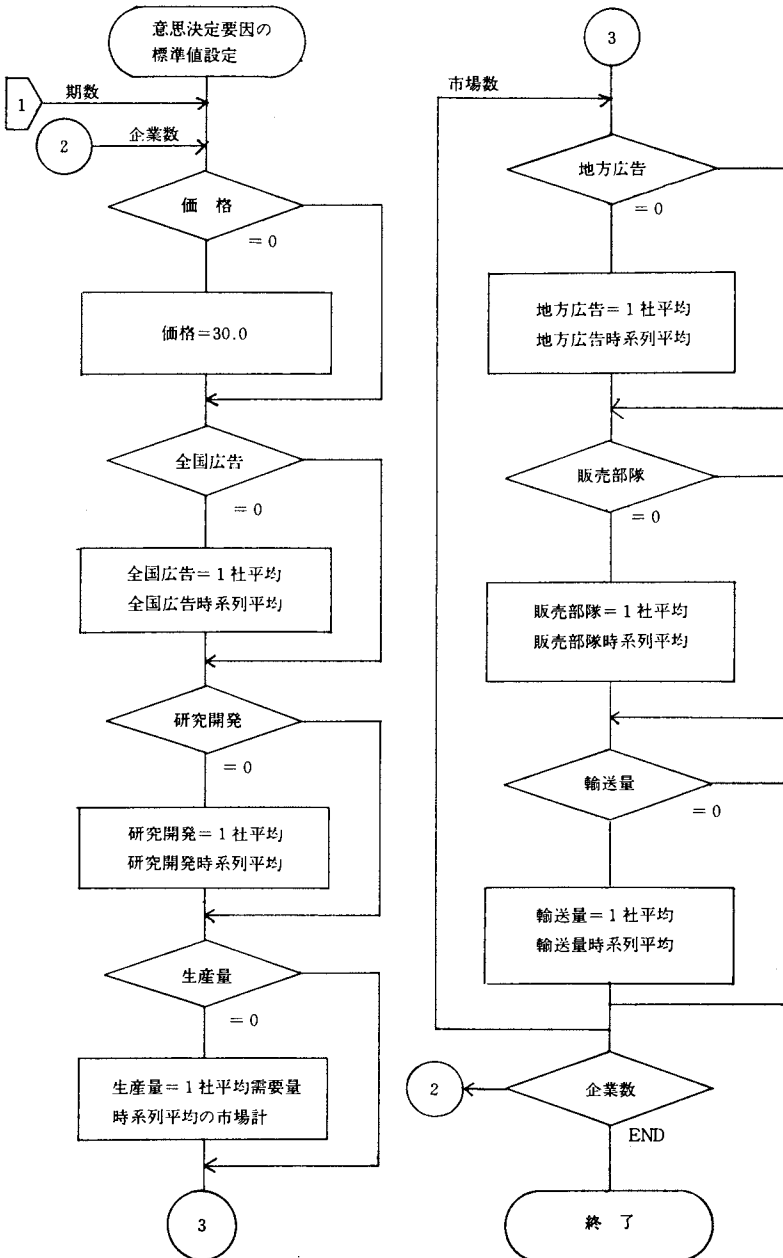
〔付録 I〕 マーケティング・戦略教育ゲームのフローチャート

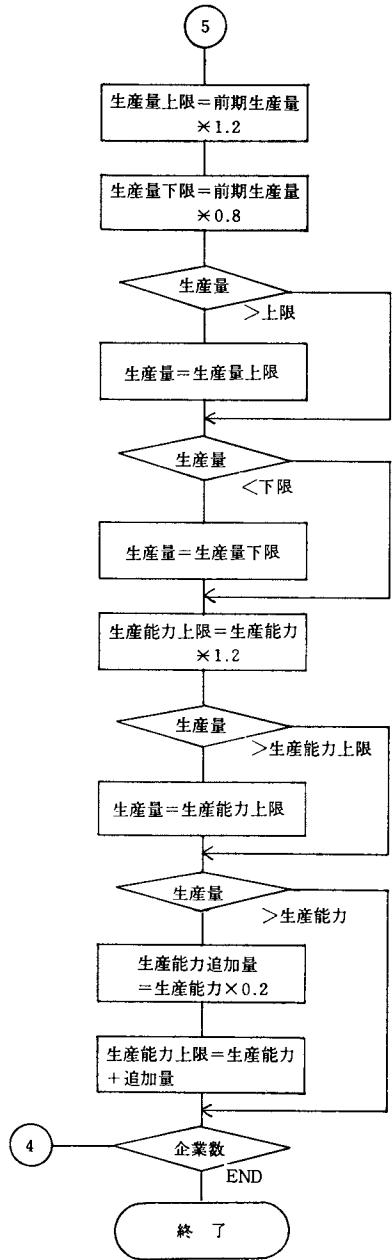
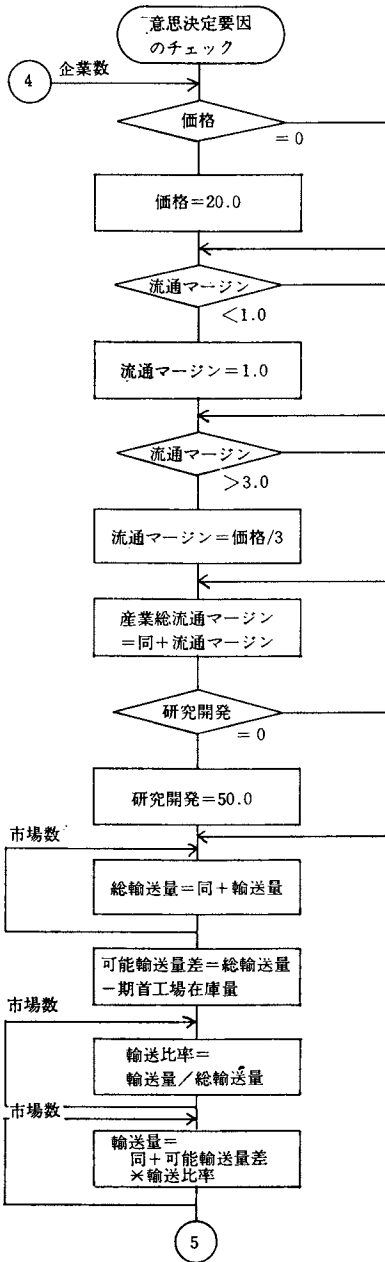
SMSSの市場行動システム

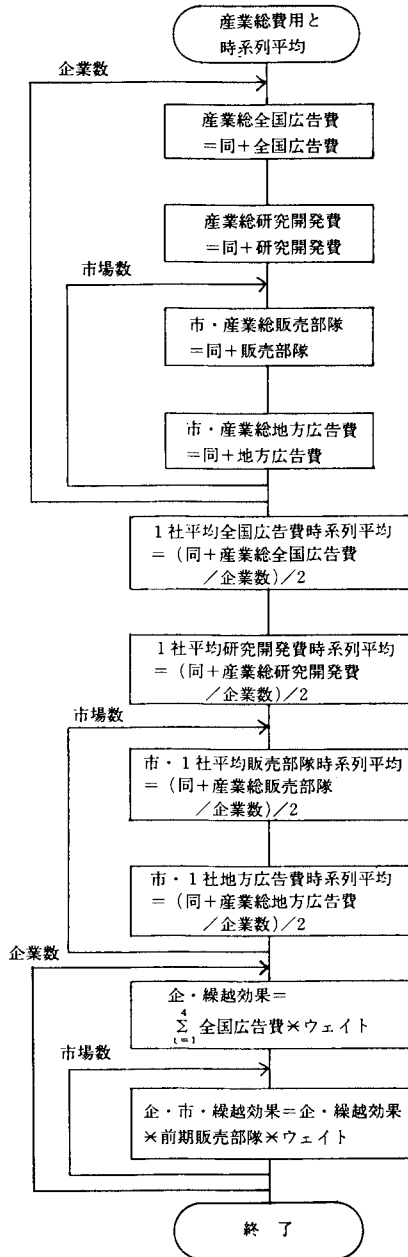


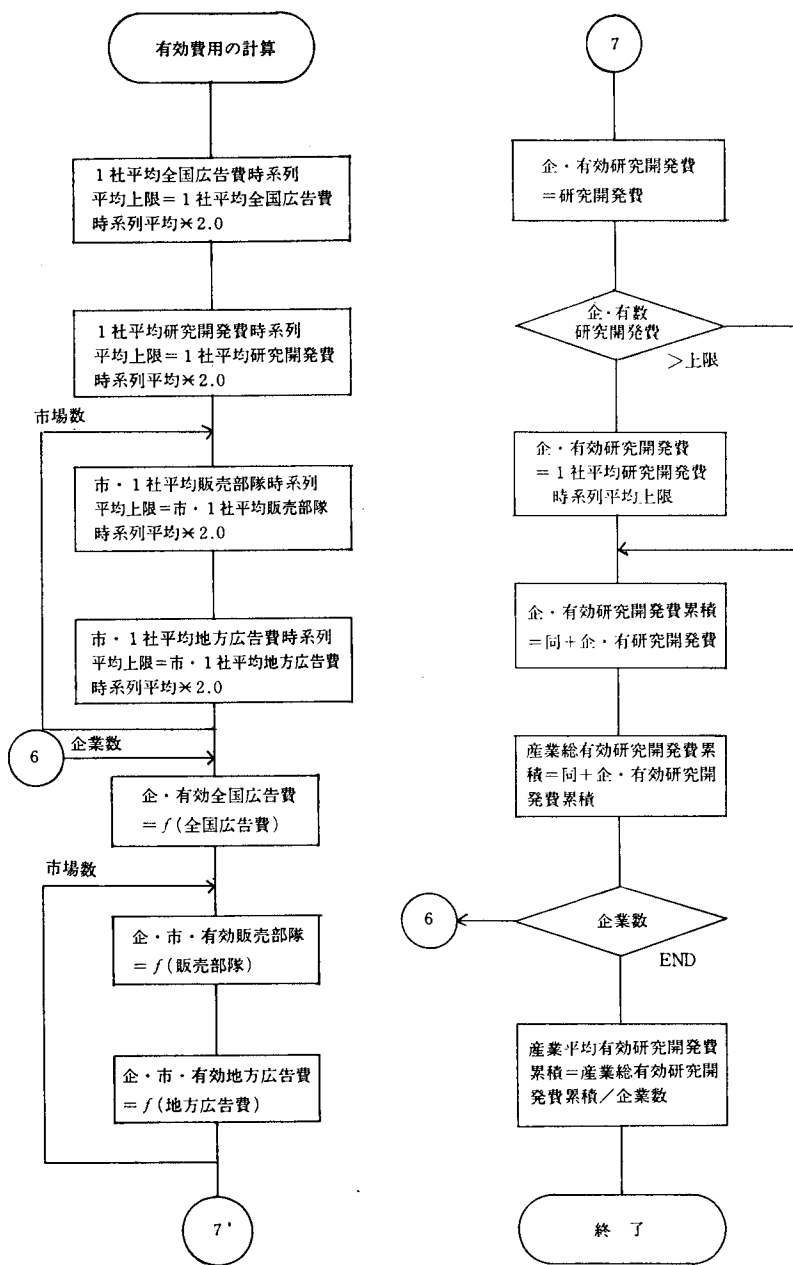
SMSSのフローチャート

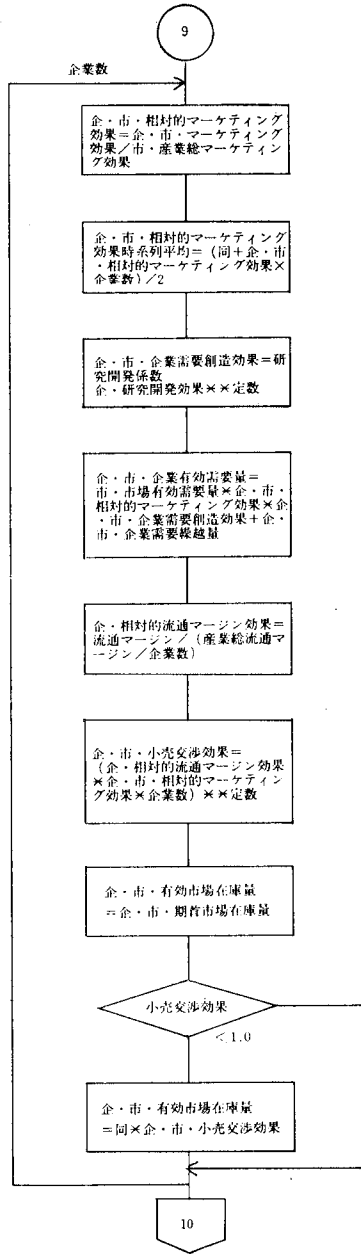
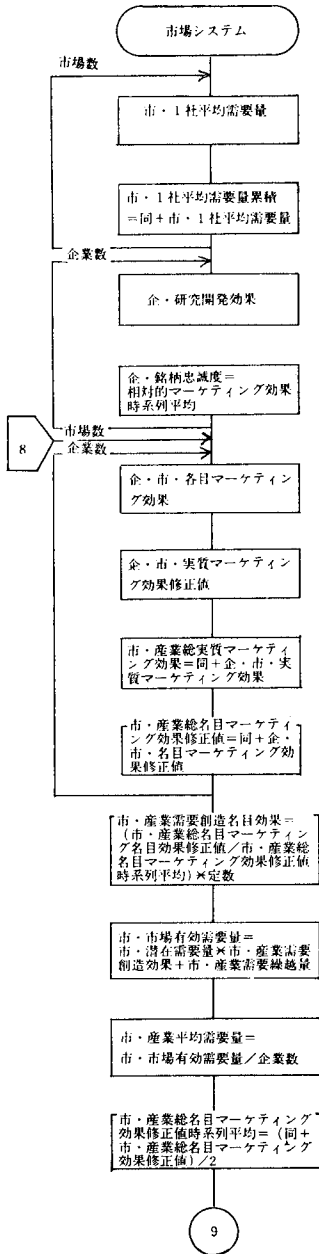


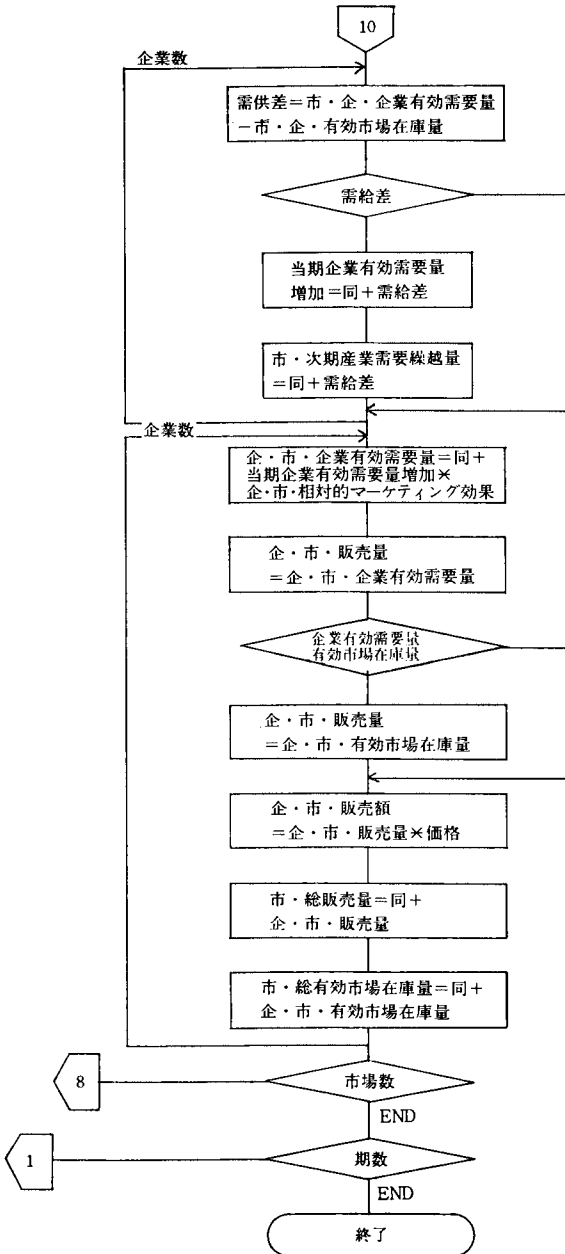












〔付録Ⅲ〕 プログラム・リスト

CARD	ISM	SOURCE STATEMENT					
1							00000010
2							00000020
3	00000001	REAL ANNA(26+10) ,AD1(26+10) ,AD2(26+10) ,					00000030
4		+ ,AD3(26+10) ,AD4(26+10) ,AVDM(26+10) ,					00000040
5		+ ,AVFS(26+2) ,AVAV(26+2) ,AVVAL(26+2) ,					00000050
6		+ ,AVLM(26+2) ,					00000060
7		+ ,AVXC(26) ,AVXP(26) ,AVRR(26) ,					00000070
8		+ ,ADT(26+8) ,ASSESH(26+8) ,ASSET(26+8) ,					00000080
9		+ ,AVXS(26+2) ,ADLOC(26+8+2) ,AVXST(26) ,					00000090
10		+ ,AVXTH(26) ,BASSET(26+8) ,BDEBT(26+8) ,					00000100
11		+ ,BVEST(26+8) ,BANALS(26+8+20) ,BFINV(26+8) ,					00000110
12		+ ,BCASH(26+8) ,BGDDOS(26+8) ,BFUND(26+8) ,					00000120
13		+ ,BTINV(26+8) ,BMINV(26+8+2) ,BTSALE(26+8) ,					00000130
14		+ ,BRCCEV(26+8) ,BPAYBL(26+8) ,BCAP(26+8) ,					00000140
15	00000002	REAL DFCE(26+8) ,BSLQ(26+8+2)					00000150
16		+ ,BFXLB(26+8) ,					00000160
17		+ ,BSALE(26+8) ,					00000170
18		+ ,BRAND(26+10) ,COSTPM(26) ,CASHS(26+8) ,					00000180
19		+ ,COST(26+8) ,CASSET(26+8) ,CLIABT(26+8) ,					00000190
20		+ ,CAPME(26+8) ,DEMAND(26+2) ,DM(26+2) ,					00000200
21		+ ,DMT(26) ,E(26+8+2) ,EPAST(26+8+2) ,					00000210
22		+ ,ELOVAL(26+8+2) ,EMINV(26+8+2) ,EMKT(26) ,					00000220
23		+ ,EADNAT(26+8) ,EADLOC(26+8+2) ,ESFRR(26+8+2) ,					00000230
24		+ ,ERAND(26+8) ,ERANDW(26+8+2) ,ERANDW(26+8+2) ,					00000240
25		+ ,ES(26+8+2) ,ECRET(26+2) ,ERT(10+2) ,					00000250
26		+ ,ETN(26+8) ,EFCRET(26+8+2) ,ERANDT(26) ,					00000260
27		+ ,EPRICE(26) ,ETM(26+8+2) ,FASSET(26+8) ,					00000270
28		+ ,FOEB(26+8) ,FCASH(26+8) ,FGDDOS(26+8) ,					00000280
29		+ ,FIXM(26+8) ,FIXC(26+8) ,FIXMC(26+8) ,					00000290
30	00000003	REAL FIPXC(26+8) ,FFXLB(26+8)					00000291
31		+ ,FDEM(26+8) ,FSLQ(26+8+2) ,					00000300
32		+ ,FFIN(26+8) ,FTINV(26+8) ,FSALE(26+8) ,					00000320
33		+ ,FTSALE(26+8) ,FMINV(26+8+2) ,FUND(26+8) ,					00000330
34		+ ,FHEBR(26+10) ,FPAYBL(26+8) ,FCAP(26+8) ,					00000340
35		+ ,FUNOC(26+8) ,FFUND(26+8) ,FVEST(26+8) ,					00000350
36		+ ,GSHARE(26+8) ,GSM(26+8) ,GVALU(26+8) ,					00000360
37		+ ,GST(26+10) ,GP(26+10) ,PRICE(26+10) ,					00000370
38		+ ,POPAST(26+10) ,POLICY(2+10) ,PPMAX(26) ,					00000380
39		+ ,PVCOST(26+10) ,					00000390
40		+ ,PLOT(26) ,PROFIT(26+10) ,R(1+1) ,					00000400
41		+ ,PROFIT(26+10) ,PROFX(26+10) ,PROF(26+10) ,					00000410
42		+ ,RASSET(26+10) ,RANDC(26+10) ,RS(26+10) ,					00000420
43		+ ,RE(26+10+2) ,RENT(26+10) ,					00000430
44		+ ,RIM(26+20) ,RAND(26+10) ,REPAST(26+10+2) ,					00000440
45	00000004	REAL RESRCH(26+8+8) ,RECOST(26+8) ,ST(26+8+2)					00000450
46		+ ,SFI(26+8+2) ,SFORCE(26+8+2) ,SFUND(26+8) ,					00000460
47		+ ,SALES(26+8) ,STCOST(26+8) ,SLQ(26+8+2) ,					00000470
48		+ ,SD(26+8+2) ,SMARE(26+8+2) ,SCOST(26+8) ,					00000480
49		+ ,SANALS(26+2+3) ,					00000490
50		+ ,ST(26+10) ,TAX(26+10) ,TCOST(26+10) ,					00000500
51		+ ,TRANC(26+10) ,TM(26+10) ,TMC(26+10) ,					00000510
52		+ ,TMGC(26+10) ,TLICAP(26+10) ,TLIABT(26+10) ,					00000520
FORTRAN (V05-00)	MAIN	SOURCE LISTING	78-01-17	13:40:47	PAGE	2	
53		+ ,TCAPT(26+10) ,					00000530
54		+ ,TRAN(26+10+2) ,TSALE(26+10) ,TMCOST(26+10) ,					00000540
55		+ ,THEBR(26+10) ,UCOST(26+10) ,VESTC(26) ,					00000550
56		+ ,XRAND(26) ,					00000560
57		+ ,XE(26+2) ,XS(26+2) ,XSD(26+2) ,					00000570
58		+ ,XRD(26) ,XPCOST(26) ,XCT(26) ,					00000580
59		+ ,XST(26) ,XSBT(26) ,XADNAT(26) ,					00000590
60	00000005	REAL XMM(26) ,XSDI(26) ,XEST(26)					00000600
61		+ ,XADLOC(26+2) ,XSFORC(26+2) ,XPRICE(26+2) ,					00000610
62		+ ,XMINV(26+2) ,XES(26+2) ,					00000620
63		+ ,Z(26+25) ,ZZ(25) ,ZZZ(25) ,					00000621
64	00000006	INTEGER CR					00000630
65	00000007	INTEGER RANK(26+10)					00000640
66	00000008	EQUIVALENCE(BASSET,FASSET),BDEBT,FOEB),BVEST,FVEST),					00000641
67		1 (BFINV,FFINV),BCASH,FCASH),BGDDOS,FGDDOS),					00000642
68		2 (BFUND,FFUND),BMINV,FTINV),BMINV,FMINV),					00000643
69		3 (BTSALE,FTSALE),BRCCEV,FRCCEV),BPAYBL,FPAYBL),					00000644
70		4 (RCAP,FCAP),BDEM,FDEM),BSLQ,FSLQ),					00000645
71		5 (BFXLB,FFXLB),BBSALE,FBSALE)					00000646
72	00000009	NAMELIST /NAME01/PRO,PTL,RQL,RANDP					00000647
73	00000010	NAMELIST /NAME02/A1,A3,A4,A5					00000660
74	00000011	NAMELIST /NAME03/ALPHA,BETA,GAMMA,DELTA,ZETA,ETA,EP					00000670
75	00000012	NAMELIST /NAME04/CASH,CRCREV,CFIX,CFUND,CGDDOS,CVEST,					00000680
76		+ CPAYBL,CDEBT,CCAPT,CRESER					00000690
77	00000013	NAMELIST /NAME05/CASHMM					00000700
78	00000014	LOGICAL W1,HEAD1,W10+0					00000710
79	00000015	DATA CR/2/,LP/0/					00000711
80	00000016	DATA L1/1/,L2/0/,LS1/1/,LE/5/					00000730
81	00000017	-0-					00000740
82	00000018	10 J=1					00000750
83	00000019	IF AD1(1)+FMD=1001) (HEAD(J),1)=1+Q0					00000760
84	00000020	11 FORMAT(10)					00000770
85	00000021	WRITE(5,11111) (HEAD(J),1),J=1,40					00000781
86	00000022	11111 FORMAT(1H,40A)					00000790
87	00000023	GO TO 10					00000800
88	00000024	1001 CONTINUE					00000810
89	00000025	IF AD1(CR,NAME01)					00000820
90	00000026	IF AD1(CR,NAME02)					00000830
91	00000027	IF AD1(CR,NAME03)					00000840
92	00000028	IF AD1(CR,NAME04)					00000850
93	00000029	IF AD1(CR,NAME05)					00000860
94	00000030	WRITE(LP,NAME01)					00000870
95	00000031	WRITE(LP,NAME02)					00000880
96	00000032	WRITE(LP,NAME03)					00000890
97	00000033	WRITE(LP,NAME04)					00000900
98	00000034	WRITE(LP,NAME05)					00000910
99							00000920
100	00000035	C					00000930
101	00000036	READ(C,35)AVXAU(1)					00000940
102	00000037	WRITE(LP,35)AVXAU(1)					00000941
103	00000038	WRITE(LP,35)BFINV(1),FTINV(1),BMINV(1),FMINV(1),2)					00000950
104	00000039	WRITE(LP,35)BFUND(1),FFUND(1),BMINV(1),FMINV(1),2)					00000960
105	00000040	WRITE(LP,35)RCAP(1),FCAP(1),BDEM(1),FDEM(1),2)					00000970
106	00000041	WRITE(LP,35)BSLQ(1),FSLQ(1),2)					00000980
107	00000042	WRITE(LP,35)SF(1),1),SF(1),1),2)					00000990
108	00000043	WRITE(LP,35)AD1(1),AD2(1),AD3(1),AD4(1),1)					00001000
109	00000044	WRITE(LP,35)AU(1),1),AD2(1),AD3(1),AD4(1),1)					00001010
110	00000045	WRITE(LP,35)AVXAU(1)					00001020
110	FORTRAN (V05-00)	MAIN	SOURCE LISTING	78-01-17	13:40:47	PAGE	3

```

111 00000466 WRITE(LP,33)AVXR(K) 00001040
112 00000467 READ(CR,35)AVXSF(1,1),AVXSF(1,2) 00001050
113 00000468 WRITE(LP,33)AVXSF(1,1),AVXSF(1,2) 00001060
114 00000469 READ(CR,33)AVXAL(1,1),AVXAL(1,2) 00001070
115 00000470 WRITE(LP,33)AVXAL(1,1),AVXAL(1,2) 00001080
116 00000471 C 00001090
117 00000472 C 33 FORMAT(12,10.1) 00001100
118 00000473 C 00001110
119 00000474 C 00001120
120 00000475 C 1011 CONTINUE 00001130
121 00000476 KKE = 20 00001140
122 00000477 JJ = 2 00001150
123 C ***** (( MARKET CONDITIONS )) *****00001160
124 00000478 DO 1100 KP = 1,KKE 00001170
125 00000479 WRITE(LP,110) 00001180
126 00000480 READ(CR,110) K, COSTPM(K), ( AVL(K,J),J = 1,JJ),EPRICE(K) 00001190
127 00000481 WRITE(LP,110) K, COSTPM(K), ( AVL(K,J),J = 1,JJ),EPRICE(K) 00001200
128 00000482 EMKT(K) = EPRICE(K)*1.5 00001210
129 00000483 C 00001220
130 00000484 COSTPM(K) = 20.0 + EPRICE(K)/5.0 00001230
131 00000485 C 00001240
132 00000486 EPRICE(K) = FPHICE(K)/2.0 00001250
133 00000487 IF ( K.EQ. 0 ) GO TO 1007 00001260
134 00000488 IF ( K.EQ. 0 ) GO TO 8000 00001270
135 00000489 WRITE(LP,110) K, COSTPM(K), ( AVL(K,J),J = 1,JJ),EPRICE(K) 00001280
136 00000490 1100 CONTINUE 00001290
137 00000491 1007 CONTINUE 00001300
138 00000492 C 00001310
139 00000493 WRITE(LP,110) 00001320
140 00000494 C 00001330
141 00000495 1000 CONTINUE 00001340
142 00000496 C 00001350
143 00000497 WRITE(LP,510) 00001360
144 00000498 C 00001370
145 00000499 READ(CR,10011) IW, KW 00001380
146 00000500 WRITE(LP,10011) IW, KW 00001390
147 00000501 IF ( IW.EQ. 0 ) IW = 1 00001400
148 00000502 IF ( ! (.GE. GO) ) GO TO 1011 00001410
149 00000503 LSM = -4 00001420
150 00000504 KWP = 5 00001430
151 00000505 C 00001440
152 00000506 MR = 0 00001450
153 00000507 FII = II 00001460
154 00000508 FJJ = JJ 00001470
155 00000509 C 00001480
156 00000510 IWT = 1 00001490
157 00000511 READ(CR,101) ( ( BRAND(I,IB), IB = 1,3), I = 1,1) 00001500
158 00000512 WRITE(LP,101) ( ( BRAND(I,IB), IB = 1,3), I = 1,1) 00001510
159 00000513 DO 6 I = 1,11 00001520
160 00000514 READ(CR,101) ( ( YMEMBR(I, MB, IB), IB = 1,3), MB = 1,6 ) 00001530
161 00000515 WRITE(LP,101) ( ( YMEMBR(I, MB, IB), IB = 1,3), MB = 1,6 ) 00001540
162 00000516 WRITE(LP,101) 00001550
163 00000517 C 00001560
164 00000518 ***** (( OPENING CONDITIONS )) *****00001570
165 00000519 K = 1 00001580
166 00000520 C 00001590
167 00000521 PRR = 0.0 00001600
168 00000522 C PRR = 10.0 00001610

```

```

XFORTRAN (V05-00) MAIN SOURCE LISTING 78-01-17 13:40:47 PAGE 4
169 C PRR = 6.0 00001620
170 C PRR = 8.0 00001630
171 C PTL = 1.0 00001640
172 C RDL = 50.0 00001650
173 C RANPD = 1.0 00001660
174 C CASMMN = 1000000.0 00001670
175 00001680
176 00000523 BDEM(K,1) = 0.0 00001690
177 00000524 BDEM(K,2) = 0.0 00001700
178 00000525 AVXAD(K) = 100.0 00001710
179 00000526 AVXAL(K,1) = 40.0 00001720
180 00000527 AVXAL(K,2) = 30.0 00001730
181 00000528 AVXSF(K,1) = 60.0 00001740
182 00000529 AVXSF(K,2) = 50.0 00001750
183 00000530 DMT(1) = 0.0 00001760
184 00000531 DMT(2) = 0.0 00001770
185 00000532 XRAMDT = 0.0 00001780
186 00000533 AVXR(K) = 150.0 00001790
187 00000534 333 FORMAT(8F10.0) 00001800
188 00000535 RESERV = IFIX(TOTALRCRESER)/100001000 00001810
189 00000536 CAPITL = IFIX(TOTALMCCAPTL)/100001000 00001820
190 00000537 DO 7 I = 1,11 00001830
191 00000538 XST(I) = 0.0 00001840
192 00000539 ZZZ(I) = 0.0 00001850
193 00000540 ZZZ(I) = 0.0 00001860
194 00000541 BCAPL(K,1) = BCAPL(1,1) 00001870
195 00000542 AD1( K,1 ) = AD1(1,1) 00001880
196 00000543 AD2( K,1 ) = AD2(1,1) 00001890
197 00000544 AD3( K,1 ) = AD3(1,1) 00001900
198 00000545 AD4( K,1 ) = AD4(1,1) 00001910
199 00000546 SF1(K,1,1) = SF1(1,1,1) 00001920
200 00000547 SF1(K,1,2) = SF1(1,1,2) 00001930
201 00000548 ERT(1,1) = 1.0 00001940
202 00000549 ERT(1,2) = 1.0 00001950
203 00000550 PGMAX(I) = PGMAX(1) 00001960
204 00000551 ERAND(I) = 0.0 00001970
205 00000552 BASALE(K,1) = 0.0 00001980
206 00000553 B7SALE(K,1) = 0.0 00001990
207 00000554 RCASH(K,1) = IFIX(TOTALRCASH)/100001000 00002000
208 00000555 BRCEVE(K,1) = IFIX(TOTALBRCEVE)/100001000 00002010
209 00000556 BF1XLB(K,1) = IFIX(TOTALBF1XLB)/100001000 00002020
210 00000557 BRFUND(K,1) = IFIX(TOTALBRFUND)/100001000 00002030
211 00000558 BGGUOS(K,1) = IFIX(TOTALBGGUOS)/100001000 00002040
212 00000559 BVEST(K,1) = IFIX(TOTALBVEST)/100001000 00002050
213 00000560 HASSET(I,1) = BCASH (K,1)+BRCEVE(K,1)+RFUND (K,1)+BGGUOS(K,1) 00002060
214 00000561 + BVEST(K,1) 00002070
215 00000562 BPAYBL(K,1) = IFIX(TOTALBPAYBL)/100001000 00002080
216 00000563 BRPERT(K,1) = IFIX(TOTALBRPERT)/100001000 00002090
217 00000564 PRDP (K,1) = HASSET(I,1)+BPAYBL (K,1)+RDEBT (K,1)+RF1XLB(K,1) 00002100
218 00000565 1 -CAPITL -RESERV 00002110
219 00000566 RSLD(IK,1+1) = 0.0 00002120
220 00000567 RSLD(IK,1+2) = 0.0 00002130
221 00000568 BF1NV(K,1) = BF1NV(1,1) 00002140
222 00000569 BR1NV(K,1,2) = BR1NV(1,1,2) 00002150
223 00000570 BR1NV(K,1,1) = BR1NV(1,1,1) 00002160
224 00000571 BT1NV(K,1) = BT1NV(1,1) 00002170
225 00000572 PGPAST(K,1) = MGPAST(1,1) 00002180
226 00000573 7 CONTINUE 00002190

```

```

XFORTRAN (V05-00)      MAIN                SOURCE LISTING                78-01-17    13:40:47    PAGE    5

227      C                                     00002200
228      C                                     00002210
229      C      A1 = 0.4                            00002220
230      C      A2 = 0.5                            00002230
231      C      A3 = 0.2                            00002240
232      C      A4 = 0.1                            00002250
233      C      A5 = 0.6                            00002260
234      C                                     00002270
235      C      ALPHA = 0.575                       00002280
236      C      BETA = 0.225                         00002290
237      C      GAMMA = 0.300                       00002300
238      C      DELTA = 0.250                       00002310
239      C      ZETA = 0.225                         00002320
240      C      ETA = 0.275                         00002330
241      C      EP = 2.0                             00002340
242      C                                     00002350
243      C      ***** (( DECISION MAKINGS )) ***** 00002360
244      C      WRITE(LP, $02)                       00002370
245      C      DD 110 KP = 1, KKE                   00002380
246      C      DD 110 IP = 1, I                     00002390
247      C      DD 111 K+1, P0(K+1), PRICE(K+1), ( ADLOC(K+1,J),J = 1,JJ), 00002400
248      C      1 ( SFORCE(K+1,J), J = 1,JJ), ADMAT(K+1), ( TRAN(K+1,J),J = 1,JJ), 00002410
249      C      2 TM(K+1),RAND(K+1),FUND(K+1),)      00002420
250      C      3 ( RESRC(K+1,LR), LR = 2*0), ( POLICY(1,1,LP), LP = 1, 3) 00002430
251      C      LR = 1                                00002440
252      C      RESRC(K+1,LR) = 1.0                   00002450
253      C      IF ( K.EQ. KKE) GO TO 1110           00002460
254      C      1110 CONTINUE                         00002470
255      C      1118 CONTINUE                         00002480
256      C                                     00002490
257      C      KK = KP-1                             00002500
258      C      FKK = KK                              00002510
259      C      KKK = KK*1                            00002520
260      C                                     00002530
261      C      DD 1121 I = 1, I                     00002540
262      C      WRITE(LP, 110)                       00002550
263      C      DD 1121 K = 1, KKK                   00002560
264      C      WRITE(LP, $05) K+1, P0(K+1), PRICE(K+1), ( ADLOC(K+1,J),J = 1,JJ), 00002570
265      C      1 ( SFORCE(K+1,J), J = 1,JJ), ADMAT(K+1), ( TRAN(K+1,J),J = 1,JJ), 00002580
266      C      2 TM(K+1),RAND(K+1),FUND(K+1),)      00002590
267      C      3 ( RESRC(K+1,LR), LR = 2*0), ( POLICY(1,1,LP), LP = 1, 3) 00002600
268      C      1121 CONTINUE                         00002610
269      C                                     00002620
270      C      DD 1123 I = 1, I                     00002630
271      C      1123 WRITE(LP, $05) KKK, I           00002640
272      C      WRITE(LP, 110)                       00002650
273      C      WRITE(LP, $002)                      00002660
274      C      DD 1122 K = 1, KKK                   00002670
275      C      FK = K                                00002680
276      C      WRITE(LP, 110)                       00002690
277      C      DD 1122 I = 1, I                     00002700
278      C      WRITE(LP, $05) K+1, P0(K+1), PRICE(K+1), ( ADLOC(K+1,J),J = 1,JJ), 00002710
279      C      1 ( SFORCE(K+1,J), J = 1,JJ), ADMAT(K+1), ( TRAN(K+1,J),J = 1,JJ), 00002720
280      C      2 TM(K+1),RAND(K+1),FUND(K+1),)      00002730
281      C      3 ( RESRC(K+1,LR), LR = 2*0), ( POLICY(1,1,LP), LP = 1, 3) 00002740
282      C      1122 CONTINUE                         00002750
283      C                                     00002760
284      C      DD 1108 K = 1, KKK                   00002770

```

```

XFORTRAN (V05-00)      MAIN                SOURCE LISTING                78-01-17    13:40:47    PAGE    6

285      C      DD 1108 J = 1, JJ                   00002780
286      C      DEMAND(K, J) = AVLM(K, J) * B[[      00002790
287      C      1108 CONTINUE                         00002800
288      C                                     00002810
289      C                                     00002820
290      C      GO TO 7713                            00002830
291      C      DD 7001 K = 2, KKK                   00002840
292      C      FK = K                                00002850
293      C      K1 = K*1                              00002860
294      C      K2 = K*2                              00002870
295      C      KR = K-1                              00002880
296      C      K11 = K*1                             00002890
297      C      DD 2235 I = 1, I                     00002900
298      C      IF ( PRICE(K+1), EQ. 0.0) PRICE(K, I) = 15.0 + PRR 00002910
299      C      IF ( ADMAT(K, I), EQ. 0.0) ADMAT(K, I) = AVRAD(K) 00002920
300      C      IF ( P0(K, I), EQ. 0.0) P0(K, I) = AVDM(K, I) + AVDM(K11, 2) 00002930
301      C      IF ( RAND(K, I), EQ. 0.0) RAND(K, I) = AVXRD(K) 00002940
302      C      DD 2235 J = 1, JJ                     00002950
303      C      IF ( ADLOC(K+1, J), EQ. 0.0) ADLOC(K+1, J) = AVXAL(K, J) 00002960
304      C      IF ( SFORCE(K+1, J), EQ. 0.0) SFORCE(K+1, J) = AVXSF(K, J) 00002970
305      C      IF ( TRAN(K+1, J), EQ. 0.0) TRAN(K+1, J) = AVDM(K+1, J) 00002980
306      C      2235 CONTINUE                         00002990
307      C      DD 2230 I = 1, I                     00003000
308      C      P0(K, I) = P0(K, I) * 1.2            00003010
309      C      PRICE(K, I) = PRICE(K, I) * 1.5      00003020
310      C      2236 CONTINUE                         00003030
311      C      COSTPM(K) = COSTPM(K) + PRR / B.0    00003040
312      C      CASHAD = 0.0                          00003050
313      C      TTM = 0.0                              00003060
314      C      XRAND(K) = 0.0                         00003070
315      C      XADMAT(K) = 0.0                       00003080
316      C      XADNAT(K) = 0.0                      00003090
317      C      DD 2401 J = 1, JJ                     00003100
318      C      XADLOC(K, J) = 0.0                    00003110
319      C      XSFORC(K, J) = 0.0                    00003120
320      C      XRAND(K, J) = 0.0                    00003130
321      C      2401 CONTINUE                         00003140
322      C      DD 2100 I = 1, I                     00003150
323      C      PRICE > 20.0                          00003160
324      C      PRICE(K, I) = AMAX1(PRICE(K, I), 20.0) 00003170
325      C      #PRICE > TM > 1.                    00003180
326      C      PTK3 = 0.0                             00003190
327      C      PTK = PRICE(K, I) / PTK3              00003200
328      C      TM(K, I) = AMAX1(TM(K, I), PTK)        00003210
329      C      TM(K, I) = AMIN1(TM(K, I), PTK)        00003220
330      C      TTM = TTM + TM(K, I)                  00003230
331      C      NAND > 50.                            00003240
332      C      RAND(K, I) = AMAX1(RAND(K, I), RNL)    00003250
333      C      TTT = TRAN(K, I+1) + TRAN(K, I+2)      00003260
334      C      TMU = TTT - BFINV(XB, J)              00003270
335      C      RTM = TRAN(K, I+1) + (TM(K, I) + TTT) + TRAN(K, I+2) 00003280
336      C      DD 2200 J = 1, JJ                     00003290
337      C      TM(K, I, J) = TRAN(K, I, J) + TMVBRM 00003300
338      C      RTM = 1.0 - RTM                       00003310
339      C      2200 CONTINUE                         00003320
340      C      2200 CONTINUE                         00003330
341      C      2200 CONTINUE                         00003340
342      C      2200 CONTINUE                         00003350

```

XFORTRAN (V05-00)	MAIN	SOURCE LISTING	78-01-17 13:40:47	PAGE 7
363	00000205	PQQ = PGPAST(KB+I)M1.20		00003390
364	00000206	PQ(K,I) = AMINI(PQ(K,I)+PQQ)		00003360
365	00000207	PQQ = PGPAST(KB+I)M0.80		00003370
366	00000208	PQ(K,I) = AMAXI(PQ(K,I)+PQQ)		00003380
367				00003390
368	00000209	C		00003400
369	00000210	CAPHEM(K,I) = 0.0		00003410
370	00000211	POLIT = PGMAX(I)M1.2		00003420
371	00000212	PQ(K,I) = AMINI(PQ(K,I)+POLIT)		00003430
372	00000213	PGUP = PQ(K,I)+PGMAX(I)		00003440
373	00000214	(F PGPUP .GT. 0.0) CAPHEM(K,I) = PGMAX(I)M0.2		00003450
374	00000215	PQMAX(I) = PGMAX(I)+CAPHEM(K,I)		00003460
375	00000216	FCAP(K,I) = PQMAX(I)		00003470
376				00003480
377	00000217	C 2100 CONTINUE		00003490
378	00000218	C		00003500
379	00000219	C ***** (MARKET SYSTEM) *****		00003510
380	00000220	DD 2222 I = 1+I		00003520
381	00000221	XADNAT(K) = XADNAT(K)+ADNAT(K,I)		00003530
382	00000222	XRAND(K) = XRAND(K)+RAND(K,I)		00003540
383	00000223	DD 2222 J = 1+J		00003550
384	00000224	XSFORC(K,I,J) = XSFORC(K,I,J)+SFORCE(K,I,J)		00003560
385	00000225	XADLOC(K,I,J) = XADLOC(K,I,J)+ADLOC(K,I,J)		00003570
386				00003580
387	00000226	C 2222 CONTINUE		00003590
388	00000227	AVXRD(K) = (AVXRD(K)+XRAND(K)/F11)/2.0		00003600
389	00000228	AVXAD(K) = (AVXAD(K)+XADNAT(K)/F11)/2.0		00003610
390	00000229	DD 2225 J = 1+J		00003620
391	00000230	AVXAL(K,I,J) = (AVXAL(K,I,J) + XADLOC(K,I,J)/F11)/2.0		00003630
392	00000231	AVXSF(K,I,J) = (AVXSF(K,I,J) + XSFORC(K,I,J)/F11)/2.0		00003640
393	00000232	DD 2225 I = 1+I		00003650
394				00003660
395	00000233	C		00003670
396	00000234	DD 2230 I = 1+I		00003680
397	00000235	ADT(K,I) = ADT(K,I)+AD2(K,I)*M2+AD3(K,I)*M3+AD4(K,I)*M4		00003690
398	00000236	DD 2230 J = 1+J		00003700
399	00000237	EPAST(K,I,J) = ADT(K,I) * SF1(K,I,J) * M5		00003710
400	00000238	DD 2230 CONTINUE		00003720
401				00003730
402	00000239	C		00003740
403	00000240	AVXMT(K) = TM/F11		00003750
404				00003760
405	00000241	C		00003770
406	00000242	AVXR02 = AVXR0(K)*M2.0		00003780
407	00000243	AVXR02 = AVXR0(K)*M2.0		00003790
408	00000244	DD 2210 J = 1+J		00003800
409	00000245	DMT(I,J) = DMT(I,J)+AVLW(K,I)		00003810
410	00000246	DD 2220 I = 1+I		00003820
411	00000247	DD 2220 J = 1+J		00003830
412	00000248	ERAND0(K,I,J) = 1.0		00003840
413	00000249	EFCRET(K,I,J) = 1.0		00003850
414				00003860
415	00000250	C 2220 CONTINUE		00003870
416	00000251	DD 2250 I = 1+I		00003880
417	00000252	DIF ADNAT(K,I) = AVXAD(K)		00003890
418	00000253	EADNAT(K,I) = (AMINI(ADNAT(K,I),AVXAD(K)))M1.0		00003900
419	00000254	+ (AMAXI(0.0,DIF))M0.6		00003910
420	00000255	EADMAT(K,I) = AMINI(EADNAT(K,I), AVXAD2)		00003920
421				00003930
422	00000256	C		00003940
423	00000257	AVXAL2 = AVXAL(K,I)*M2.0		00003950
424	00000258	AVXSF2 = AVXSF(K,I)*M2.0		00003960
425	00000259	DIF = ADLOC(K,I,J)+AVXAL(K,I)		00003970
426	00000260	EADLOC(K,I,J) = (AMINI(ADLOC(K,I,J)+AVXAL(K,I))M1.0		00003980
427	00000261	+ (AMAXI(0.0,DIF))M0.6		00003990
428	00000262	EADLOC(K,I,J) = AMINI(EADLOC(K,I,J), AVXAL2)		00004000
429				00004010
430	00000263	DIF = SFORCE(K,I,J)+AVXSF(K,I)		00004020
431	00000264	ESFORC(K,I,J) = (ESFORC(K,I,J)+AVXSF(K,I))M1.0		00004030
432	00000265	+ (AMAXI(0.0,DIF))M0.6		00004040
433	00000266	ESFORC(K,I,J) = AMINI(ESFORC(K,I,J), AVXSF2)		00004050
434				00004060
435	00000267	C 2240 CONTINUE		00004070
436	00000268	ERAND(K,I) = AMINI(ERAND(K,I), AVXR02)		00004080
437	00000269	ERAND(I) = ERAND(I)+ERAND(K,I)		00004090
438	00000270	XRANDT = XRANDT + ERAND(K,I)		00004100
439				00004110
440	00000271	C 2250 CONTINUE		00004120
441	00000272	ARANDT = XRANDT / F11		00004130
442				00004140
443	00000273	DD 2260 I = 1+I		00004150
444	00000274	DD 2260 J = 1+J		00004160
445	00000275	ERANDM(K,I,J) = ERANDT(I)*M2 / ARANDT/(DMT(I)+DMT(2)) * RANDP		00004170
446	00000276	ERE = ERANDM(K,I,J)		00004180
447	00000277	ERANDE(K,I,J) = AMAXI(1.0,ERE)		00004190
448	00000278	ELOYAL(K,I,J) = ERT(I,J)		00004200
449	00000279	C 2260 CONTINUE		00004210
450				00004220
451	00000280	DD 3300 J = 1+J		00004230
452	00000281	DD 3100 I = 1+I		00004240
453	00000282	FK = K		00004250
454				00004260
455	00000283	C		00004270
456	00000284	E(K,I,J) = EPAST(K,I,J)*MDELTA		00004280
457	00000285	1 READNAT(K,I)*MDELTA		00004290
458	00000286	2 READLOC(K,I)*MBETA		00004300
459	00000287	3 READFORC(K,I)*MGGMA		00004310
460	00000288	4 REHANDE(K,I)*MZETA		00004320
461	00000289	5 RELLOYAL(K,I)*MDELTA		00004330
462	00000290	A /MPRICE(K,I)*MFKT (K)*MNEP		00004340
463	00000291	ESIK(I,J) = E(K,I,J)*M(IPRICE(K,I)+EMTK (K))		00004350
464	00000292	1		00004360
465	00000293	2 /PRICE(K,I)*MNEP		00004370
466				00004380
467	00000294	C 3100 CONTINUE		00004390
468	00000295	XE(K,I) = 0.0		00004400
469	00000296	XES(K,I) = 0.0		00004410
470	00000297	DD 3110 I = 1+I		00004420
471	00000298	XES(K,I) = XES(K,I)+ES(K,I)		00004430
472	00000299	XFK(K,I) = XEK(K,I)+E(K,I)		00004440
473	00000300	C 3110 CONTINUE		00004450
474	00000301	(FK.EQ.1) XEST(J) = XFS(K,I)		00004460
475	00000302	ECRET(K,I) = SUM(I) XEST(K,I) XEST(J)		00004470
476	00000303	DM(K,I) = (DFHAND(K,I)+ECRET(K,I) + RDEMK(K,I))		00004480
477	00000304	ADVM(K,I) = DM(K,I)/F11		00004490
478	00000305	XEST(J) = (XEST(J)+XES(K,I))/2.0		00004500

XFORTRAN (V05-00)	MAIN	SOURCE LISTING	78-01-17 13:40:47	PAGE 8
401	00000250	DD 2240 J = 1+J		00003930
402	00000251	AVXAL2 = AVXAL(K,I)*M2.0		00003940
403	00000252	AVXSF2 = AVXSF(K,I)*M2.0		00003950
404	00000253	DIF = ADLOC(K,I,J)+AVXAL(K,I)		00003960
405	00000254	EADLOC(K,I,J) = (AMINI(ADLOC(K,I,J)+AVXAL(K,I))M1.0		00003970
406	00000255	+ (AMAXI(0.0,DIF))M0.6		00003980
407	00000256	EADLOC(K,I,J) = AMINI(EADLOC(K,I,J), AVXAL2)		00003990
408				00004000
409	00000257	DIF = SFORCE(K,I,J)+AVXSF(K,I)		00004010
410	00000258	ESFORC(K,I,J) = (ESFORC(K,I,J)+AVXSF(K,I))M1.0		00004020
411	00000259	+ (AMAXI(0.0,DIF))M0.6		00004030
412	00000260	ESFORC(K,I,J) = AMINI(ESFORC(K,I,J), AVXSF2)		00004040
413	00000261	C 2240 CONTINUE		00004050
414				00004060
415	00000262	ERAND(K,I) = AMINI(ERAND(K,I), AVXR02)		00004070
416	00000263	ERAND(I) = ERAND(I)+ERAND(K,I)		00004080
417	00000264	XRANDT = XRANDT + ERAND(K,I)		00004090
418				00004100
419	00000265	C 2250 CONTINUE		00004110
420	00000266	ARANDT = XRANDT / F11		00004120
421	00000267	DD 2260 I = 1+I		00004130
422	00000268	DD 2260 J = 1+J		00004140
423	00000269	ERANDM(K,I,J) = ERANDT(I)*M2 / ARANDT/(DMT(I)+DMT(2)) * RANDP		00004150
424	00000270	ERE = ERANDM(K,I,J)		00004160
425	00000271	ERANDE(K,I,J) = AMAXI(1.0,ERE)		00004170
426	00000272	ELOYAL(K,I,J) = ERT(I,J)		00004180
427	00000273	C 2260 CONTINUE		00004190
428	00000274	DD 3300 J = 1+J		00004200
429	00000275	DD 3100 I = 1+I		00004210
430	00000276	FK = K		00004220
431				00004230
432	00000277	C		00004240
433	00000278	E(K,I,J) = EPAST(K,I,J)*MDELTA		00004250
434	00000279	1 READNAT(K,I)*MDELTA		00004260
435	00000280	2 READLOC(K,I)*MBETA		00004270
436	00000281	3 READFORC(K,I)*MGGMA		00004280
437	00000282	4 REHANDE(K,I)*MZETA		00004290
438	00000283	5 RELLOYAL(K,I)*MDELTA		00004300
439	00000284	A /MPRICE(K,I)*MFKT (K)*MNEP		00004310
440	00000285	ESIK(I,J) = E(K,I,J)*M(IPRICE(K,I)+EMTK (K))		00004320
441	00000286	1		00004330
442	00000287	2 /PRICE(K,I)*MNEP		00004340
443				00004350
444	00000288	C 3100 CONTINUE		00004360
445	00000289	XE(K,I) = 0.0		00004370
446	00000290	XES(K,I) = 0.0		00004380
447	00000291	DD 3110 I = 1+I		00004390
448	00000292	XES(K,I) = XES(K,I)+ES(K,I)		00004400
449	00000293	XFK(K,I) = XEK(K,I)+E(K,I)		00004410
450	00000294	C 3110 CONTINUE		00004420
451	00000295	(FK.EQ.1) XEST(J) = XFS(K,I)		00004430
452	00000296	ECRET(K,I) = SUM(I) XEST(K,I) XEST(J)		00004440
453	00000297	DM(K,I) = (DFHAND(K,I)+ECRET(K,I) + RDEMK(K,I))		00004450
454	00000298	ADVM(K,I) = DM(K,I)/F11		00004460
455	00000299	XEST(J) = (XEST(J)+XES(K,I))/2.0		00004470
456	00000300			00004480
457	00000301			00004490
458	00000302			00004500


```

XFORTRAN (V05-D0)      MAIN              SOURCE LISTING              78-01-17  13:40:47      PAGE    9
450  00000289  DO 3130 I = 1+I 00004510
460  00000290  RE(K,I,J) = E(K,I,J) / NF(K,J) 00004520
461  00000291  ERET(I,J) = (ERT(I,J)+RE(K,I,J)*F(I))/2.0 00004530
462  00000292  C 00004540
463  00000292  C EFCRET(K,I,J) = ENANDE(K,I,J)*R0.112 00004550
464  00000293  C 00004560
465  00000293  C SLQ(K,I,J) = DM(K,J)*RE(K,I,J)+EFCRET(K,I,J) + BS(Q(KB,I,J)) 00004570
466  00000294  C 00004580
467  00000294  C ETR(K,I) = (TRM(K,I)/TTNMF(I)) 00004590
468  00000295  ETRM(K,I,J) = (ETM(K,I)+RE(K,I,J)*F(I))*R0.225 00004600
469  00000296  C 00004610
470  00000296  C EMINV(K,I,J) = RMINV(KB,I,J) 00004620
471  00000297  IF (ETRM(K,I,J)-LT.1.0) EMINV(K,I,J) = BMINVKR(I,J)+ETRM(K,I,J) 00004630
472  00000298  3130 CONTINUE 00004640
473  00000298  FQEM(K,J) = 0.0 00004650
474  00000300  TSEM = 0.0 00004660
475  00000301  DN 3131 I = 1+I 00004670
476  00000302  SLEM = ANAKI((SLQ(K,I,J)-EMINV(K,I,J)),0.0) 00004680
477  00000303  TSEM = TSEM+SLEM*0.0 00004690
478  00000304  FSLQ(K,I,J) = SLEM*0.2 00004700
479  00000305  FQEM(K,J) = FQEM(K,J)+SLEM*0.2 00004710
480  00000306  3131 CONTINUE 00004720
481  00000307  DO 3132 I = 1+I 00004730
482  00000308  SLQ(K,I,J) = SLQ(K,I,J)+TSEM*RE(K,I,J) 00004740
483  00000309  3132 CONTINUE 00004750
484  00000310  DO 3133 I = 1+I 00004760
485  00000311  C SQ(K,I,J) = EMINV(K,I,J) 00004770
486  00000312  IF (SLQ(K,I,J)-LT.EMINV(K,I,J)) SQ(K,I,J) = SLQ(K,I,J) 00004780
487  00000312  C 00004790
488  00000313  C 00004800
489  00000313  C S(K,I,J) = SQ(K,I,J)*PRICE(K,I)*R 1000.0 00004810
490  00000314  3133 CONTINUE 00004820
491  00000314  C 00004830
492  00000315  3300 CONTINUE 00004840
493  00000315  C 00004850
494  00000316  C 00004860
495  00000316  DO 3400 J = 1+JJ 00004870
496  00000317  XMINV(K,J) = 0.0 00004880
497  00000318  XSQ(K,J) = 0.0 00004890
498  00000319  XS(K,J) = 0.0 00004900
499  00000320  3400 CONTINUE 00004910
500  00000321  C 00004920
501  00000321  DO 3600 I = 1+I 00004930
502  00000322  DO 3600 J = 1+JJ 00004940
503  00000323  XMINV(K,J) = XMINV(K,J)+EMINV(K,I,J) 00004950
504  00000324  XSQ(K,J) = XSQ(K,J)+SQ(K,I,J) 00004960
505  00000325  XS(K,J) = XS(K,J)+S(K,I,J) 00004970
506  00000326  3600 CONTINUE 00004980
507  00000327  DO 3600 J = 1+JJ 00004990
508  00000328  BOEM(K,J) = FQEM(K,J) 00005000
509  00000329  3605 CONTINUE 00005010
510  00000330  C 00005020
511  00000330  DO 3610 J = 1+JJ 00005030
512  00000331  DO 3500 I = 1+I 00005040
513  00000332  S*RE(K,I,J) = SQ(K,I,J) + XSQ(K,J) # 100.0 00005050
514  00000333  3500 CONTINUE 00005060
515  00000334  XPRICE(K,J) = XS(K,J)/XSQ(K,J) /1000.0 00005070
516  00000335  AVXSQ(K,J) = XSQ(K,J)/F(I) 00005080

```

```

XFORTRAN (V05-D0)      MAIN              SOURCE LISTING              78-01-17  13:40:47      PAGE   10
517  00000336  3610 CONTINUE 00005090
518  00000337  C 00005100
519  00000337  C ##### ( ACCOUNTING SYSTEM ) ##### 00005110
520  00000338  C ##### 00005120
521  00000338  C ##### ( COST CALCULATION ) ##### 00005130
522  00000339  C 00005140
523  00000337  DO 1111 I = 1+I 00005150
524  00000338  TSALE(K,I) = 0.0 00005160
525  00000339  DO 10 J = 1+JJ 00005170
526  00000340  TSALE(K,I) = TSALE(K,I) + SQ(K,I,J) 00005180
527  00000341  14 CONTINUE 00005190
528  00000342  RANCK(K,I) = RANQ(K,I)*1000.0 00005200
529  00000343  TRM(K,I) = TSALE(K,I)*TRM(K,I)*1000.0 00005210
530  00000344  FTSALE(K,I) = BTSALE(KB,I)+TSALE(K,I) 00005220
531  00000345  FTASALE(K,I) = FTSALE(K,I)/FK 00005230
532  00000346  FTINVK(K,I) = 0.0 00005240
533  00000347  DO 40 J = 1+JJ 00005250
534  00000348  FMINV(K,I,J) = RMINV(KB,I,J)+TRM(K,I,J)-SQ(K,I,J) 00005260
535  00000349  FTINVK(I) = FTINVK(I)+FMINV(K,I,J) 00005270
536  00000350  40 CONTINUE 00005280
537  00000351  FTINVK(I) = PQ(K,I) 00005290
538  00000352  FTINVK(I) = FTINVK(I)+FMINV(K,I) 00005300
539  00000353  C 00005310
540  00000354  FIXC(K,I) = (BVEST(KB,I)/10.0) 00005320
541  00000354  FIXGM(K,I) = FIXC(K,I) # 0.10 00005330
542  00000355  FIXPK(K,I) = FIXC(K,I) # 0.10 00005340
543  00000356  FIXPC(K,I) = FIXC(K,I)*100.00 00005350
544  00000357  FPC = FIXPC(K,I)/100.0 00005360
545  00000358  C 00005370
546  00000358  UCOST(K,I) = FPC /PQ(K,I) + COSTPM(K,I) - (EFCRET(K,I)/2.0) 00005380
547  00000359  PVOST(K,I) = P(K,I)*UCOST(K,I) /R1000.0 00005390
548  00000360  PVOST(K,I) = PQ(K,I) # CNSTPM(K,I)*1000.0 00005400
549  00000361  FGONDS(K,I) = UCOST(K,I)*FTINVK(I)*1000.0 00005410
550  00000362  C 00005420
551  00000362  C COST(K,I) = PCOST(K,I)+FGONDS(I)*[1]-FGONDS(K,I) 00005430
552  00000363  C 00005440
553  00000363  C 00005450
554  00000363  REOCST(K,I) = 0.0 00005460
555  00000364  DO 12 LR = 2+R 00005470
556  00000365  [L+MR -ME] GO TO 0012 00005480
557  00000366  WESRCH(L+L) = 1.0 00005490
558  00000367  A012 CONTINUE 00005500
559  00000368  WESRCH(I) = REOCST(K,I) + WESRCH(L)*LR)*R0000.0 00005510
560  00000369  12 CONTINUE 00005520
561  00000370  ADLQCC = 0.0 00005530
562  00000371  SFORCC = 0.0 00005540
563  00000372  C 00005550
564  00000372  DO 2111 J = 1+JJ 00005560
565  00000373  ADLQCC = ADLQCC+ADLQCC(K,I,J) 00005570
566  00000374  SFORCC = SFORCC + SFORCC(K,I,J) 00005580
567  00000375  2111 CONTINUE 00005590
568  00000376  SCOST(K,I) = ( ADLQCC + SFORCC + ADNAT(K,I) ) # 1000.0 00005600
569  00000377  C 00005610
570  00000377  TRANC(K,I) = 0.0 00005620
571  00000378  DO 13 J = 1+JJ 00005630
572  00000379  TRANC(I) = TRANC(K,I) + TRANC(K,I,J)*R0.5*1000.0 00005640
573  00000380  13 CONTINUE 00005650
574  00000380  C 00005660

```

```

575 0000381 BT = AMINV(KB,1)+AMINV(KB,1)+2 00005670
576 0000382 BV = BT-BCAP(K+1) 00005680
577 0000383 IF(BV.LT.0.0) BV = 0.0 00005690
578 0000384 STCOST(K,1) = BT NO.501000.0*BVB(2.0+EPRICE(K)/10.0)*1000.00005700
579 0000385 C 00005710
580 0000385 TMCOST(K,1) = (ADLDC*-SFORCC+ADNAT(K,1))*1000.0*FIMC(K,1) 00005720
581 0000385 * + RECOST(K,1) 00005730
582 0000386 TMGC(K,1) = RANDC(K,1)+FMG(K,1) 00005740
583 0000386 1 +TMCOST(K,1) 00005750
584 0000386 2 +TMANC(K,1)+STCOST(K,1) 00005760
585 0000386 3 +TMC(K,1) 00005770
586 0000386 C 00005780
587 0000387 TCOST(K,1) = COST(K,1)+TMGC(K,1) 00005790
588 0000388 SJ = 0.0 00005800
589 0000389 DO 39 J = 1,JJ 00005810
590 0000390 39 SJ = SJ + S(K,1)+J 00005820
591 0000391 ST(K,1) = SJ 00005830
592 0000392 1111 CONTINUE 00005840
593 0000392 C 00005850
594 0000392 C ***** (( ACCOUNTING SVSTPM )) ***** 00005860
595 0000392 C 00005870
596 0000393 FRATE = SORT(10.0-EPRICE(K))/100.0 00005880
597 0000394 RSC = 2.0+EPRICE(K)/10.0 00005890
598 0000394 VESTC(K) = (.35.0+EPRICE(K)/2.0)*1000.0 00005910
599 0000394 C 00005920
600 0000397 DO 3333 I = 1,I1 00005930
601 0000397 C 00005940
602 0000397 C 00005950
603 0000398 FUNDC(K,1) = 0.0 00005960
604 0000398 RFUND = FUNDC(K,1)*1000.0 00005970
605 0000400 IF(RFUND) Z281,Z282,Z283 00005980
606 0000400 C 00005990
607 0000401 2281 CONTINUE 00005990
608 0000401 ABSR = ABS(RFUND) 00006000
609 0000403 BFMIN = AMIN(BFUND(K,1),ABSR) 00006010
610 0000404 FCASH(K,1) = BCASH(K,1)+BFMIN 00006020
611 0000405 RFUND = -BFMIN 00006030
612 0000406 FUNDC(K,1) = BFMIN*(0.010+FRATE) 00006040
613 0000407 GO TO Z284 00006050
614 0000407 C 00006060
615 0000408 2282 CONTINUE 00006070
616 0000409 RMIN = BCASH(K,1)-2.0*BPAYL(KB,1) 00006080
617 0000410 BFMIN = AMAX(RMIN,0.0) 00006090
618 0000411 FCASH(K,1) = BCASH(K,1)-BFMIN 00006100
619 0000412 RFUND = BFMIN 00006110
620 0000413 BM = BDEBT(K,1) -CASHM 00006120
621 0000414 BF = BFUND(K,1) 00006130
622 0000415 IF( BF,GT.0.0 .AND. BM,GT.0.0 ) GO TO Z285 00006140
623 0000416 GO TO Z284 00006150
624 0000417 2285 RFUND = -AMIN(BF,BM) 00006160
625 0000418 FCASH(K,1) = BCASH(K,1)+ABS(RFUND) 00006170
626 0000419 FUNDC(K,1) = ABS(RFUND)*(0.010+FRATE) 00006180
627 0000420 GO TO Z284 00006190
628 0000420 C 00006200
629 0000421 2283 CONTINUE 00006210
630 0000422 RMIN = BCASH(K,1)- CASHM 00006220
631 0000423 BFMIN = AMIN(RMIN,RFUND) 00006230
632 0000424 FCASH(K,1) = BCASH(K,1)-BFMIN 00006240

```

```

633 0000425 RFUND = BFMIN 00006250
634 0000426 GO TO Z284 00006260
635 0000427 2284 CONTINUE 00006270
636 0000428 C 00006280
637 0000428 FUNDC(K,1) = RFUND/1000.0 00006290
638 0000429 FFUND(K,1) = BFUND(K,1)-RFUND 00006300
639 0000430 SFUND(K,1) = BFUND(K,1) * FRATE 00006310
640 0000430 C 00006320
641 0000431 RFB = BFILX(K,1) 00006330
642 0000432 RBT = BDEBT(K,1) * W (0.010+FRATE) 00006340
643 0000433 RENT(K,1) = FUNDC(K,1) + RFB + RBT 00006350
644 0000434 C 00006360
645 0000434 TAX(K,1) = 0.0 00006370
646 0000435 PROFITG(K,1) = ST(K,1) - TCOST(K,1) 00006380
647 0000435 C 00006390
648 0000436 PROFIT(K,1) = PROFITG(K,1)+RENT(K,1)+SFUND(K,1) 00006400
649 0000437 PROFITX(K,1) = PROFIT(K,1) 00006410
650 0000438 IF( PROFIT(K,1).LT.0.0 ) GO TO 41 00006420
651 0000439 PROFITX(K,1) = PROFIT(K,1)*0.0 00006430
652 0000440 TAX(K,1) = PROFITX(K,1)*0.4 00006440
653 0000441 41 CONTINUE 00006450
654 0000441 C 00006460
655 0000441 C 00006470
656 0000442 VEST = CAPNEW(K,1)+VESTC(K,1)+RAND(K,1)/2.0*1000.0 00006480
657 0000443 FVEST(K,1) = VEST(K,1)+VEST 00006490
658 0000444 FRECEV(K,1) = ST(K,1)+RSC 00006500
659 0000445 FPAYBL(K,1) = (PCOST(K,1)+TMGC(K,1)) * RSC 00006510
660 0000446 PAYFIX = AMIN( BFILX(K,1) / 12.0, CASHM/2.0 ) 00006520
661 0000447 FFILX(K,1) = BFILX(K,1) + VEST - PAYFIX 00006530
662 0000447 C 00006540
663 0000448 FCASH(K,1) = FCASH(K,1)+RENT(K,1)+SFUND(K,1)+ST(K,1)*(1.0-RSC) 00006550
664 0000448 1 -TAX(K,1) - PAYFIX - (PCOST(K,1)+TMGC(K,1))*(1.0-RSC) 00006560
665 0000448 2 +FRECEV(K,1)+FPAYBL(K,1) 00006570
666 0000449 RMIN = FCASH(K,1)-CASHM 00006580
667 0000450 SMIN = PDEBT(K,1)-1000000.0 00006590
668 0000450 C 00006600
669 0000450 ***** 01) JLG0261 SIZE WRN. ***** 00006610
670 0000451 PAYDEB = AMIN( RMIN+ SMIN ) 00006620
671 0000452 FCASH(K,1) = FCASH(K,1)- PAYDEB 00006630
672 0000453 FDERT(K,1) = BDEBT(K,1) - PAYDEB 00006640
673 0000454 CLIBAT(K,1) = +PAYBL(K,1)+PDEBT(K,1) 00006650
674 0000455 CASSET(K,1) = FCASH(K,1)+FRECEV(K,1)+FFUND(K,1) 00006660
675 0000456 CASSET(K,1) = CASSET(K,1)+FGOODS(K,1) 00006670
676 0000457 FASSET(K,1) = CASSET(K,1)+FVEST(K,1) 00006680
677 0000458 TLICAP(K,1) = FASSET(K,1) 00006690
678 0000459 TLIBAT(K,1) = CLIBAT(K,1)+FFILX(K,1) 00006700
679 0000460 TAPL(K,1) = TLICAP(K,1)-TLIBAT(K,1)-RESERV 00006710
680 0000461 PROF(K,1) = TAPL(K,1)-CAPITL -PROFITX(K,1) 00006720
681 0000462 3333 CONTINUE 00006730
682 0000462 C 00006740
683 0000462 C ***** GENERAL BUSINESS EVALUATION ***** 00006750
684 0000463 IC = 0.0 00006760
685 0000464 TF = 0.0 00006770
686 0000465 UN 3350 I = 1,I1 00006780
687 0000466 TF = TF + FASALE(K,1) 00006790
688 0000467 XST(I) = XST(I) + CASSET(KB,1)+FASSET(K,1) /H.0 00006800
689 0000468 Z2(I) = Z2(I) + PROF(K,1) 00006810

```

```

XFORTRAN (V05-00)      MAIN      SOURCE LISTING      78-01-17  13:40:47      PAGE 13

689  00000469      RS(K,I) = ZZ(I)/XST(I) *100.0      00006810
690  00000470      IF(RS(K,I).LT.0.0) RS(K,I) = 0.0      00006820
691  00000471      TC = TC + RS(K,I)      00006830
692  00000472      3550 CONTINUE      00006840
693      C      00006850
694  00000473      XVALUE = 0.0      00006860
695  00000474      DO 50 I = 1,I1      00006870
696  00000475      SHAS = FSALE(K,I)/TFR100.0      00006880
697  00000476      IF(TC.EQ.0.0) TC = 1.0      00006890
698  00000477      SHAC = AS(K,I)/TC*10.0      00006900
699  00000478      GVALUE(K,I) = SQRT(SHAS**2+SHAC**2)      00006910
700  00000479      XVALUE = XVALUE+GVALUE(K,I)      00006920
701  00000480      ASSES(K,I) = SHAC      00006930
702  00000481      SALES(K,I) = SHAS      00006940
703  00000482      50 CONTINUE      00006950
704      C      00006960
705  00000483      DO 51 I = 1,I1      00006970
706  00000484      GSHARE(K,I) = GVALUE(K,I)/XVALUE*100.0      00006980
707  00000485      GSH(K,I) = GSHARE(K,I)      00006990
708  00000486      51 CONTINUE      00007000
709      C      00007010
710  00000487      DO 622 KR = 1,I1      00007020
711  00000488      GKX = -200.0      00007030
712  00000489      DO 611 I = 1,I1      00007040
713  00000490      IF(GSH(K,I).LT.GMAX) GO TO 611      00007050
714  00000491      GKX = GSH(K,I)      00007060
715  00000492      IMAX = I      00007070
716  00000493      611 CONTINUE      00007080
717  00000494      RANK(KR) = KR      00007090
718  00000495      GSH(KR) = GKX      00007100
719  00000496      622 CONTINUE      00007110
720      C      00007120
721      C ***** ( NEXT BEGINNING CONDITIONS + FINAL CONDITIONS ) *****      00007130
722      C      00007140
723  00000497      KI = K+1      00007150
724  00000498      DO 700 I = 1,I1      00007160
725  00000499      RTSALE(K,I) = FTSALE(K,I)      00007170
726  00000500      BASALE(K,I) = FSALE(K,I)      00007180
727  00000501      BCASH(K,I) = FCASH(K,I)      00007190
728  00000502      BRECEV(K,I) = FRECEV(K,I)      00007200
729  00000503      BFUND(K,I) = FFUND(K,I)      00007210
730  00000504      BGOODS(K,I) = FGOODS(K,I)      00007220
731  00000505      BVEST(K,I) = FVEST(K,I)      00007230
732  00000506      BPAYBL(K,I) = FPAYBL(K,I)      00007240
733  00000507      PROF(K,I) = FPROF(K,I)      00007250
734  00000508      FOBT(K,I) = FFBT(K,I)      00007260
735  00000509      BASSET(K,I) = FASSET(K,I)      00007270
736  00000510      BF(XL(K,I)) = FF(XL(K,I))      00007280
737  00000511      BF1W(K,I) = FFF1W(K,I)      00007290
738  00000512      BT1W(K,I) = FFT1W(K,I)      00007300
739  00000513      BCAP(K,I) = FCAP(K,I)      00007310
740      C      00007320
741  00000514      AD1(K,I) = EADHAT(K,I)      00007330
742  00000515      AD2(K,I) = ADI(K,I)      00007340
743  00000516      AD3(K,I) = ADZ(K,I)      00007350
744  00000517      AD4(K,I) = ADS(K,I)      00007360
745      C      00007370
746  00000518      PGPAST(K,I) = PG(K,I)      00007380

```

```

XFORTRAN (V05-00)      MAIN      SOURCE LISTING      78-01-17  13:40:47      PAGE 14

747      C      00007390
748  00000519      AVXAD(K,I) = AVXAD(K)      00007400
749  00000520      AVXR(K,I) = AVXR(K)      00007410
750      C      00007420
751  00000521      DO 6666 J = 1,JJ      00007430
752  00000522      AVXAL(K,I) = AVXAL(K,J)      00007440
753  00000523      AVXSF(K,I) = AVXSF(K,J)      00007450
754      C      00007460
755  00000524      BSLQ(K,I,J) = FSLQ(K,I,J)      00007470
756  00000525      BM1W(K,I,J) = F1W(K,I,J)      00007480
757  00000526      SF1C(K,I,J) = F1C(K,I,J)      00007490
758  00000527      6666 CONTINUE      00007500
759      C      00007510
760  00000528      7001 CONTINUE      00007520
761      C *****      00007530
762      C *****      00007540
763      C ***** ( INFORMATION OF BUSINESS BEHAVIOUR ) *****      00007550
764      C ***** ( BEGINNING CONDITIONS FOR DECISIONS ) *****      00007560
765      C *****      00007570
766      C ***** ( BUSINESS ANALYSIS ) *****      00007580
767      C *****      00007590
768      C *****      00007600
769      C *****      00007610
770  00000529      FRK = KK      00007620
771  00000530      DO 7702 K = 1,KK      00007630
772  00000531      DO 7702 I = 1,I1      00007640
773  00000532      FC = F1C(K,I)/RENT(K,I)/2.0      00007650
774  00000533      VC = VCST(K,I)/FC+RENT(K,I)/2.0      00007660
775  00000534      STK = ST(K,I)+SFUND(K,I)      00007670
776  00000535      BREP = FC(I,0)-VC*STK      00007680
777      C      00007690
778  00000536      STK = STK(I)+SFUND(K,I)      00007700
779  00000537      BANALS(K,I) = PROF(K,K,I) / ( BASSET(KB,I)+FASSET(K,I) ) *800.0      00007710
780  00000538      ZZZ(I) = ZZZ(I)+BANALS(K,I)/FRK      00007720
781  00000539      BANALS(K,I,2) = PROF(K,K,I) / ST(K,I)*100.0      00007730
782  00000540      BANALS(K,I,4) = ST(K,I) / ( BASSET(KB,I)+FASSET(K,I) ) *8      00007740
783  00000541      BANALS(K,I,5) = ST(K,I) / ( BRECEV(KB,I)+FRECEV(K,I) ) *8      00007750
784  00000542      BANALS(K,I,6) = ST(K,I) / ( BGOODS(KB,I)+FGOODS(K,I) ) *8      00007760
785  00000543      BANALS(K,I,7) = ST(K,I) / ( BVEST(KB,I)+FVEST(K,I) ) *8      00007770
786  00000544      BANALS(K,I,8) = ST(K,I) / ( BPAYBL(KB,I)+FPAYBL(K,I) ) *8      00007780
787  00000545      BANALS(K,I,9) = CASSET(K,I)/CL1ART(K,I)*100      00007790
788  00000546      BANALS(K,I,10) = CASSET(K,I)/CL1ART(K,I)*100      00007800
789  00000547      BANALS(K,I,11) = FVST(K,I)/TCAP1L(K,I)*100      00007810
790  00000548      BANALS(K,I,12) = FVEST(K,I)/TCAP1L(K,I)+FF(XL(K,I))*100      00007820
791  00000549      BANALS(K,I,13) = TCAP1L(K,I)/FASSET(K,I)*100      00007830
792  00000550      BANALS(K,I,14) = HREP      00007840
793  00000551      BANALS(K,I,15) = ST(K,I)/BREP *8 100.0      00007850
794  00000552      HANALS(K,I,16) = H1W(KB,I,1) / SLQ(K,I,1) * 100.0      00007860
795  00000553      HANALS(K,I,17) = H1W(KB,I,2) / SLQ(K,I,2) * 100.0      00007870
796  00000554      7702 CONTINUE      00007880
797  00000555      7705 CONTINUE      00007890
798      C *****      00007900
799  00000556      DO 7711 I = 1,I1      00007910
800  00000557      UH1 7711 K = 1,KK      00007920
801  00000558      HALS = ABS( H5(K,K,I) - HANALS(K,I,1) ) / RS(K,K,I) * 100.0      00007930
802  00000559      HANALS(K,I,10) = HALS      00007940
803  00000560      7711 CONTINUE      00007950
804      C *****      00007960
805  00000561      *****WRITE(2,/,753) I,X,K(I),BANALS(K,I,1),BANALS(K,I,2),BANALS(K,I,15),U0007960

```

```

R05 C----- RWALS(K1,4),FASST(K1), SY(K,1) 00007970
R06 C----- WRITE(2753) (1,K,UCDST(K,1), SCOST(K,1) 00007980
R07 7712 CONTINUE 00007990
R08 C 00008000
R09 C 00008010
R10 C ***** INFORMATION OF MARKET SYSTEM ***** 00008020
R11 IF (1W.EQ.0) 1W = 1 00008030
R12 0000562 DO 7722 (WRITE = 1, 1W) 00008040
R13 0000563 7713 CONTINUE 00008050
R14 C 00008060
R15 L2 = 40 00008070
R16 0000565 KK = 5 00008080
R17 C 00008090
R18 C 11 = 5 00008100
R19 CCCCCCCCC 00008110
R20 L1 = 1 00008120
R21 DD 244 (111 = 1,11) 00008130
R22 C TITLE AND INITIAL CONDITION LIST 00008140
R23 225 CONTINUE 00008150
R24 C FORMAT(221-225) 00008160
R25 K = 1 00008170
R26 LS = 1111 00008180
R27 LE = 1111 00008190
R28 225 CONTINUE 00008200
R29 WRITE(LP,200) 00008210
R30 201 FORMAT(1M ,T11,30(1M), 'STRATEGIC MARKETING SIMULATION SYSTEM', 00008220
R31 * 35(1M) / ) 00008230
R32 0000575 WRITE(LP,211) (1111, (BRAND(1111,18),1B = 1,3) 00008240
R33 0000576 211 FORMAT(1M ,T11, 'COMPANY NO. & BRAND ',T35,14, 2X, 244A2 /) 00008250
R34 0000577 WRITE(LP,212) ((TMEBR(1111,MB,1B),1B = 1,3),MB = 1,8) 00008260
R35 0000578 212 FORMAT(1M ,T11, 'MEMBER ',T35,8(244A2)) 00008270
R36 0000579 WRITE(LP,200) 00008280
R37 0000580 WRITE(LP,294) (1,1 = LS,LE) 00008290
R38 0000581 WRITE(LP,210) (HEAD(002,1),L = L1,L2) 00008300
R39 0000582 WRITE(LP,260) (HEAD(003,1),L = L1,L2), (BCAP (K,1) ) 1 = LS,LE) 00008310
R40 0000583 WRITE(LP,260) (HEAD(004,1),L = L1,L2), (POPAST(K,1) ) 1 = LS,LE) 00008320
R41 0000584 WRITE(LP,260) (HEAD(005,1),L = L1,L2), (BTSALE(K,1) ) 1 = LS,LE) 00008330
R42 0000585 WRITE(LP,260) (HEAD(006,1),L = L1,L2), (BASALE(K,1) ) 1 = LS,LE) 00008340
R43 0000586 WRITE(LP,260) (HEAD(007,1),L = L1,L2), (BASALE(K,1) ) 1 = LS,LE) 00008350
R44 0000587 WRITE(LP,298) (HEAD(008,1),L = L1,L2) 00008360
R45 0000588 WRITE(LP,260) (HEAD(009,1),L = L1,L2), (BTINV (K,1) ) 1 = LS,LE) 00008370
R46 0000589 WRITE(LP,260) (HEAD(010,1),L = L1,L2), (BFIN (K,1) ) 1 = LS,LE) 00008380
R47 0000590 WRITE(LP,260) (HEAD(011,1),L = L1,L2), (BFINV (K,1,2)) 1 = LS,LE) 00008390
R48 0000591 WRITE(LP,260) (HEAD(012,1),L = L1,L2), (BMINV (K,1,2)) 1 = LS,LE) 00008400
R49 0000592 WRITE(LP,298) (HEAD(013,1),L = L1,L2) 00008410
R50 0000593 WRITE(LP,260) (HEAD(014,1),L = L1,L2), (BSLD (K,1,2)) 1 = LS,LE) 00008420
R51 0000594 WRITE(LP,260) (HEAD(015,1),L = L1,L2), (BSLD (K,1,2)) 1 = LS,LE) 00008430
R52 0000595 WRITE(LP,298) (HEAD(016,1),L = L1,L2) 00008440
R53 0000596 WRITE(LP,260) (HEAD(017,1),L = L1,L2), (AD1 (K,1) ) 1 = LS,LE) 00008450
R54 0000597 WRITE(LP,260) (HEAD(018,1),L = L1,L2), (AD2 (K,1) ) 1 = LS,LE) 00008460
R55 0000598 WRITE(LP,260) (HEAD(019,1),L = L1,L2), (AD3 (K,1) ) 1 = LS,LE) 00008470
R56 0000599 WRITE(LP,260) (HEAD(020,1),L = L1,L2), (AD4 (K,1) ) 1 = LS,LE) 00008480
R57 0000600 WRITE(LP,260) (HEAD(021,1),L = L1,L2), (SF1 (K,1,2)) 1 = LS,LE) 00008490
R58 0000601 WRITE(LP,260) (HEAD(022,1),L = L1,L2), (SF1 (K,1,2)) 1 = LS,LE) 00008500
R59 0000602 WRITE(LP,298) 00008510
R60 C 0000520
R61 I2=230 0000520
R62 C 0000530
R63 LS = LSM 0000530

```

```

XFORTRAN (V05-00) MAIN SOURCE LISTING 78-01-17 13:40:47 PAGE 16
R63 IF( KK .GT. KW ) LS = LS+( KK-KW ) 00008550
R64 IF ( KM.ED.0) LS = LSM 00008560
R65 229 CONTINUE 00008570
R66 LS = LS + KW 00008580
R67 LE = LS + KW - 1 00008590
R68 IF (LE .GT. KK) LE = KK 00008600
R69 LFL = LSL 00008610
R70 LEI = LEL 00008620
R71 C 00008630
R72 WRITE(LP,200) 00008640
R73 WRITE(LP,212) 00008650
R74 202 FORMAT(1M ,T11,15(2H +), 'INFORMATION OF BUSINESS BEHAVIOUR', 00008660
R75 * 20(2H) ) 00008670
R76 0000615 WRITE(LP,211) (1111, (BRAND(1111,18),1B = 1,3) 00008680
R77 0000616 WRITE(LP,212) ((TMEBR(1111,MB,1B),1B = 1,3),MB = 1,8) 00008690
R78 1 = 1111 00008700
R79 0000618 WRITE(LP,200) 00008710
R80 0000619 WRITE(LP,292) (K,K = LS,LE) 00008720
R81 0000620 WRITE(LP,210) (HEAD(023,1),L = L1,L2) 00008730
R82 0000621 WRITE(LP,260) (HEAD(024,1),L = L1,L2), (PB (K,1) ) K = LS,LE) 00008740
R83 0000622 WRITE(LP,260) (HEAD(025,1),L = L1,L2), (PRICE (K,1) ) K = LS,LE) 00008750
R84 0000623 WRITE(LP,260) (HEAD(026,1),L = L1,L2), (ADLOC (K,1,1)) K = LS,LE) 00008760
R85 0000624 WRITE(LP,260) (HEAD(027,1),L = L1,L2), (ADLOC (K,1,2)) K = LS,LE) 00008770
R86 0000625 WRITE(LP,260) (HEAD(028,1),L = L1,L2), (SFORCE(K,1,2)) K = LS,LE) 00008780
R87 0000626 WRITE(LP,260) (HEAD(029,1),L = L1,L2), (SFORCE(K,1,2)) K = LS,LE) 00008790
R88 0000627 WRITE(LP,260) (HEAD(030,1),L = L1,L2), (ADMAT (K,1) ) K = LS,LE) 00008800
R89 0000628 WRITE(LP,260) (HEAD(031,1),L = L1,L2), (TRAN (K,1,2)) K = LS,LE) 00008810
R90 0000629 WRITE(LP,260) (HEAD(032,1),L = L1,L2), (TRAN (K,1,2)) K = LS,LE) 00008820
R91 0000630 WRITE(LP,260) (HEAD(033,1),L = L1,L2), (ITW (K,1) ) K = LS,LE) 00008830
R92 0000631 WRITE(LP,260) (HEAD(034,1),L = L1,L2), (RAND (K,1) ) K = LS,LE) 00008840
R93 0000632 WRITE(LP,260) (HEAD(035,1),L = L1,L2), (FUND (K,1) ) K = LS,LE) 00008850
R94 0000633 WRITE(LP,210) (HEAD(048,1),L = L1,L2) 00008860
R95 0000634 WRITE(LP,260) (HEAD(049,1),L = L1,L2), (TSALE (K,1) ) K = LS,LE) 00008870
R96 0000635 WRITE(LP,260) (HEAD(050,1),L = L1,L2), (TSALE (K,1) ) K = LS,LE) 00008880
R97 0000636 WRITE(LP,260) (HEAD(051,1),L = L1,L2), (SQ (K,1,2)) K = LS,LE) 00008890
R98 0000637 WRITE(LP,260) (HEAD(052,1),L = L1,L2), (ST (K,1) ) K = LS,LE) 00008900
R99 0000638 WRITE(LP,260) (HEAD(053,1),L = L1,L2), (TECOST (K,1) ) K = LS,LE) 00008910
R00 0000639 WRITE(LP,260) (HEAD(054,1),L = L1,L2), (S (K,1,2)) K = LS,LE) 00008920
R01 0000640 WRITE(LP,210) (HEAD(036,1),L = L1,L2) 00008930
R02 0000641 WRITE(LP,260) (HEAD(055,1),L = L1,L2), (UCDST (K,1) ) K = LS,LE) 00008940
R03 0000642 WRITE(LP,260) (HEAD(056,1),L = L1,L2), (PVCDST (K,1) ) K = LS,LE) 00008950
R04 0000643 WRITE(LP,260) (HEAD(039,1),L = L1,L2), (FIXPC (K,1) ) K = LS,LE) 00008960
R05 0000644 WRITE(LP,260) (HEAD(060,1),L = L1,L2), (SCDST (K,1) ) K = LS,LE) 00008970
R06 0000645 WRITE(LP,260) (HEAD(041,1),L = L1,L2), (FIXMC (K,1) ) K = LS,LE) 00008980
R07 0000646 WRITE(LP,260) (HEAD(042,1),L = L1,L2), (STCOST (K,1) ) K = LS,LE) 00008990
R08 0000647 WRITE(LP,260) (HEAD(043,1),L = L1,L2), (TMC (K,1) ) K = LS,LE) 00009000
R09 0000648 WRITE(LP,260) (HEAD(044,1),L = L1,L2), (TREN (K,1) ) K = LS,LE) 00009010
R10 0000649 WRITE(LP,260) (HEAD(045,1),L = L1,L2), (FIXGM (K,1) ) K = LS,LE) 00009020
R11 0000650 WRITE(LP,260) (HEAD(046,1),L = L1,L2), (RANDC (K,1) ) K = LS,LE) 00009030
R12 0000651 WRITE(LP,260) (HEAD(047,1),L = L1,L2), (TECOST (K,1) ) K = LS,LE) 00009040
R13 0000652 WRITE(LP,210) (HEAD(055,1),L = L1,L2) 00009050
R14 0000653 WRITE(LP,260) (HEAD(056,1),L = L1,L2), (FCAP (K,1) ) K = LS,LE) 00009060
R15 0000654 WRITE(LP,260) (HEAD(057,1),L = L1,L2), (POPAST (K,1) ) K = LS,LE) 00009070
R16 0000655 WRITE(LP,260) (HEAD(048,1),L = L1,L2), (FTSALE (K,1) ) K = LS,LE) 00009080
R17 0000656 WRITE(LP,260) (HEAD(049,1),L = L1,L2), (FASALE (K,1) ) K = LS,LE) 00009090
R18 0000657 WRITE(LP,260) (HEAD(050,1),L = L1,L2), (RRE (K,1) ) K = LS,LE) 00009100
R19 0000658 WRITE(LP,260) (HEAD(051,1),L = L1,L2), (FTINV (K,1) ) K = LS,LE) 00009110
R20 0000659 WRITE(LP,260) (HEAD(052,1),L = L1,L2), (FFINV (K,1) ) K = LS,LE) 00009120

```

KFDATRAM (V05-00)	MAIN	SOURCE LISTING	78-01-17	13:40:47	PAGE 17
921	00000600	WRITE(LP,260) (HEAD(063)L) L = L1+L2) * (FMINV(K,1)) K = L5+LE)			00009130
922	00000601	WRITE(LP,260) (HEAD(064)L) L = L1+L2) * (FMINV(K,1+2)) K = L5+LE)			00009140
923	00000602	WRITE(LP,260) (HEAD(065)L) L = L1+L2) * (FSLG(K,1)) K = L5+LE)			00009150
924	00000603	WRITE(LP,260) (HEAD(066)L) L = L1+L2) * (FSLG(K,1)) K = L5+LE)			00009160
925	00000604	WRITE(LP,260)			00009170
926	00000605	IF(LC.LT.KK) GO TO 229			00009180
927	00000606	230 CONTINUE			00009190
928					00009200
929					00009210
930	00000607	LS = LSM			00009220
931	00000608	IF(KK .GT. KW) LS = LS*(KK-KW)			00009230
932	00000609	IF(KW.EQ.O) LS = LSM			00009240
933	00000610	234 CONTINUE			00009250
934	00000611	LS = LS * KWP			00009260
935	00000612	LF = LS * KWP - 1			00009270
936	00000613	IF(LC.GT.KK) LE = KK			00009280
937					00009290
938	00000614				00009300
939	00000615	WRITE(LP,200)			00009310
940	00000616	203 FORMAT(1H ,111,15(2M ,1) ,1'FINANCIAL STRUCTURE',27(2M ,1)			00009320
941	00000617	WRITE(LP,211) 11111(BRAND(1111)B) ,1B = 1,3)			00009330
942	00000618	WRITE(LP,290)			00009340
943	00000619	(K,K = L5+LE)			00009350
944	00000620	WRITE(LP,210) (HEAD(115)L) L = L1+L2)			00009360
945	00000621	WRITE(LP,260) (HEAD(116)L) L = L1+L2) * (FASSET(K,1)) K = L5+LE)			00009370
946	00000622	WRITE(LP,260) (HEAD(117)L) L = L1+L2) * (CASSET(K,1)) K = L5+LE)			00009380
947	00000623	WRITE(LP,260) (HEAD(118)L) L = L1+L2) * (QASSET(K,1)) K = L5+LE)			00009390
948	00000624	WRITE(LP,260) (HEAD(119)L) L = L1+L2) * (FCASH(K,1)) K = L5+LE)			00009400
949	00000625	WRITE(LP,260) (HEAD(120)L) L = L1+L2) * (FFRCEV(K,1)) K = L5+LE)			00009410
950	00000626	WRITE(LP,260) (HEAD(121)L) L = L1+L2) * (FFUND(K,1)) K = L5+LE)			00009420
951	00000627	WRITE(LP,260) (HEAD(122)L) L = L1+L2) * (FGOODS(K,1)) K = L5+LE)			00009430
952	00000628	WRITE(LP,260) (HEAD(123)L) L = L1+L2) * (FVEST(K,1)) K = L5+LE)			00009440
953	00000629	WRITE(LP,260) (HEAD(124)L) L = L1+L2) * (ITLCAP(K,1)) K = L5+LE)			00009450
954	00000630	WRITE(LP,260) (HEAD(125)L) L = L1+L2) * (ITLIABT(K,1)) K = L5+LE)			00009460
955	00000631	WRITE(LP,260) (HEAD(126)L) L = L1+L2) * (CLIABT(K,1)) K = L5+LE)			00009470
956	00000632	WRITE(LP,260) (HEAD(127)L) L = L1+L2) * (FRAYAL(K,1)) K = L5+LE)			00009480
957	00000633	WRITE(LP,260) (HEAD(128)L) L = L1+L2) * (FDEBT(K,1)) K = L5+LE)			00009490
958	00000634	WRITE(LP,260) (HEAD(129)L) L = L1+L2) * (FFIXL(K,1)) K = L5+LE)			00009500
959	00000635	WRITE(LP,260) (HEAD(130)L) L = L1+L2) * (RESERV(K,1)) K = L5+LE)			00009510
960	00000636	WRITE(LP,260) (HEAD(131)L) L = L1+L2) * (TCAPITL(K,1)) K = L5+LE)			00009520
961	00000637	WRITE(LP,260) (HEAD(132)L) L = L1+L2) * (CAPITL(K,1)) K = L5+LE)			00009530
962	00000638	WRITE(LP,260) (HEAD(133)L) L = L1+L2) * (PRDF(K,1)) K = L5+LE)			00009540
963	00000639	WRITE(LP,260) (HEAD(134)L) L = L1+L2) * (PRDF1(K,1)) K = L5+LE)			00009550
964	00000700	WRITE(LP,290)			00009560
965	00000701	WRITE(LP,298) (HEAD(135)L) L = L1+L2)			00009570
966	00000702	WRITE(LP,260) (HEAD(136)L) L = L1+L2) * (ST(K,1)) K = L5+LE)			00009580
967	00000703	WRITE(LP,260) (HEAD(137)L) L = L1+L2) * (COST(K,1)) K = L5+LE)			00009590
968	00000704	WRITE(LP,260) (HEAD(138)L) L = L1+L2) * (TMGC(K,1)) K = L5+LE)			00009600
969	00000705	WRITE(LP,260) (HEAD(139)L) L = L1+L2) * (TCOST(K,1)) K = L5+LE)			00009610
970	00000706	WRITE(LP,260) (HEAD(140)L) L = L1+L2) * (PRDFTG(K,1)) K = L5+LE)			00009620
971	00000707	WRITE(LP,260) (HEAD(141)L) L = L1+L2) * (SFUND(K,1)) K = L5+LE)			00009630
972	00000708	WRITE(LP,260) (HEAD(142)L) L = L1+L2) * (REMT(K,1)) K = L5+LE)			00009640
973	00000709	WRITE(LP,260) (HEAD(143)L) L = L1+L2) * (PRDF1T(K,1)) K = L5+LE)			00009650
974	00000710	WRITE(LP,260) (HEAD(144)L) L = L1+L2) * (TAX(K,1)) K = L5+LE)			00009660
975	00000711	WRITE(LP,260) (HEAD(145)L) L = L1+L2) * (PRDF1K(K,1)) K = L5+LE)			00009670
976	00000712	WRITE(LP,290)			00009680
977	00000713	WRITE(LP,298) (HEAD(146)L) L = L1+L2)			00009690
978	00000714	WRITE(LP,260) (HEAD(147)L) L = L1+L2) * (BANALS(K,1)) K = L5+LE)			00009700

KFDATRAM (V05-00)	MAIN	SOURCE LISTING	78-01-17	13:40:47	PAGE 18
979	00000715	WRITE(LP,260) (HEAD(148)L) L = L1+L2) * (BANALS(K,1,2)) K = L5+LE)			00009710
980	00000716	WRITE(LP,260) (HEAD(149)L) L = L1+L2) * (BANALS(K,1,3)) K = L5+LE)			00009720
981	00000717	WRITE(LP,260) (HEAD(150)L) L = L1+L2) * (BANALS(K,1,4)) K = L5+LE)			00009730
982	00000718	WRITE(LP,260) (HEAD(151)L) L = L1+L2) * (BANALS(K,1,5)) K = L5+LE)			00009740
983	00000719	WRITE(LP,260) (HEAD(152)L) L = L1+L2) * (BANALS(K,1,6)) K = L5+LE)			00009750
984	00000720	WRITE(LP,260) (HEAD(153)L) L = L1+L2) * (BANALS(K,1,7)) K = L5+LE)			00009760
985	00000721	WRITE(LP,260) (HEAD(154)L) L = L1+L2) * (BANALS(K,1,8)) K = L5+LE)			00009770
986	00000722	WRITE(LP,298) (HEAD(155)L) L = L1+L2)			00009780
987	00000723	WRITE(LP,260) (HEAD(156)L) L = L1+L2) * (BANALS(K,1,9)) K = L5+LE)			00009790
988	00000724	WRITE(LP,260) (HEAD(157)L) L = L1+L2) * (BANALS(K,1,10)) K = L5+LE)			00009800
989	00000725	WRITE(LP,260) (HEAD(158)L) L = L1+L2) * (BANALS(K,1,11)) K = L5+LE)			00009810
990	00000726	WRITE(LP,260) (HEAD(159)L) L = L1+L2) * (BANALS(K,1,12)) K = L5+LE)			00009820
991	00000727	WRITE(LP,260) (HEAD(160)L) L = L1+L2) * (BANALS(K,1,13)) K = L5+LE)			00009830
992	00000728	WRITE(LP,260) (HEAD(161)L) L = L1+L2) * (BANALS(K,1,14)) K = L5+LE)			00009840
993	00000729	WRITE(LP,260) (HEAD(162)L) L = L1+L2) * (BANALS(K,1,15)) K = L5+LE)			00009850
994	00000730	WRITE(LP,260) (HEAD(163)L) L = L1+L2)			00009860
995	00000731	WRITE(LP,260) (HEAD(164)L) L = L1+L2) * (BANALS(K,1,16)) K = L5+LE)			00009870
996	00000732	WRITE(LP,260) (HEAD(165)L) L = L1+L2) * (BANALS(K,1,17)) K = L5+LE)			00009880
997	00000733	WRITE(LP,290)			00009890
998	00000734	IF(LC.LT.KK) GO TO 234			00009900
999	00000735	235 CONTINUE			00009910
1000					00009920
1001	00000736	LS = LSM			00009930
1002	00000737	IF(KK .GT. KW) LS = LS*(KK-KW)			00009940
1003	00000738	IF(KW.EQ.O) LS = LSM			00009950
1004	00000739	240 CONTINUE			00009960
1005	00000740	LS = LS * KWP			00009970
1006	00000741	LE = LS * KWP - 1			00009980
1007	00000742	IF(LC.GT.KK) LE = KK			00009990
1008					00010000
1009	00000743	WRITE(LP,290)			00010010
1010	00000744	WRITE(LP,298)			00010020
1011	00000745	204 FORMAT(1H ,111,15(2M ,1) ,1'INFORMATION OF MARKET RESEARCH',21(2M ,1)			00010030
1012	00000746	WRITE(LP,211) 11111(BRAND(1111)B) ,1B = 1,3)			00010040
1013	00000747	WRITE(LP,290)			00010050
1014	00000748	WRITE(LP,292) (K,K = L5+LE)			00010060
1015	00000749	WRITE(LP,210) (HEAD(067)L) L = L1+L2)			00010070
1016	00000750	WRITE(LP,291) (HEAD(068)L) L = L1+L2)			00010080
1017	00000751	WRITE(LP,270) (HEAD(069)L) L = L1+L2)			00010090
1018	00000752	DO 241 I = 1,11			00010100
1019	00000753	WRITE(LP,271) (BRAND(111)B) ,1B = 1,3) , (PRICE(#,1)) K = L5+LE)			00010110
1020	00000754	WRITE(LP,294) (HEAD(070)L) L = L1+L2)			00010120
1021	00000755	WRITE(LP,260) (HEAD(075)L) L = L1+L2) * (XPRICE(K,1)) K = L5+LE)			00010130
1022	00000756	WRITE(LP,260) (HEAD(076)L) L = L1+L2) * (AVXAL(K,1)) K = L5+LE)			00010140
1023	00000757	WRITE(LP,260) (HEAD(077)L) L = L1+L2) * (AVXAL(K,2)) K = L5+LE)			00010150
1024	00000758	WRITE(LP,260) (HEAD(078)L) L = L1+L2) * (AVXAL(K,3)) K = L5+LE)			00010160
1025	00000759	WRITE(LP,260) (HEAD(079)L) L = L1+L2) * (AVXPS(K,2)) K = L5+LE)			00010170
1026	00000760	WRITE(LP,260) (HEAD(080)L) L = L1+L2) * (AVXAD(K,1)) K = L5+LE)			00010180
1027	00000761	WRITE(LP,294) (HEAD(081)L) L = L1+L2)			00010190
1028	00000762	WRITE(LP,260) (HEAD(072)L) L = L1+L2) * (AVDM(K,1)) K = L5+LE)			00010200
1029	00000763	J = 1			00010210
1030	00000764	WRITE(LP,294) J			00010220
1031	00000765	240 FORMAT(1H ,146 ,1'K',11)			00010230
1032	00000766	J = 2			00010240
1033	00000767	WRITE(LP,260) (HEAD(072)L) L = L1+L2) * (AVDM(K,2)) K = L5+LE)			00010250
1034	00000768	WRITE(LP,260) J			00010260
1035	00000769	DO 242 I = 1, J			00010270
1036	00000770	WRITE(LP,294) (HEAD(073)L) L = L1+L2) J			00010280

```

1037 00000771 DO 247 I = 1,I1
1038 00000772 242 WRITE(LP+275)(I,(BRAND(I),L),L = 1,3),(SLQ (K,I),J),K = LS,LE)
1039 00000773 WRITE(LP+270) (HEAD(074),L),L = L1,L2)
1040 00000774 DO 248 J = 1,J1
1041 00000775 WRITE(LP+290) (HEAD(075),L),L = L1,L2),J
1042 00000776 DO 249 I = 1,I1
1043 00000777 243 WRITE(LP+275)(I,(BRAND(I),L),L = 1,3),(SHARE (K,I),J),K = LS,LE)
1044 00000778 WRITE(LP+290)
1045 00000779 WRITE(LP+270) (HEAD(106),L),L = L1,L2)
1046 00000780 DO 401 I = 1,I1
1047 00000781 WRITE(LP+285) I,(BRAND(I),L),L = 1,3),(ASSESK(K,I),J),K = LS,LE)
1048 00000782 401 CONTINUE
1049 00000783 WRITE(LP+270) (HEAD(107),L),L = L1,L2)
1050 00000784 DO 402 I = 1,I1
1051 00000785 WRITE(LP+285) I,(BRAND(I),L),L = 1,3),(SALESK(K,I),J),K = LS,LE)
1052 00000786 402 CONTINUE
1053 00000787 WRITE(LP+270) (HEAD(108),L),L = L1,L2)
1054 00000788 DO 403 I = 1,I1
1055 00000789 WRITE(LP+285) I,(BRAND(I),L),L = 1,3),(GSHARE(K,I),J),K = LS,LE)
1056 00000790 403 CONTINUE
1057 00000791 WRITE(LP+270) (HEAD(109),L),L = L1,L2)
1058 00000792 DO 404 I = 1,I1
1059 00000793 WRITE(LP+280) I,(BRAND(I),L),L = 1,3),(RANK (K,I),J),K = LS,LE)
1060 00000794 404 CONTINUE
1061 00000795 WRITE(LP+290)
1062 00000796 IF (L1.LT.KK) GO TO 240
1063 00000797 244 CONTINUE
1064
1065 00000798 C
1066 7722 CONTINUE
1067
1068 LS = LSM
1069 IF (KK .GT. KW) LS = (LS*(KK-KW)
1070 IF (KW .EQ.0) LS = LSM
1071 245 CONTINUE
1072 LS = LS * KMP
1073 LE = LE * KMP - 1
1074 IF (L1.GT.KK) LE = KK
1075 C
1076 WRITE(LP+200)
1077 WRITE(LP+205)
1078 205 FORMAT(1X,'I11',15(2H '*'),'MARKET SYSTEM',30(2H '+'))
1079 WRITE(LP+290)
1080 WRITE(LP+292) (K,K = LS,LE)
1081 WRITE(LP+210) (HEAD(078),L),L = L1,L2)
1082 WRITE(LP+270) (HEAD(079),L),L = L1,L2)
1083 DO 246 I = 1,I1
1084 246 WRITE(LP+275)(I,(BRAND(I),L),L = 1,3),(AD1 (K,I) ),K = LS,LE)
1085 WRITE(LP+290)
1086 DO 247 I = 1,I1
1087 247 WRITE(LP+275)(I,(BRAND(I),L),L = 1,3),(AD2 (K,I) ),K = LS,LE)
1088 WRITE(LP+270) (HEAD(081),L),L = L1,L2)
1089 DO 248 I = 1,I1
1090 248 WRITE(LP+275)(I,(BRAND(I),L),L = 1,3),(AD3 (K,I) ),K = LS,LE)
1091 WRITE(LP+270) (HEAD(082),L),L = L1,L2)
1092 DO 249 I = 1,I1
1093 249 WRITE(LP+275)(I,(BRAND(I),L),L = 1,3),(AD4 (K,I) ),K = LS,LE)
1094 DO 250 J = 1,J1
1095 WRITE(LP+299) (HEAD(083),L),L = L1,L2),J

```

155

```

1095 00000826 DO 250 I = 1,I1
1096 00000827 250 WRITE(LP+275)(I,(BRAND(I),L),L = 1,3),(SF1 (K,I),J),K = LS,LE)
1097 00000828 WRITE(LP+290)
1098 00000829 WRITE(LP+200)
1099 00000830 WRITE(LP+200)
1100 00000831 WRITE(LP+292) (K,K = LS,LE)
1101 00000832 WRITE(LP+210) (HEAD(084),L),L = L1,L2)
1102 00000833 WRITE(LP+270) (HEAD(085),L),L = L1,L2)
1103 00000834 DO 251 I = 1,I1
1104 251 WRITE(LP+275)(I,(BRAND(I),L),L = 1,3),(EADNAT(K,I) ),K = LS,LE)
1105 00000836 DO 252 J = 1,J1
1106 252 WRITE(LP+299) (HEAD(086),L),L = L1,L2),J
1107 00000838 DO 253 I = 1,I1
1108 253 WRITE(LP+275)(I,(BRAND(I),L),L = 1,3),(ESFORC(K,I),J),K = LS,LE)
1109 00000840 DO 253 J = 1,J1
1110 253 WRITE(LP+299) (HEAD(087),L),L = L1,L2),J
1111 00000842 DO 253 I = 1,I1
1112 253 WRITE(LP+275)(I,(BRAND(I),L),L = 1,3),(EADLOC(K,I),J),K = LS,LE)
1113 00000844 DO 254 I = 1,I1
1114 254 WRITE(LP+270) (HEAD(088),L),L = L1,L2)
1115 00000846 DO 254 J = 1,J1
1116 254 WRITE(LP+275)(I,(BRAND(I),L),L = 1,3),(PRICE (K,I) ),K = LS,LE)
1117 00000848 DO 255 I = 1,I1
1118 255 WRITE(LP+299) (HEAD(089),L),L = L1,L2)
1119 00000850 DO 255 J = 1,J1
1120 255 WRITE(LP+299) (HEAD(090),L),L = L1,L2)
1121 00000852 DO 256 I = 1,I1
1122 256 WRITE(LP+275)(I,(BRAND(I),L),L = 1,3),(TM (K,I) ),K = LS,LE)
1123 00000854 WRITE(LP+200)
1124 00000855 WRITE(LP+290)
1125 00000856 WRITE(LP+292) (K,K = LS,LE)
1126 00000857 WRITE(LP+210) (HEAD(091),L),L = L1,L2)
1127 00000858 WRITE(LP+270) (HEAD(092),L),L = L1,L2)
1128 00000859 DO 257 I = 1,I1
1129 257 WRITE(LP+275)(I,(BRAND(I),L),L = 1,3),(EPANDE(K,I),J),K = LS,LE)
1130 00000861 DO 258 J = 1,J1
1131 258 WRITE(LP+299) (HEAD(093),L),L = L1,L2),J
1132 00000863 DO 258 I = 1,I1
1133 258 WRITE(LP+202)(I,(BRAND(I),L),L = 1,3),(ELOWAL(K,I),J),K = LS,LE)
1134 00000865 DO 259 J = 1,J1
1135 259 WRITE(LP+299) (HEAD(094),L),L = L1,L2),J
1136 00000867 DO 259 I = 1,I1
1137 259 WRITE(LP+282)(I,(BRAND(I),L),L = 1,3),(E (K,I),J),K = LS,LE)
1138 00000869 DO 261 I = 1,I1
1139 261 WRITE(LP+299) (HEAD(095),L),L = L1,L2),J
1140 00000871 DO 261 I = 1,I1
1141 261 WRITE(LP+282)(I,(BRAND(I),L),L = 1,3),(RE (K,I),J),K = LS,LE)
1142 00000873 DO 262 I = 1,I1
1143 262 WRITE(LP+299) (HEAD(096),L),L = L1,L2),J
1144 00000875 DO 262 I = 1,I1
1145 262 WRITE(LP+282)(I,(BRAND(I),L),L = 1,3),(FCFRET(K,I),J),K = LS,LE)
1146 00000877 WRITE(LP+284) (HEAD(097),L),L = L1,L2),(ECRET (K,I) ),K = LS,LE)
1147 00000878 J = 1
1148 00000879 WRITE(LP+288) J
1149 00000880 WRITE(LP+284) (HEAD(097),L),L = L1,L2),(ECRET (K,2) ),K = LS,LE)
1150 00000881 J = 2
1151 00000882 WRITE(LP+288) J
1152 00000883 WRITE(LP+290)

```

XF0RTRAN (V05-00)	MAIN	SOURCE LISTING	78-01-17	13:40:47	PAGE 21
1153	00000884	WRITE(LP+200)			00011450
1154	00000885	WRITE(LP+200)			00011460
1155	00000886	WRITE(LP+292) (K,K = LS+LE)			00011470
1156	00000887	WRITE(LP+201) (HEAD(098+L),L = L1+L2)			00011480
1157	00000888	WRITE(LP+200) (HEAD(099+L),L = L1+L2)+(BDEM (K-1))K = LS+LE)			00011490
1158	00000889	J = 1			00011500
1159	00000890	WRITE(LP+280) J			00011510
1160	00000891	WRITE(LP+260) (HEAD(099+L),L = L1+L2)+(BDEM (K-2))K = LS+LE)			00011520
1161	00000892	J = 2			00011530
1162	00000893	WRITE(LP+280) J			00011540
1163	00000894	WRITE(LP+260) (HEAD(100+L),L = L1+L2)+(AVLM (K-1))K = LS+LE)			00011550
1164	00000895	J = 1			00011560
1165	00000896	WRITE(LP+280) J			00011570
1166	00000897	WRITE(LP+260) (HEAD(100+L),L = L1+L2)+(AVLM (K-2))K = LS+LE)			00011580
1167	00000898	J = 2			00011590
1168	00000899	WRITE(LP+280) J			00011600
1169	00000900	WRITE(LP+260) (HEAD(101+L),L = L1+L2)+(AVDM (K-1))K = LS+LE)			00011610
1170	00000901	J = 1			00011620
1171	00000902	WRITE(LP+280) J			00011630
1172	00000903	WRITE(LP+260) (HEAD(101+L),L = L1+L2)+(AVDM (K-2))K = LS+LE)			00011640
1173	00000904	J = 2			00011650
1174	00000905	WRITE(LP+280) J			00011660
1175	00000906	WRITE(LP+298) (HEAD(102+L),L = L1+L2)			00011670
1176	00000907	DD 263 J = 1+J			00011680
1177	00000908	WRITE(LP+299) (HEAD(103+L),L = L1+L2)+J			00011690
1178	00000909	DD 263 I = 1+I			00011700
1179	00000910	203 WRITE(LP+275) (I+(BRAND(I),L = 1+3)+(BSLO (K+1),J)K = LS+LE)			00011710
1180	00000911	DD 264 J = 1+J			00011720
1181	00000912	WRITE(LP+299) (HEAD(104+L),L = L1+L2)+J			00011730
1182	00000913	DD 264 I = 1+I			00011740
1183	00000914	264 WRITE(LP+275) (I+(BRAND(I),L = 1+3)+(SLQ (K+1),J)K = LS+LE)			00011750
1184	00000915	WRITE(LP+298) (HEAD(105+L),L = L1+L2)			00011760
1185	00000916	DD 265 J = 1+J			00011770
1186	00000917	WRITE(LP+299) (HEAD(106+L),L = L1+L2)+J			00011780
1187	00000918	DD 265 I = 1+I			00011790
1188	00000919	285 WRITE(LP+275) (I+(BRAND(I),L = 1+3)+(BMJW (K+1),J)K = LS+LE)			00011800
1189	00000920	DD 266 J = 1+J			00011810
1190	00000921	WRITE(LP+299) (HEAD(107+L),L = L1+L2)+J			00011820
1191	00000922	DD 266 I = 1+I			00011830
1192	00000923	266 WRITE(LP+275) (I+(BRAND(I),L = 1+3)+(EMJW (K+1),J)K = LS+LE)			00011840
1193	00000924	WRITE(LP+298) (HEAD(108+L),L = L1+L2)			00011850
1194	00000925	DD 267 J = 1+J			00011860
1195	00000926	WRITE(LP+299) (HEAD(109+L),L = L1+L2)+J			00011870
1196	00000927	DD 267 I = 1+I			00011880
1197	00000928	267 WRITE(LP+275) (I+(BRAND(I),L = 1+3)+(SQ (K+1),J)K = LS+LE)			00011890
1198	00000929	DD 268 J = 1+J			00011900
1199	00000930	WRITE(LP+299) (HEAD(110+L),L = L1+L2)+J			00011910
1200	00000931	DD 268 I = 1+I			00011920
1201	00000932	268 WRITE(LP+275) (I+(BRAND(I),L = 1+3)+(S (K+1),J)K = LS+LE)			00011930
1202	00000933	DD 269 J = 1+J			00011940
1203	00000934	WRITE(LP+299) (HEAD(111+L),L = L1+L2)+J			00011950
1204	00000935	DD 269 I = 1+I			00011960
1205	00000936	269 WRITE(LP+275) (I+(BRAND(I),L = 1+3)+(SHARE (K+1),J)K = LS+LE)			00011970
1206	00000937	WRITE(LP+270) (HEAD(166+L),L = L1+L2)			00011980
1207	00000938	DD 411 I = 1+I			00011990
1208	00000939	WRITE(LP+285) I, (BRAND(I),L = 1+3)+(ASSESS(K+1),K = LS+LE)			00012000
1209	00000940	411 CONTINUE			00012010
1210	00000941	WRITE(LP+270) (HEAD(167+L),L = L1+L2)			00012020

XF0RTRAN (V05-00)	MAIN	SOURCE LISTING	78-01-17	13:40:47	PAGE 22
1211	00000942	DD 412 I = 1+I			00012030
1212	00000943	WRITE(LP+285) I, (BRAND(I),L = 1+3)+(SALES(K+1),K = LS+LE)			00012040
1213	00000944	412 CONTINUE			00012050
1214	00000945	WRITE(LP+270) (HEAD(168+L),L = L1+L2)			00012060
1215	00000946	DD 413 I = 1+I			00012070
1216	00000947	WRITE(LP+285) I, (BRAND(I),L = 1+3)+(GSHARE(K+1),K = LS+LE)			00012080
1217	00000948	413 CONTINUE			00012090
1218	00000949	WRITE(LP+270) (HEAD(169+L),L = L1+L2)			00012100
1219	00000950	DD 414 I = 1+I			00012110
1220	00000951	WRITE(LP+280) I, (BRAND(I),L = 1+3)+(RANK (K+1),K = LS+LE)			00012120
1221	00000952	414 CONTINUE			00012130
1222	00000953	WRITE(LP+3285)			00012140
1223	00000954	3285 FORMAT '10%, ' FLUCTUATION OF PROFIT RATE '			00012150
1224	00000955	DD 7285 I = 1, 1			00012160
1225	00000956	WRITE(LP+285) I, (BRAND(I),L = 1+3)+(BANALS(K+1),K = LS+LE)			00012170
1226	00000957	7285 CONTINUE			00012180
1227	00000958	WRITE(LP+290)			00012190
1228	00000959	IF (LE.LT.KK) GO TO 265			00012200
1229					00012210
1230	00000960	C KWI = KW * 1			00012220
1231	00000961	KSTART = 1			00012230
1232	00000962	IF (KK.GT. KWI) KSTART = KK-KW			00012240
1233	00000963	IF (KW.EQ. 0) KSTART = 1			00012250
1234					00012260
1235	00000964	C DD 273 K = KSTART + KK			00012270
1236					00012280
1237	00000965	C LS = LSM			00012290
1238	00000966	271 CONTINUE			00012300
1239	00000967	LS = LS + KWP			00012310
1240	00000968	LE = LS + KWP - 1			00012320
1241	00000969	IF (LE.GT.11) LE = 11			00012330
1242	00000970	WRITE(LP+200)			00012340
1243	00000971	WRITE(LP+202)			00012350
1244	00000972	WRITE(LP+213) K			00012360
1245	00000973	213 FORMAT('H ',11,'PERIOD ',14)			00012370
1246	00000974	K1 = K+1			00012380
1247	00000975	WRITE(LP+290)			00012390
1248	00000976	WRITE(LP+294) (I-1 = LS+LE)			00012400
1249	00000977	WRITE(LP+210) (HEAD(073+L),L = L1+L2)			00012410
1250	00000978	WRITE(LP+260) (HEAD(024+L),L = L1+L2)+(PQ (K+1))I = LS+LE)			00012420
1251	00000979	WRITE(LP+270) (HEAD(025+L),L = L1+L2)+(PRICE (K+1))I = LS+LE)			00012430
1252	00000980	WRITE(LP+260) (HEAD(026+L),L = L1+L2)+(ADLOC (K+1),I)I = LS+LE)			00012440
1253	00000981	WRITE(LP+260) (HEAD(027+L),L = L1+L2)+(ADLOC (K+1),I)I = LS+LE)			00012450
1254	00000982	WRITE(LP+260) (HEAD(028+L),L = L1+L2)+(SFORCE(K+1),I)I = LS+LE)			00012460
1255	00000983	WRITE(LP+260) (HEAD(029+L),L = L1+L2)+(SFORCE(K+1),I)I = LS+LE)			00012470
1256	00000984	WRITE(LP+260) (HEAD(030+L),L = L1+L2)+(ADMAT (K+1))I = LS+LE)			00012480
1257	00000985	WRITE(LP+260) (HEAD(031+L),L = L1+L2)+(TRAN (K+1),I)I = LS+LE)			00012490
1258	00000986	WRITE(LP+260) (HEAD(032+L),L = L1+L2)+(TRAN (K+1),I)I = LS+LE)			00012500
1259	00000987	WRITE(LP+260) (HEAD(033+L),L = L1+L2)+(TM (K+1))I = LS+LE)			00012510
1260	00000988	WRITE(LP+260) (HEAD(034+L),L = L1+L2)+(FUND (K+1))I = LS+LE)			00012520
1261	00000989	WRITE(LP+260) (HEAD(035+L),L = L1+L2)+(FUND (K+1))I = LS+LE)			00012530
1262	00000990	WRITE(LP+210) (HEAD(046+L),L = L1+L2)			00012540
1263	00000991	WRITE(LP+260) (HEAD(049+L),L = L1+L2)+(TSALE (K+1))I = LS+LE)			00012550
1264	00000992	WRITE(LP+260) (HEAD(050+L),L = L1+L2)+(TSALE (K+1),I)I = LS+LE)			00012560
1265	00000993	WRITE(LP+260) (HEAD(051+L),L = L1+L2)+(SO (K+1),I)I = LS+LE)			00012570
1266	00000994	WRITE(LP+260) (HEAD(052+L),L = L1+L2)+(ST (K+1))I = LS+LE)			00012580
1267	00000995	WRITE(LP+260) (HEAD(053+L),L = L1+L2)+(S (K+1),I)I = LS+LE)			00012590
1268	00000996	WRITE(LP+260) (HEAD(054+L),L = L1+L2)+(S (K+1),I)I = LS+LE)			00012600

```

1260 00000907 WRITE(LP210) (HEAD055)L=L L1+L2) 00012610
1270 00000908 WRITE(LP200) (HEAD056)L=L L1+L2)+(FCAP (K,I) )I = LS+LE) 00012620
1271 00000909 WRITE(LP260) (HEAD057)L=L L1+L2)+(PPAST(K1,I) )I = LS+LE) 00012630
1272 00001000 WRITE(LP260) (HEAD058)L=L L1+L2)+(FTSAL(K,I) )I = LS+LE) 00012640
1273 00001001 WRITE(LP260) (HEAD059)L=L L1+L2)+(FASAL(K,I) )I = LS+LE) 00012650
1274 00001002 WRITE(LP260) (HEAD060)L=L L1+L2)+(RS (K,I) )I = LS+LE) 00012660
1275 00001003 WRITE(LP260) (HEAD061)L=L L1+L2)+(FTINH (K,I) )I = LS+LE) 00012670
1276 00001004 WRITE(LP260) (HEAD062)L=L L1+L2)+(FFINH (K,I) )I = LS+LE) 00012680
1277 00001005 WRITE(LP260) (HEAD063)L=L L1+L2)+(FMINH (K,I) )I = LS+LE) 00012690
1278 00001006 WRITE(LP260) (HEAD064)L=L L1+L2)+(FMINH (K,I) )I = LS+LE) 00012700
1279 00001007 WRITE(LP260) (HEAD065)L=L L1+L2)+(FSLB (K,I) )I = LS+LE) 00012710
1280 00001008 WRITE(LP260) (HEAD066)L=L L1+L2)+(FSLB (K,I) )I = LS+LE) 00012720
1281 00001009 WRITE(LP210) (HEAD068)L=L L1+L2) 00012730
1282 00001010 WRITE(LP260) (HEAD067)L=L L1+L2)+(UCOST (K,I) )I = LS+LE) 00012740
1283 00001011 WRITE(LP260) (HEAD068)L=L L1+L2)+(RCOST(K,I) )I = LS+LE) 00012750
1284 00001012 WRITE(LP260) (HEAD069)L=L L1+L2)+(FIXPC (K,I) )I = LS+LE) 00012760
1285 00001013 WRITE(LP260) (HEAD070)L=L L1+L2)+(SCOST (K,I) )I = LS+LE) 00012770
1286 00001014 WRITE(LP260) (HEAD071)L=L L1+L2)+(FIXMC (K,I) )I = LS+LE) 00012780
1287 00001015 WRITE(LP260) (HEAD072)L=L L1+L2)+(STCOST(K,I) )I = LS+LE) 00012790
1288 00001016 WRITE(LP260) (HEAD073)L=L L1+L2)+(TMC (K,I) )I = LS+LE) 00012800
1289 00001017 WRITE(LP260) (HEAD074)L=L L1+L2)+(RENT (K,I) )I = LS+LE) 00012810
1290 00001018 WRITE(LP260) (HEAD075)L=L L1+L2)+(FIXGM (K,I) )I = LS+LE) 00012820
1291 00001019 WRITE(LP260) (HEAD076)L=L L1+L2)+(RANOC (K,I) )I = LS+LE) 00012830
1292 00001020 WRITE(LP260) (HEAD077)L=L L1+L2)+(FCOST (K,I) )I = LS+LE) 00012840
1293 00001021 WRITE(LP260) 00012850
1294 00001022 IF(LE.LT.1) GO TO 271 00012860
1295 00001023 C WRITE(LP200) 00012870
1296 00001024 C LS = LSM 00012880
1297 00001025 C 272 CONTINUE 00012890
1298 00001026 C LS = LS + KWP 00012900
1299 00001027 C LE = LS + KWP - 1 00012910
1300 00001028 C IF(LE.GT.1) LE = II 00012920
1301 00001029 C 00012930
1302 00001030 C WRITE(LP200) 00012940
1303 00001031 C WRITE(LP203) 00012950
1304 00001032 C WRITE(LP213) K 00012960
1305 00001033 C WRITE(LP200) 00012970
1306 00001034 C WRITE(LP294) (I,I = LS+LE) 00012980
1307 00001035 C 00012990
1308 00001036 C WRITE(LP210) (HEAD115)L=L L1+L2) 00013000
1309 00001037 C WRITE(LP260) (HEAD116)L=L L1+L2)+(FASSET(K,I) )I = LS+LE) 00013010
1310 00001038 C WRITE(LP260) (HEAD117)L=L L1+L2)+(CASSET(K,I) )I = LS+LE) 00013020
1311 00001039 C WRITE(LP260) (HEAD118)L=L L1+L2)+(DASSET(K,I) )I = LS+LE) 00013030
1312 00001040 C WRITE(LP260) (HEAD119)L=L L1+L2)+(FCASH (K,I) )I = LS+LE) 00013040
1313 00001041 C WRITE(LP260) (HEAD120)L=L L1+L2)+(FRCCE(K,I) )I = LS+LE) 00013050
1314 00001042 C WRITE(LP260) (HEAD121)L=L L1+L2)+(FFUND (K,I) )I = LS+LE) 00013060
1315 00001043 C WRITE(LP260) (HEAD122)L=L L1+L2)+(FBDOS(K,I) )I = LS+LE) 00013070
1316 00001044 C WRITE(LP260) (HEAD123)L=L L1+L2)+(FRCST (K,I) )I = LS+LE) 00013080
1317 00001045 C WRITE(LP260) (HEAD124)L=L L1+L2)+(LICAP(K,I) )I = LS+LE) 00013090
1318 00001046 C WRITE(LP260) (HEAD125)L=L L1+L2)+(TLIABT(K,I) )I = LS+LE) 00013100
1319 00001047 C WRITE(LP260) (HEAD126)L=L L1+L2)+(CLIABT(K,I) )I = LS+LE) 00013110
1320 00001048 C WRITE(LP260) (HEAD127)L=L L1+L2)+(FPAYBL(K,I) )I = LS+LE) 00013120
1321 00001049 C WRITE(LP260) (HEAD128)L=L L1+L2)+(FDEBY (K,I) )I = LS+LE) 00013130
1322 00001050 C WRITE(LP260) (HEAD129)L=L L1+L2)+(FFTLX(K,I) )I = LS+LE) 00013140
1323 00001051 C WRITE(LP260) (HEAD130)L=L L1+L2)+(RESERV (K,I) )I = LS+LE) 00013150
1324 00001052 C WRITE(LP260) (HEAD131)L=L L1+L2)+(TCAPTL(K,I) )I = LS+LE) 00013160
1325 00001053 C WRITE(LP260) (HEAD132)L=L L1+L2)+(CAPITL (K,I) )I = LS+LE) 00013170
1326 00001054 C WRITE(LP260) (HEAD133)L=L L1+L2)+(PROF (K,I) )I = LS+LE) 00013180

```

```

1327 00001052 WRITE(LP260) (HEAD134)L=L L1+L2)+(PROF1(K,I) )I = LS+LE) 00013190
1328 00001053 WRITE(LP290) 00013200
1329 00001054 WRITE(LP298) (HEAD135)L=L L1+L2) 00013210
1330 00001055 WRITE(LP260) (HEAD136)L=L L1+L2)+(ST (K,I) )I = LS+LE) 00013220
1331 00001056 WRITE(LP260) (HEAD137)L=L L1+L2)+(COST (K,I) )I = LS+LE) 00013230
1332 00001057 WRITE(LP260) (HEAD138)L=L L1+L2)+(TMGC (K,I) )I = LS+LE) 00013240
1333 00001058 WRITE(LP260) (HEAD139)L=L L1+L2)+(TCOST (K,I) )I = LS+LE) 00013250
1334 00001059 WRITE(LP260) (HEAD140)L=L L1+L2)+(PROFIT(K,I) )I = LS+LE) 00013260
1335 00001060 WRITE(LP260) (HEAD141)L=L L1+L2)+(SFUND (K,I) )I = LS+LE) 00013270
1336 00001061 WRITE(LP260) (HEAD142)L=L L1+L2)+(RENT (K,I) )I = LS+LE) 00013280
1337 00001062 WRITE(LP260) (HEAD143)L=L L1+L2)+(PROFIT(K,I) )I = LS+LE) 00013290
1338 00001063 WRITE(LP260) (HEAD144)L=L L1+L2)+(TAX (K,I) )I = LS+LE) 00013300
1339 00001064 WRITE(LP260) (HEAD145)L=L L1+L2)+(PROF1(K,I) )I = LS+LE) 00013310
1340 00001065 WRITE(LP290) 00013320
1341 00001066 WRITE(LP298) (HEAD146)L=L L1+L2) 00013330
1342 00001067 WRITE(LP260) (HEAD147)L=L L1+L2)+(BANALS(K,I) )I = LS+LE) 00013340
1343 00001068 WRITE(LP260) (HEAD148)L=L L1+L2)+(BANALS(K,I) )I = LS+LE) 00013350
1344 00001069 WRITE(LP260) (HEAD149)L=L L1+L2)+(BANALS(K,I) )I = LS+LE) 00013360
1345 00001070 WRITE(LP260) (HEAD150)L=L L1+L2)+(BANALS(K,I) )I = LS+LE) 00013370
1346 00001071 WRITE(LP260) (HEAD151)L=L L1+L2)+(BANALS(K,I) )I = LS+LE) 00013380
1347 00001072 WRITE(LP260) (HEAD152)L=L L1+L2)+(BANALS(K,I) )I = LS+LE) 00013390
1348 00001073 WRITE(LP260) (HEAD153)L=L L1+L2)+(BANALS(K,I) )I = LS+LE) 00013400
1349 00001074 WRITE(LP260) (HEAD154)L=L L1+L2)+(BANALS(K,I) )I = LS+LE) 00013410
1350 00001075 WRITE(LP298) (HEAD155)L=L L1+L2) 00013420
1351 00001076 WRITE(LP260) (HEAD156)L=L L1+L2)+(BANALS(K,I) )I = LS+LE) 00013430
1352 00001077 WRITE(LP260) (HEAD157)L=L L1+L2)+(BANALS(K,I) )I = LS+LE) 00013440
1353 00001078 WRITE(LP260) (HEAD158)L=L L1+L2)+(BANALS(K,I) )I = LS+LE) 00013450
1354 00001079 WRITE(LP260) (HEAD159)L=L L1+L2)+(BANALS(K,I) )I = LS+LE) 00013460
1355 00001080 WRITE(LP260) (HEAD160)L=L L1+L2)+(BANALS(K,I) )I = LS+LE) 00013470
1356 00001081 WRITE(LP260) (HEAD161)L=L L1+L2)+(BANALS(K,I) )I = LS+LE) 00013480
1357 00001082 WRITE(LP260) (HEAD162)L=L L1+L2)+(BANALS(K,I) )I = LS+LE) 00013490
1358 00001083 WRITE(LP298) (HEAD163)L=L L1+L2) 00013500
1359 00001084 WRITE(LP260) (HEAD164)L=L L1+L2)+(BANALS(K,I) )I = LS+LE) 00013510
1360 00001085 WRITE(LP260) (HEAD165)L=L L1+L2)+(BANALS(K,I) )I = LS+LE) 00013520
1361 00001086 WRITE(LP290) 00013530
1362 00001087 C IF(LE.LT.1) GO TO 272 00013540
1363 00001088 C 00013550
1364 00001089 C 273 CONTINUE 00013560
1365 00001090 C 206 FORMAT(M) 00013570
1366 00001091 C 210 FORMAT(1M ,T11,'',10('(',')')',1M ,T11,'',T50,'',T12,'', 00013580
1367 00001092 C * JM ,T12,'',T11,40A1) 00013590
1368 00001093 C 270 FORMAT(1M ,T12,'',T11,40A1) 00013600
1369 00001094 C 00013610
1370 00001095 C 00013620
1371 00001096 C 260 FORMAT(1M ,T12a,'',T11,40A1,5F12,1) 00013630
1372 00001097 C 275 FORMAT(1M ,T11,'',T11,1) ,T11,1) ,T31,1a,24a,42,751,5F12,1) 00013640
1373 00001098 C 242 FORMAT(1M ,T12a,'',T50,'',T11,1) ,T31,1a,24a,42,751,5F12,1) 00013650
1374 00001099 C 244 FORMAT(1M ,T12a,'',T11,40A1,5F12,3) 00013660
1375 00001100 C 265 FORMAT(1M ,T12a,'',T50,'',T11,1) ,T31,1a,24a,42,751,5F12,2) 00013670
1376 00001101 C 240 FORMAT(1M ,T12a,'',T50,'',T11,1) ,T31,1a,24a,42,751,5112) 00013680
1377 00001102 C 240 FORMAT(1M ,T11,'',10('(',')')',1) 00013690
1378 00001103 C 292 FORMAT(1M ,T11,'',T40,'PERIOD ' ,4(' ' ,T12,1) , 00013700
1379 00001104 C T5,1)12) 00013710
1380 00001105 C 294 FORMAT(1M ,T11,'',T40,'COMPANY ' ,4(' ' ,T12,1) , 00013720
1381 00001106 C T5,1)12) 00013730
1382 00001107 C 296 FORMAT(1M ,T12a,'',T11,40A1) 00013740
1383 00001108 C 244 FORMAT(1M ,T12a,'',T11,40A1,T40,'MKT,1) 00013750
1384 00001109 C 00013760

```



```

XFORTRAN (V05-00)      MAIN                SOURCE LISTING      78-01-17  13:40:47      PAGE 25
1305      00001103      GO TO 1000
1306
1307      C
1308      00001104      100 FORMAT ( 3I5 )
1309      00001105      101 FORMAT ( 8I 2A4, A2 )
1390      00001106      110 FORMAT ( 3I, F7.1, 5F10.0 )
1391      00001107      111 FORMAT ( 1X, 2I2+F5.0+F5.1+7F5.0+F5.1+2F5.0+5X+10F1.0 )
1392
1393      00001108      C
1394      1'  TH RAD FUND*' MR= 1 2 3 4 5 6 7'
1395      00001109      305 FORMAT ( 1X, 2I2+F5.0+F5.1+7F5.0+F5.1+2F5.0+5X+10F2.0 )
1396      00001110      310 FORMAT ( 1I, // 3X, 95('67'), 'MARKETING GAME SYSTEM', 35('81'))
1397      00001111      8808 STOP
1398      00001112      END
00013770
00013780
00013790
00013800
00013810
00013820
00013830
00013840
00013850
00013860
00013870
00013880
00013890
00013900
    
```

〔付録Ⅳ〕 実行結果

***** INFORMATION OF BUSINESS BEHAVIOUR *****

COMPANY NO. & BRAND	3 C							
MEMBER	C-1	C-2	C-3	C-4	C-5	C-6	C-7	C-8
	PERIOD		8	9	10	11	12	
((DECISION MAKING))								
PRODUCTION			198.0	257.6	270.0	240.0	210.0	
PRICE			36.0	37.0	38.0	40.0	39.0	
LOCAL ADVERTISEMENT MARKET(1)			62.0	84.0	80.0	92.0	90.0	
SALES FORCE MARKET(2)			44.0	60.0	64.0	68.0	66.0	
MARKET(1)			75.0	96.0	92.0	88.0	84.0	
MARKET(2)			50.0	76.0	74.0	70.0	65.0	
NATIONAL ADVERTISEMENT			105.0	152.0	148.0	140.0	130.0	
TRANSPORTATION MARKET(1)			116.1	125.6	130.8	156.3	136.0	
MARKET(2)			77.4	72.4	98.8	113.7	104.0	
TRADE MARGIN			1.7	2.4	2.6	3.0	2.8	
RESEARCH & DEVELOPMENT			148.0	150.0	152.0	153.0	160.0	
FREQ APPLICATION			0.0	0.0	0.0	0.0	0.0	
((SALES QUANTITY & VALUE))								
SALES QUANTITY TOTAL			157.1	207.4	235.7	240.9	205.5	
MARKET(1)			93.2	119.5	135.1	134.8	113.3	
MARKET(2)			63.9	87.9	100.5	106.1	92.2	
SALES VALUE TOTAL			5694773.0	7674224.0	8955946.0	9634473.0	8012693.0	
MARKET(1)			3543546.0	4422003.0	5135535.0	5391847.0	4417828.0	
MARKET(2)			2361227.0	3252221.0	3820411.0	4242626.0	3594865.0	
((COST STRUCTURE))								
(UNIT PRODUCTION COST)			25.3	24.1	24.4	25.5	25.5	
PRODUCTION VARIABLE COST			3722307.0	4867035.0	5723906.0	5231996.0	4451096.0	
PRODUCTION FIXED COST			993799.2	996659.2	1005639.2	1011719.2	1017834.2	
MARKETING ACTIVITY COST			338000.0	468000.0	468000.0	498000.0	435000.0	
MARKETING FIXED COST			124224.9	124934.9	125704.9	126664.9	127224.9	
INVENTORY COST			166438.1	265310.4	248288.8	259330.9	335261.9	
TRADE COST			267890.7	497787.3	612775.6	722585.9	575268.4	
FINANCIAL COST			614731.4	564759.7	551155.9	527346.6	581677.2	
MANAGEMENT FIXED COST			124224.9	124934.9	125704.9	126664.9	127224.9	
RESEARCH & DEVELOPMENT COST			144000.0	150000.0	152000.0	153000.0	160000.0	
TOTAL OPERATING COST			5982753.0	6346824.0	7463818.0	7912612.0	7149245.0	
((FINAL BUSINESS STRUCTURE))								
FINAL CAPACITY OF PRODUCTION			288.0	288.0	288.0	288.0	288.0	
FINAL PAST PERIOD PRODUCTION			198.0	237.6	270.0	240.0	210.0	
FINAL ACCUMULATED TOTAL SALES			1490.4	1657.8	1693.3	2134.4	2339.9	
FINAL ACCUMULATED AV. SALES			161.3	184.2	189.3	194.0	195.0	
FINAL ACCUMULATED PROFIT RATE			9.5	9.0	8.8	9.1	8.4	
FINAL INVENTORY TOTAL			532.7	562.8	597.2	596.3	600.8	
FACTORY			108.0	237.6	270.0	240.0	210.0	
MARKET(1)			172.7	178.9	182.5	204.0	226.8	
MARKET(2)			161.9	146.4	144.8	146.4	144.1	
CARRIED OVER DEMAND MARKET(1)			0.0	0.0	0.0	0.0	0.0	
MARKET(2)			0.0	0.0	0.0	0.0	0.0	

***** INFORMATION OF MARKET RESEARCH *****
 COMPANY NO. & BRAND 3 C

		PERIOD	8	9	10	11	12
((MARKET RESEARCH INFORMATION))							
((PRICE OF EACH COMPANY))							
PRICE OF FIRM							
1	A		44.0	42.0	41.0	40.0	40.0
2	B		39.0	42.0	42.5	43.0	40.0
3	C		36.0	37.0	38.0	40.0	39.0
4	D		37.5	40.5	41.5	44.0	42.0
5	E		40.0	39.0	41.0	41.0	39.0
((AVERAGE MARKETING COST OF INDUSTRY))							
PRICE			39.5	40.3	40.7	41.7	40.1
LOCAL ADVERTISEMENT MARKET(1)			101.2	109.0	118.3	126.4	125.2
MARKET(2)			72.7	77.4	87.1	89.8	87.0
SALES FORCE MARKET(1)			125.4	135.3	141.9	146.2	142.0
MARKET(2)			92.4	99.3	103.6	106.3	103.1
NATIONAL ADVERTISEMENT			147.4	182.9	172.7	176.9	173.4
((DEMAND OF MARKET))							
EFFECTIVE AV. DEMAND OF INDUSTRY(MKT1)			114.3	148.1	172.3	190.6	173.9
EFFECTIVE AV. DEMAND OF INDUSTRY(MKT2)			82.9	105.1	120.0	138.0	130.8
EFFECTIVE DEMAND OF COMPANY MKT1							
1	A		143.6	198.6	245.4	303.0	271.4
2	B		142.0	174.5	211.0	232.5	207.0
3	C		93.2	119.5	135.1	134.8	113.3
4	D		169.8	218.9	265.2	346.4	332.9
5	E		95.8	134.2	156.8	172.8	160.1
EFFECTIVE DEMAND OF COMPANY MKT2							
1	A		104.9	145.7	175.3	216.4	196.7
2	B		112.9	140.6	157.3	171.9	150.0
3	C		63.9	87.9	100.5	106.1	92.2
4	D		119.2	150.9	192.9	254.7	252.5
5	E		70.0	99.3	113.2	135.2	136.4
((MARKET SHARE))							
SHARE OF YOUR SALES QUANTITY MKT1							
1	A		23.1	23.9	20.4	19.2	21.7
2	B		24.1	21.3	21.0	22.8	20.0
3	C		15.8	16.5	17.2	16.2	14.1
4	D		20.5	19.7	21.4	21.1	21.6
5	E		16.3	16.5	20.0	20.7	19.9
SHARE OF YOUR SALES QUANTITY MKT2							
1	A		26.3	18.0	20.4	18.7	20.9
2	B		20.7	22.1	21.0	20.8	20.0
3	C		15.1	18.2	18.7	18.1	15.4
4	D		23.5	21.2	18.9	19.4	20.8
5	E		16.5	20.6	21.0	23.0	22.8
((SHARE OF PROFIT/ASSET RATIO))							
1	A		21.55	22.59	22.43	21.53	21.82
2	B		22.86	23.40	24.87	25.38	25.49
3	C		16.73	15.21	14.56	14.63	14.05
4	D		23.63	23.35	23.29	23.35	23.86
5	E		15.25	15.04	14.84	15.10	14.59
((SHARE OF SALES QUANTITIES))							
1	A		21.44	21.49	21.70	21.37	21.17
2	B		21.14	21.28	21.17	21.27	21.24
3	C		18.61	18.42	18.34	18.17	17.89
4	D		20.39	20.37	20.38	20.58	20.49
5	E		17.93	18.12	18.41	18.61	19.05
((SHARE OF BUSINESS EVALUATION RATIO))							
1	A		21.71	22.19	21.99	21.38	21.51
2	B		21.97	22.49	23.02	23.34	23.44
3	C		17.87	16.45	16.51	16.45	15.97
4	D		22.03	21.86	21.81	21.84	22.15
5	E		16.61	16.61	16.87	17.00	16.88
((MARK OF BUSINESS EVALUATION RATIO))							
1	A		3	2	2	3	3
2	B		2	1	1	1	1
3	C		4	4	3	5	5
4	D		1	1	1	1	2
5	E		5	5	4	4	4

***** MARKET SYSTEM *****

		PERIOD	8	9	10	11	12
((PAST MARKETING EFFORTS))							
NATIONAL ADV. PAST(1)							
1	A		152.1	154.9	160.0	165.0	160.0
2	B		155.1	154.9	173.2	177.1	170.0
3	C		131.0	105.0	152.0	148.0	140.0
4	D		158.1	160.9	191.2	207.1	223.7
5	E		158.1	160.9	179.2	183.1	178.7
PAST(2)							
1	A		152.1	152.1	154.9	160.0	165.0
2	B		130.0	155.1	154.9	173.2	177.1
3	C		101.0	131.0	150.0	152.0	148.0
4	D		110.0	158.1	140.0	191.2	207.1
5	E		149.1	158.1	160.9	174.2	183.1
PAST(3)							
1	A		152.7	152.1	152.1	154.9	160.0
2	B		149.7	130.0	155.1	154.9	173.2
3	C		116.0	101.0	131.0	150.0	152.0
4	D		155.7	110.0	158.1	140.0	191.2
5	E		149.7	149.1	158.1	160.9	179.2
PAST(4)							
1	A		210.7	192.7	192.1	152.1	154.9
2	B		130.0	149.7	130.0	155.1	154.9
3	C		148.0	118.0	161.0	151.0	151.0
4	D		180.7	195.7	110.0	158.1	140.0
5	E		140.0	149.7	149.1	158.1	160.9
SALES FORCE PAST(1) MKT1							
1	A		168.4	182.2	174.1	182.7	184.5
2	B		132.4	140.2	162.1	158.7	148.3
3	C		94.0	75.0	96.0	92.0	88.0
4	D		126.4	120.0	150.1	164.7	164.3
5	E		85.0	85.0	90.0	90.0	95.0
SALES FORCE PAST(1) MKT2							
1	A		132.3	139.0	135.7	131.4	132.5
2	B		111.3	103.0	117.7	110.4	95.0
3	C		62.0	50.0	76.0	74.0	78.0
4	D		90.0	80.0	89.7	119.4	115.8
5	E		65.0	60.0	65.0	78.0	75.0

		PERIOD	8	9	10	11	12
((MARKETING DECISION IN THIS HEREOD))							
NATIONAL	ADVERTISE	1 A	154.9	160.0	165.0	166.0	199.0
		2 B	154.9	173.2	177.1	170.0	180.0
		3 C	109.0	152.0	146.0	146.0	130.0
		4 D	140.0	191.2	207.1	229.7	210.4
		5 E	160.9	179.2	183.1	178.7	177.4
SALES	FORCE	1 A	182.2	174.1	182.7	184.5	176.8
		2 B	140.2	162.1	158.7	148.5	135.0
		3 C	75.0	90.0	92.0	88.0	84.0
		4 D	120.0	150.1	164.7	184.5	184.8
		5 E	85.0	90.0	90.0	95.0	90.0
SALES	FORCE	1 A	139.0	135.7	131.4	132.5	125.3
		2 B	103.0	117.7	115.4	95.0	85.0
		3 C	50.0	76.0	74.0	70.0	65.0
		4 D	80.0	99.7	119.4	135.5	119.3
		5 E	60.0	65.0	70.0	75.0	80.0
LOCAL	ADVERTISE	1 A	124.5	121.6	123.3	128.5	120.0
		2 B	112.5	127.6	137.3	140.5	134.1
		3 C	62.0	84.0	88.0	92.0	90.0
		4 D	106.5	130.6	149.3	170.5	158.1
		5 E	80.0	85.0	100.0	100.0	96.0
LOCAL	ADVERTISE	1 A	70.0	70.0	70.0	80.0	70.0
		2 B	95.1	102.9	108.2	107.9	94.8
		3 C	64.0	60.0	64.0	66.0	66.0
		4 D	74.1	90.9	102.2	119.9	103.8
		5 E	55.0	60.0	70.0	75.0	70.0
PRICE		1 A	44.0	42.0	41.0	40.0	40.0
		2 B	39.0	42.0	42.5	43.0	40.0
		3 C	36.0	37.0	36.0	40.0	39.0
		4 D	37.5	40.5	41.5	44.0	42.0
		5 E	40.0	39.0	40.0	41.0	39.0
RESEARCH & DEVELOPMANT		1 A	240.0	240.0	240.0	240.0	235.0
		2 B	180.0	190.0	190.0	190.0	195.0
		3 C	146.0	150.0	152.0	155.0	160.0
		4 D	235.0	220.0	220.0	220.0	220.0
		5 E	200.0	200.0	205.0	210.0	210.0
TRADE MARGIN		1 A	3.0	3.0	3.0	3.0	2.5
		2 B	2.2	2.3	2.4	2.4	2.3
		3 C	1.7	2.4	2.6	3.0	2.8
		4 D	2.7	2.8	3.0	3.1	2.9
		5 E	2.2	2.4	2.5	2.6	2.6

		PERIOD	8	9	10	11	12
((EFFECTS OF MARKET SYSTEM))							
R & D	EFFECT	1 A	1.659	1.719	1.739	1.725	1.725
		2 B	1.488	1.470	1.434	1.381	1.362
		3 C	1.444	1.357	1.278	1.193	1.151
		4 D	2.213	2.158	2.084	1.992	1.942
		5 E	1.659	1.637	1.605	1.359	1.541
LOYALTY	EFFECT	1 A	1.310	1.203	1.193	1.191	1.222
		2 B	1.016	1.078	1.068	1.056	1.029
		3 C	0.831	0.760	0.760	0.721	0.661
		4 D	1.045	1.169	1.197	1.254	1.324
		5 E	0.761	0.760	0.782	0.778	0.765
LOYALTY	EFFECT	1 A	1.953	1.213	1.195	1.173	1.180
		2 B	1.044	1.143	1.132	1.106	1.047
		3 C	0.786	0.766	0.737	0.727	0.682
		4 D	1.045	1.131	1.150	1.207	1.243
		5 E	0.772	0.767	0.787	0.788	0.790
MARKETING	EFFECT	1 A	0.896	0.539	0.448	0.396	0.449
		2 B	0.930	0.482	0.393	0.317	0.378
		3 C	0.618	0.333	0.257	0.190	0.216
		4 D	1.042	0.958	0.993	0.941	0.957
		5 E	0.626	0.367	0.291	0.238	0.297
MARKETING	EFFECT	1 A	0.698	0.419	0.340	0.301	0.361
		2 B	0.807	0.309	0.319	0.247	0.260
		3 C	0.459	0.259	0.212	0.159	0.180
		4 D	0.791	0.616	0.573	0.545	0.430
		5 E	0.495	0.287	0.233	0.198	0.259
MARKETING RLT. EFFECT		1 A	0.218	0.237	0.238	0.250	0.261
		2 B	0.228	0.212	0.209	0.200	0.188
		3 C	0.150	0.146	0.137	0.120	0.113
		4 D	0.253	0.245	0.262	0.279	0.292
		5 E	0.152	0.161	0.155	0.150	0.156
MARKETING RLT. EFFECT		1 A	0.215	0.235	0.230	0.241	0.229
		2 B	0.248	0.224	0.216	0.198	0.188
		3 C	0.141	0.165	0.143	0.128	0.121
		4 D	0.243	0.234	0.253	0.276	0.288
		5 E	0.152	0.161	0.158	0.158	0.174
FIRM DEMAND CR. EFFECT		1 A	1.058	1.063	1.064	1.063	1.063
		2 B	1.046	1.044	1.041	1.037	1.035
		3 C	1.042	1.035	1.028	1.030	1.016
		4 D	1.093	1.090	1.086	1.080	1.077
		5 E	1.058	1.057	1.054	1.051	1.050
FIRM DEMAND CR. EFFECT		1 A	1.058	1.063	1.064	1.063	1.063
		2 B	1.046	1.044	1.041	1.037	1.035
		3 C	1.042	1.035	1.028	1.030	1.016
		4 D	1.093	1.090	1.086	1.080	1.077
		5 E	1.058	1.057	1.054	1.051	1.050
MARKET DEMAND CREATION EFFECT		1 A	1.018	1.048	1.054	1.054	1.054
		2 B	1.012	1.038	1.054	1.054	1.054
		3 C	1.012	1.038	1.054	1.054	1.054
		4 D	1.012	1.038	1.054	1.054	1.054
		5 E	1.012	1.038	1.054	1.054	1.054

		PERIOD	8	9	10	11	12
((DEMAND OF MARKET))							
CARRIED OVER DEMAND OF MARKET	MKT1		11.6	8.7	17.2	34.9	49.5
LATENT DEMAND OF MARKET	MKT1		9.9	6.9	20.1	28.3	41.6
LATENT DEMAND OF MARKET	MKT1		110.0	140.0	160.0	180.0	160.0
LATENT DEMAND OF MARKET	MKT2		80.0	100.0	110.0	130.0	120.0
EFFECTIVE DEMAND OF MARKET	MKT1		114.3	148.1	172.3	190.6	173.9
EFFECTIVE DEMAND OF MARKET	MKT2		82.9	105.1	120.0	138.6	136.6
((DEMAND OF EACH COMPANY))							
CARRIED OVER DEMAND							
1	A	MKT1	6.4	0.2	2.6	12.1	21.2
2	B		0.0	0.0	1.9	4.8	2.5
3	C		0.0	0.0	0.0	0.0	0.0
4	D		5.3	0.5	12.7	18.0	25.8
5	E		0.0	0.0	0.0	0.0	0.0
CARRIED OVER DEMAND							
1	A	MKT2	6.2	0.0	9.0	9.2	15.3
2	B		0.0	4.0	4.1	5.2	5.1
3	C		0.0	0.0	0.0	0.0	0.0
4	D		3.7	2.9	6.9	13.9	21.2
5	E		0.0	0.0	0.0	0.0	0.0
EFFECTIVE DEMAND							
1	A	MKT1	143.6	198.4	245.4	303.0	271.4
2	B		142.0	174.5	211.0	232.9	203.9
3	C		93.2	119.5	135.1	134.8	113.3
4	D		109.8	210.9	255.2	346.4	313.1
5	E		95.8	134.2	156.8	172.8	160.1
EFFECTIVE DEMAND							
1	A	MKT2	104.9	145.7	175.3	216.4	196.7
2	B		112.9	140.4	157.3	171.9	150.7
3	C		63.9	87.9	100.5	106.1	92.2
4	D		119.2	150.9	162.9	254.7	252.5
5	E		70.0	99.3	113.2	135.2	136.4
((INVENTORY OF EACH COMPANY))							
BEGINNING MKT INVENTORY							
1	A	MKT1	137.1	173.3	160.0	160.0	175.0
2	B		154.6	156.2	188.0	197.1	187.0
3	C		149.8	172.7	178.9	182.9	204.0
4	D		120.5	142.9	168.0	175.9	175.9
5	E		241.4	285.6	319.4	319.6	268.8
BEGINNING MKT INVENTORY							
1	A	MKT2	102.9	86.7	110.0	110.0	125.0
2	B		87.5	106.4	114.1	124.8	124.8
3	C		148.4	161.9	146.4	144.6	152.3
4	D		99.5	102.1	102.0	114.1	124.1
5	E		197.2	227.1	219.8	219.6	202.2
EFFECTIVE MKT INVENTORY							
1	A	MKT1	137.1	173.3	160.0	160.0	175.0
2	B		154.6	154.1	165.2	190.1	181.2
3	C		130.4	158.4	162.8	165.1	182.3
4	D		120.5	142.9	168.0	175.9	175.9
5	E		223.3	267.5	296.3	294.2	281.4
EFFECTIVE MKT INVENTORY							
1	A	MKT2	102.9	86.7	110.0	110.0	125.0
2	B		87.5	106.4	113.0	121.9	119.5
3	C		127.5	148.3	134.7	132.6	139.0
4	D		99.5	102.1	102.0	114.1	124.1
5	E		182.5	212.9	204.9	204.4	195.9
((SALES OF EACH COMPANY))							
SALES QUANTITY							
1	A	MKT1	137.1	173.3	160.0	160.0	175.0
2	B		142.0	154.1	165.2	190.1	181.2
3	C		93.2	119.5	135.1	134.8	113.3
4	D		120.5	142.9	168.0	175.9	175.9
5	E		95.8	134.2	156.8	172.8	160.1
SALES QUANTITY							
1	A	MKT2	102.9	86.7	110.0	110.0	125.0
2	B		87.5	106.4	113.0	121.9	119.5
3	C		63.9	87.9	100.5	106.1	92.2
4	D		99.5	102.1	102.0	114.1	124.1
5	E		70.0	99.3	113.2	135.2	136.4
SALES VALUE							
1	A	MKT1	6034281.0	7279906.0	6559906.0	6399988.0	6999996.0
2	B		5538304.0	6473671.0	7019285.0	8175507.0	7246394.0
3	C		3353546.0	4422003.0	5135935.0	5391847.0	4417828.0
4	D		4517855.0	5768121.0	6971902.0	7738112.0	7306186.0
5	E		3831745.0	5233062.0	6272792.0	7085652.0	6242495.0
SALES VALUE							
1	A	MKT2	4525710.0	3639989.0	4510000.0	4400000.0	5000000.0
2	B		341439.0	4467312.0	4804437.0	5242906.0	4781643.0
3	C		2301227.0	3252221.0	3820413.0	4242628.0	3595085.0
4	D		3732142.0	4134371.0	4237000.0	5021675.0	5213760.0
5	E		260089.0	3672537.0	4528125.0	5541468.0	5319077.0
SALES QUANTITY SHARE							
1	A	MKT1	23.3	23.9	20.4	19.2	21.7
2	B		24.1	21.3	21.0	22.6	22.5
3	C		15.8	16.5	17.2	16.2	14.1
4	D		20.5	19.7	21.4	21.1	21.6
5	E		16.3	18.5	20.0	20.7	19.9
SALES QUANTITY SHARE							
1	A	MKT2	24.3	18.0	20.4	18.7	20.9
2	B		20.7	21.0	21.0	20.8	20.8
3	C		15.1	18.2	18.7	18.1	15.4
4	D		23.5	21.2	18.9	19.4	20.4
5	E		16.5	20.6	21.0	23.0	22.8
SHARE OF PROFIT/ASSET RATIO							
1	A		21.95	22.59	22.43	21.53	21.82
2	B		21.84	23.80	24.87	25.38	25.49
3	C		16.73	15.21	14.56	14.65	14.05
4	D		23.63	23.35	23.29	23.35	23.86
5	E		15.25	15.04	14.84	15.10	14.95
SHARE OF SALES QUANTITIES							
1	A		21.94	21.89	21.70	21.37	21.37
2	B		21.14	21.20	21.17	21.27	21.29
3	C		18.61	18.42	18.34	18.17	17.80
4	D		20.39	20.37	20.38	20.38	20.49
5	E		17.93	18.12	18.41	18.81	19.05
SHARE OF BUSINESS EVALUATION RATIO							
1	A		21.71	22.19	21.99	21.58	21.71
2	B		21.97	22.49	23.02	23.34	23.49
3	C		15.87	14.68	14.5	14.48	15.47
4	D		22.03	21.86	21.81	21.84	22.15
5	E		16.61	16.61	16.67	17.00	16.88
RANK OF BUSINESS EVALUATION RATIO							
1	A		3	3	2	3	3
2	B		2	1	1	1	1
3	C		4	4	4	4	4
4	D		1	2	3	2	2
5	E		5	5	4	4	4

制御シミュレーション・プログラム

—OPCON*—

内田幸夫

1. はじめに

システムの構造・挙動の理論的究明と時系列データのコンピュータによる数理統計的解析を結合させて、システムのモデル・ビルディングがなされたとき、その後の研究の一つに、モデルを対象としたシミュレーション分析がある。シミュレーションにおいては、ある程度の意志決定をとまなうことになり、実験者はモデルに対して意志決定を外生的に与えるか、または部分的なフィードバック機能をモデルに組込んでシミュレートさせる。このとき、できるかぎり内生的に意志決定を行うことが要望される場合がある。1970年代に入って、経営・経済システムの現代制御理論的究明がなされるようになってきたが（代表的なものに経営学において A. Bensoussan, et. al. [1]、経済学において G. C. Chow [2]がある）、最適制御によるフィードバック制御として意志決定を行うことは、一つの大きな研究成果である。

OPCON は最適制御によって意志決定を機械的に行うシミュレーション用プログラムであり、つぎの特色がある。

1 目標と手段の関係を明らかにすれば、フィードバック制御として意志決定がなされる。

2 かなり大きな非線形モデルでもシミュレーション可能である。

モデルと評価基準の設定に的を絞って、OPCON に組込まれている制御技術ならびに利用方法を解説する。

* この研究の一部は昭和52年度文部省科学研究補助金試験研究(2)「経営経済データ・バンクの拡充とデータ・バンク管理システム」（代表者能勢信子教授）によってなされたものである。OPCON (OPTimal CONtrol simulation program).

2. モデルと評価基準

OPCON を利用するためには、シミュレーションの対象となるモデルが、つぎの一階の非線形連立定差方程式で表現されていなければならない。

$$(1) \quad \mathbf{s}(t) = \mathbf{g}(\mathbf{s}(t), \mathbf{s}(t-1), \mathbf{x}(t), \mathbf{z}(t))$$

モデル (1) を状態方程式とよび、各変数はつぎのようにになっている。

- \mathbf{s} NS 次元状態変数ベクトル
- \mathbf{x} NX 次元手段 (制御) 変数ベクトル
- \mathbf{z} NZ 次元与件 (外生) 変数ベクトル

離散型表現の経営・経済モデルは、より一般的には、高階の定差方程式になる。しかし、高階の定差方程式モデルも、次元を大きくすることにより一階の定差方程式で表現可能である。したがって、OPCON は (1) のモデルに限定しているが、高階のラグを含むモデルも十分シミュレーションの対象になりうる。なお高階の定差方程式を (1) の形に変換する手順については、たとえば G. C. Chow [2] の第 6 章に述べられているが、ここでも例を用いて後述する。

モデルのつぎに用意すべきものは、目標変数と評価基準の定義である。目標変数は状態変数の線形結合として、シミュレーションの評価基準は 2 次の損失関数として与えられていなければならない。

$$(2) \quad \mathbf{y}(t) = \mathbf{H} \mathbf{s}(t)$$

$$(3) \quad \phi = \sum_{t=1}^{NT} \| \mathbf{y}(t) - \mathbf{d}(t) \|^2_{\mathbf{W}(t)}$$

$$(\quad = \sum_{t=1}^{NT} (\mathbf{y}(t) - \mathbf{d}(t))' \mathbf{W}(t) (\mathbf{y}(t) - \mathbf{d}(t)))$$

ただし、

- \mathbf{y} NY 次元目標変数ベクトル
- \mathbf{H} NY × NS 行列
- NT 制御シミュレーションの期間
- \mathbf{d} NY 次元目標値ベクトル

W 対角要素が正の $NY \times NY$ 対角行列であり、目標間の重みを表わす

また手段変数の許容領域をつぎの形で限定することができる。

$$\begin{aligned} & \mathbf{x1}(t) \leq \mathbf{x}(t) \\ (4) \quad & \mathbf{x}(t) \leq \mathbf{x2}(t) \\ & \mathbf{x1}(t) \leq \mathbf{x}(t) \leq \mathbf{x2}(t) \end{aligned}$$

ただし、 $\mathbf{x1}$ 、 $\mathbf{x2}$ はそれぞれ手段変数の下限と上限を示し、 NY 個の手段変数の任意の変数に制約を付加できる。

以上が制御シミュレーションに必要なモデル、式等であり、OPCONは(1)、(2)、または(1)(2)(4)の制約下に(3)を最小化する $\mathbf{x}(t)$, $t=1, 2, \dots, NX$ を求める計算を行う。

3. OPCON のアルゴリズム

OPCON を用いて経営・経済システムの分析を行うのであれば、アウトプットされた計算結果の解釈とともに、どのような方法でシミュレーションが行なわれているのか、計算内容を理解しておくことも重要である。計算結果が悪い場合、その原因がモデルと評価基準の設定にあるのか、それともプログラムによるものなのかを判別するためにも、アルゴリズムを知っておく必要がある。しかし、細かい数式展開は専ら G. C. Chow [3] を参考にすることにして、ごく簡単にアルゴリズムを紹介する。

OPCON が採用している制御技術は、試験的経路 (tentative path) において非線形モデルの線形化を行い、その線形近似モデルに対する最適経路 (optimal path) の計算である。変数の上に「 \wedge 」をつけて試験的経路を示し、また変数の右肩に「 \star 」をつけて最適経路を表わす。初期状態、手段変数および与件変数の試験的経路 $\hat{\mathbf{s}}(0)$, $\hat{\mathbf{x}}(t)$, $\hat{\mathbf{z}}(t)$, $t=1, 2, \dots, NT$ は入力装置より読込まれる。そして変形ガウス・ザイデル法によって求められたモデル(1)の解を、状態変数の試験的経路 $\hat{\mathbf{s}}(t)$, $t=1, 2, \dots, NT$ とする。以上のようにして得られた

最初の試験的径路におけるモデル(1)の線形近似モデルはつぎのようになる。

$$(5) \quad \mathbf{s}(t) = \mathbf{A}(t)\mathbf{s}(t-1) + \mathbf{B}(t)\mathbf{x}(t) + \mathbf{u}(t)$$

ただし、

$$\mathbf{A}(t) = \left(\mathbf{I} - \frac{\hat{\partial \mathbf{g}}}{\partial \mathbf{s}(t)} \right)^{-1} \frac{\hat{\partial \mathbf{g}}}{\partial \mathbf{s}(t-1)} \quad \text{NS} \times \text{NS} \text{ 時変行列}$$

$$\mathbf{B}(t) = \left(\mathbf{I} - \frac{\hat{\partial \mathbf{g}}}{\partial \mathbf{s}(t)} \right)^{-1} \frac{\hat{\partial \mathbf{g}}}{\partial \mathbf{x}(t)} \quad \text{NS} \times \text{NX} \text{ 時変行列}$$

$$\mathbf{u}(t) = \hat{\mathbf{s}}(t) - \mathbf{A}(t)\hat{\mathbf{s}}(t-1) - \mathbf{B}(t)\hat{\mathbf{x}}(t) \quad \text{NS 次元線形誤差値ベクトル}$$

(5)と(2)の制約式の下に、(3)を最小化する問題を解くと、つぎのフィードバック制御解が得られる。

$$(6) \quad \mathbf{x}(t) = \mathbf{f}(t) - \mathbf{F}(t)\mathbf{s}(t-1)$$

ただし、フィードバック行列 \mathbf{F} とフィードフォワード・ベクトル \mathbf{f} は、リカッチ行列 \mathbf{R} およびトラッキング・ベクトル \mathbf{r} を用いて、つぎのように時間軸に対して逆方向に計算されていく。

$$\mathbf{F}(t) = (\mathbf{B}(t)' \mathbf{R}(t) \mathbf{B}(t))^{-1} \mathbf{B}(t)' \mathbf{R}(t) \mathbf{A}(t)$$

$$\mathbf{f}(t) = (\mathbf{B}(t)' \mathbf{R}(t) \mathbf{B}(t))^{-1} (\mathbf{r}(t) - \mathbf{R}(t) \mathbf{u}(t))$$

(以上、 $t=1, 2, \dots, \text{NT}$)

$$\mathbf{R}(t) = \mathbf{H}' \mathbf{W}(t) \mathbf{H} + (\mathbf{A}(t+1) - \mathbf{B}(t+1) \mathbf{F}(t+1))' \mathbf{R}(t+1) \mathbf{A}(t+1)$$

$$\mathbf{r}(t) = \mathbf{H}' \mathbf{W}(t) \mathbf{d}(t) + (\mathbf{A}(t+1) - \mathbf{B}(t+1) \mathbf{F}(t+1))' (\mathbf{r}(t+1) - \mathbf{R}(t+1) \mathbf{u}(t+1))$$

(以上、 $t=1, 2, \dots, \text{NT}-1$)

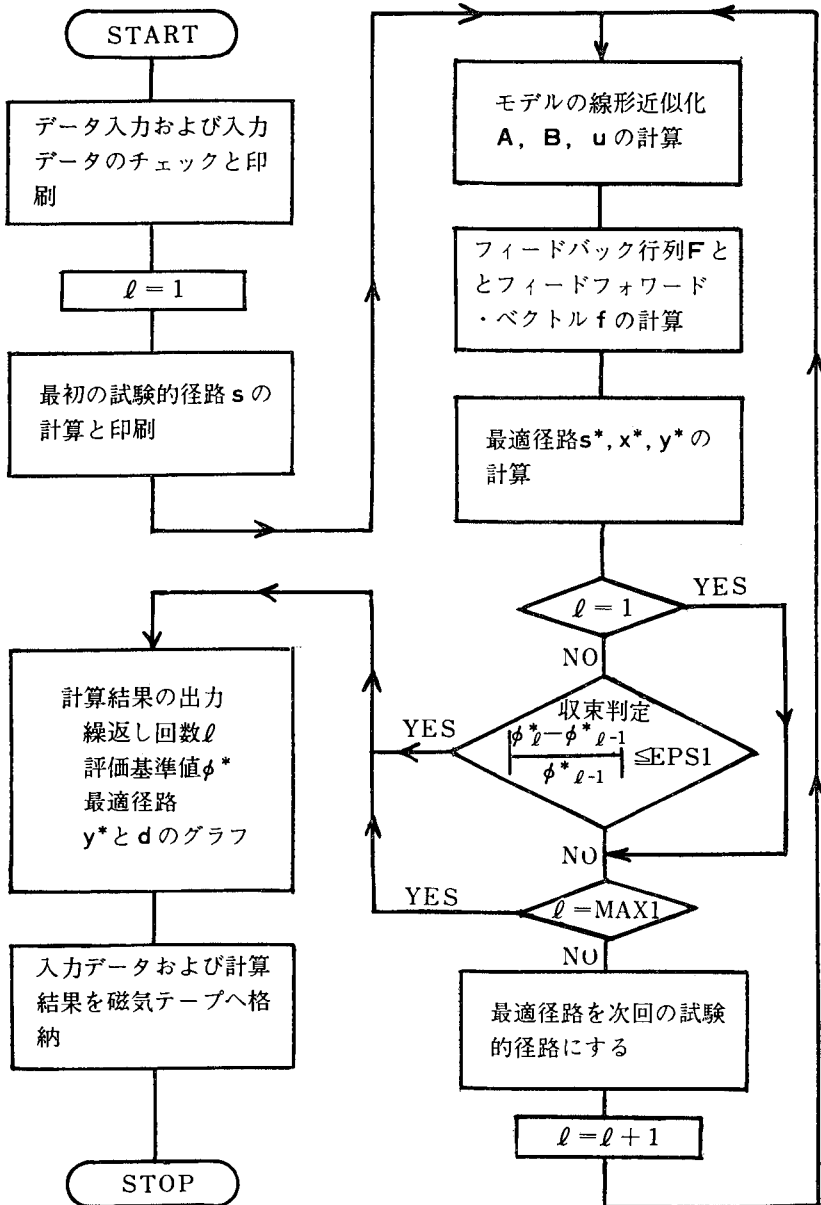
$$\mathbf{R}(\text{NT}) = \mathbf{H}' \mathbf{W}(\text{NT}) \mathbf{H}$$

$$\mathbf{r}(\text{NT}) = \mathbf{H}' \mathbf{W}(\text{NT}) \mathbf{d}(\text{NT})$$

初期状態 $\hat{\mathbf{s}}(0)$ を(6)に代入すると、手段変数の最適値 $\mathbf{x}(1)^*$ が得られる。

$\mathbf{x}(1)^*$ と $\hat{\mathbf{z}}(1)$ の下に変形ガウス・ザイデル法によって得られたモデル(1)の解を状態変数の最適値 $\mathbf{s}(1)^*$ とする。さらに、 $\mathbf{s}(1)^*$ を(6)に代入すると、 $\mathbf{x}(2)^*$ が得られ、 $\mathbf{x}(2)^*$ と $\hat{\mathbf{z}}(2)$ から $\mathbf{s}(2)^*$ が求められる。このように、(1)と(6)を交互に使っ

第1図 フローチャート



て最適経路 $\mathbf{x}(t)^*$, $\mathbf{s}(t)^*$, $t=1, 2, \dots, NT$ が計算される。もし手段変数に(4)の制約があり、しかも(6)によって得られた値が制約条件を侵害しているときは、その条件の境界値を手段変数の最適値とする。

以上のようにして求められた最適経路を、次回のイタレーションにおける試験的経路としてまったく同じ計算ステップを繰返していく。この繰返し計算は損失関数の値が収束するか、または最大繰返し回数 MAX1 まで続行される。そしてつぎのとき、第 ℓ イタレーションで収束したと判定される。

$$(7) \quad \left| \frac{\phi^*_{t} - \phi^*_{t-1}}{\phi^*_{t-1}} \right| \leq \text{EPS1}$$

ただし、

$$\phi^* = \sum_{t=1}^{NT} \|\mathbf{H}\mathbf{s}(t)^* - \mathbf{d}(t)\|_{\mathbf{W}(t)}^2$$

4. MODEL および入出力について

OPCON には以下の規則に合ったサブルーチンMODELが用意されなければならない。サブルーチン MODEL は、モデルの解を求めるとき ($N=1$)、モデルの線形近似を行うとき ($N=2$)、以上の2つの用途に供される。(第6表のC5~C13を参照せよ。)

```
SUBROUTINE MODEL (N, NS, NX, NZ, S, SL,
                   V, X, Z)
```

```
IMPLICIT REAL*8 (A-H, O-Z)
```

```
DIMENSION S(NS), SL(NS), V(NS), X(NX),
           Z(NZ)
```

```
V(1)=.....
```

```
V(2)=.....
```

```
.....
```

```
IF(N.EQ.1) RETURN
```

V(.) =

V(.) =

.....

R E T U R N

E N D

モデル(1)の左辺の $s(t)$ をV, 右辺の $s(t)$ をS, $s(t-1)$ をSL, $x(t)$ をXそして $z(t)$ をZとして, 拡張 FORTRAN でプログラムされる。また右辺にSがある式のみを

I F (N . E Q . 1) R E T U R N

の前に集める。

INPUT により入力装置ならびに重み行列 W の入力様式が区別される。

INPUT = 0 カードのみ, 重み行列 W の要素はすべて1.0

1 カードのみ

2 カードと磁気テープの両方

第1表 入力データ, カード

A 番号	B 入力データ	C番号	D 入力書式
1	計算の題目	15	(10A8)
2	INPUT, MAX1, MAX2, EPS1, EPS2, GS	16	(3I10, 3F10.0)

入力様式の相異とは無関係に, まず最初に, 第1表のデータ・カードが読込まれる。第3列のC欄は第6表のジョブ・カード例1におけるカード番号を示している。MAX2, EPS2, GS はそれぞれ変形ガウス・ザイデル法における最大繰返し回数, 収束基準値, 調整パラメータであり, 調整不要のときはGS=1.0とする。

INPUT=2のとき, 磁気テープには第2表のデータが順番に記録されていなければならない。

各変数のデータは, たとえば, 第*i*手段変数の第*t*期の値はX(*i*, *t*)にあり,

第2表 入力データ，磁気テープ

A 番号	B 入力データ	E 次元
1	NS, NT, NX, NY, NZ	
2	H	(NY, NS)
3	S0 (初期状態)	(NS)
4	X1	(NX, NT)
5	X2	(NX, NT)
6	Z	(NZ, NT)
7	D	(NY, NT)
8	W	(NY, NT)
9	X	(NX, NT)

第3表 入力データ，カード

A 番号	B 入力データ	C 番号	D 入力書式
1	NS, NT, NX, NY, NZ	17	(5 I10.0)
2	変数名, 番号, 入力書式	18	(A4, I6, 5A8)
3	入力書式に合ったデータ	19	
4	A2 と A3 の繰返し使用	20~43	
5	/*	44	

すべて列方向で磁気テープから入力される。

INPUT=0, 1のとき，第3表の方法でカードよりデータを与える。

MAX1, NS等の大きさのチェック機能が組込まれており，それらの許容範囲は INPUT= 0, 1 のときに出力される。ライン・プリンターには第4表のものが出力される。

第4表 出力データ，ライン・プリンター

A 番号	B 出力データ
1	入力データ
2	最初の状態変数の試験的径路 S
3	最後の最適径路 S, X, Y
4	Y と D のグラフ

計算が終了したとき、第5表のデータが磁気テープに記録される。

第5表 出力データ、磁気テープ

A 番号	B 出力データ
1	NS, NT, NX, NY, NZ (入力データ)
2	H (同上)
3	S0 (同上)
4	X1 (同上)
5	X2 (同上)
6	Z (同上)
7	D (同上)
8	W (同上)
9	X (最後の最適経路)
10	Y (同上)
11	S (同上)

A 1～A 8は入力データと同じ、A 9～A11は最後の最適経路である。したがって、A 1～A 9は INPUT=2 として、次回の計算に使用可能である。

5. 使用例

5.1 ジョブ・カードの設定例

つぎの2階の定差方程式モデルにおいて、目標変数 $a(t)$, $b(t)$ および不等式制約 $x_1(t) \leq x(t)$ の下に、20期間のシミュレーションを行うことにする。

$$a(t) = 0.6 \sin(a(t-1) + z_1(t) + x(t)) + 0.8 \cos(b(t) + z_2(t))$$

$$b(t) = 0.7a(t)x(t) - 0.3 a(t-2) b(t-1)$$

$s_1(t) = a(t)$, $s_2(t) = a(t)$, $s_3(t) = a(t-1)$ として、モデルを状態方程式(1)の形で表わす。

$$s_1(t) = 0.6 \sin(s_1(t-1) + z_1(t) + x(t)) + 0.8 \cos(s_2(t) + z_2(t))$$

$$s_2(t) = 0.7 s_1(t)x(t) - 0.3 s_3(t-1) s_2(t-1)$$

$$s_3(t) = s_1(t-1)$$

ゆえにサブルーチン MODEL は第6表のC 5～C13のようになる。カードよ

第6表 ジョブ・カード例1 (INPUT=0)

```

C 1 //KOBOKEN JOB 1234,A,150000,M,,N
C 2 // CATALG OPCON
C 3 // ASSGN DSET17,00
C 4 // EXEC XFORT
C 5 SUBROUTINE MODEL (N,NS,NX,NZ,S,SL,V,X,Z)
C 6 IMPLICIT REAL*8 (A-H,O-Z)
C 7 DIMENSION S(NS),SL(NS),V(NS),X(NX),Z(NZ)
C 8 V(1)=0.6*DSIN(SL(1)+Z(1)+X(1))+0.8*DCOS(S(2)+Z(2))
C 9 V(2)=0.7*S(1)*X(1)-0.3*SL(3)*SL(2)
C 10 IF(N.EQ.1) RETURN
C 11 V(3)=SL(1)
C 12 RETURN
C 13 END
C 14 // CATLG OPCONLNK
C 15 EXAMPL-1
C 16 0 2 30 0.001 0.001 1.0
C 17 3 20 1 2 2
C 18 H 1 (3F10.0)
C 19 1.0 0.0 0.0
C 20 H 2 (3F10.0)
C 21 0.0 1.0 0.0
C 22 S0 (3F10.0)
C 23 0.02 0.46 0.02
C 24 X1 1 (8F10.0)
C 25 1.5 1.5 1.5 1.5 1.5 1.3 1.3 1.3
C 26 1.3 1.3 1.1 1.1 1.1 1.1 1.1 0.9
C 27 0.9 0.9 0.9 0.9
C 28 Z 1 (20F4.0)
C 29 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1.
C 30 Z 2 (8F10.0)
C 31 0.2 0.2 0.2 0.2 0.2 0.2 0.2 0.2
C 32 0.2 0.2 0.2 0.2 0.2 0.2 0.2 0.2
C 33 0.2 0.2 0.2 0.2 0.2
C 34 D 1 (8F10.0)
C 35 0.52 0.54 0.56 0.58 0.60 .62 .64 .66
C 36 .68 .70 .72 .74 .76 .78 .80 .82
C 37 .84 .86 .88 .90
C 38 D 2 (8F10.0)
C 39 0.55 0.55 .54 .54 .53 .53 .52 .52
C 40 .51 .51 .50 .50 .49 .49 .48 .48
C 41 .47 .47 .46 .46
C 42 X 1 (20F4.0)
C 43 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1.
C 44 /*
C 45 // END

```

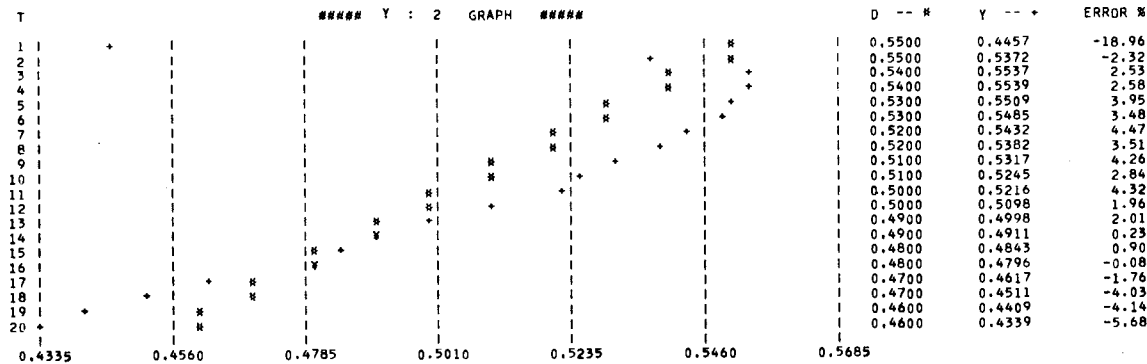
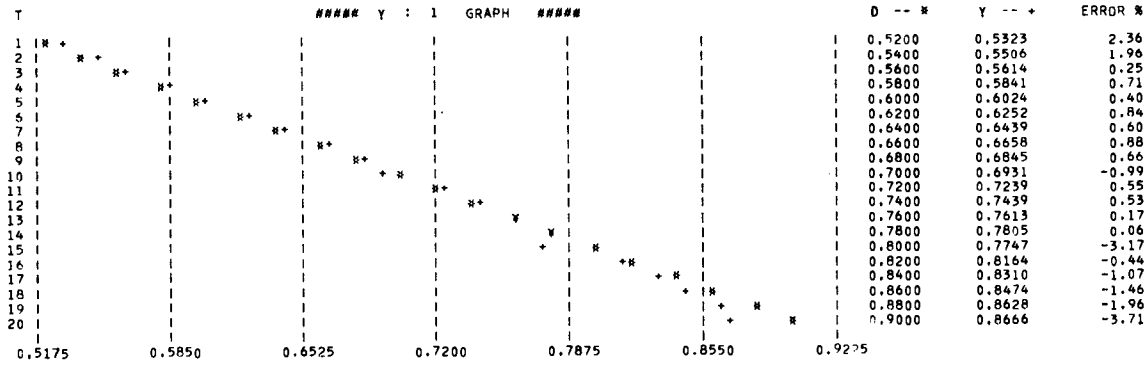
第7表 ジョブ・カード例2 (INPUT=2)

```

C 1 //KOBOKEN JOB 1234,A,150000,M,,N
C 2 // CATALG OPCON
C 3 // ASSGN DSET17,00
C 4 // EXEC XFORT
C 5 SUBROUTINE MODEL (N,NS,NX,NZ,S,SL,V,X,Z)
C 6 IMPLICIT REAL*8 (A-H,O-Z)
C 7 DIMENSION S(NS),SL(NS),V(NS),X(NX),Z(NZ)
C 8 V(1)=0.6*DSIN(SL(1)+Z(1)+X(1))+0.8*DCOS(S(2)+Z(2))
C 9 V(2)=0.7*S(1)*X(1)-0.3*SL(3)*SL(2)
C 10 IF(N.EQ.1) RETURN
C 11 V(3)=SL(1)
C 12 RETURN
C 13 END
C 14 // CATLG OPCONLNK
C 15 EXAMPL-2
C 16 2 2 30 0.001 0.001 1.0
C 17 // END

```

第2図 YとDのグラフ



りデータを与えることにすれば、たとえば第6表のジョブ・カードが用意される。

2回の繰返し計算 (MAX1=2) において、0.1% (EPS1=0.001) 以内に収束しないため、もう2回の繰返し計算を実行するときは、第7表のジョブ・カードを用意する。

5.2 結果とその読み方

計算は4回目に収束したが、YとDのグラフは第2図のようになった。上図は第1番目のそして下図は第2番目の目標変数についてのグラフである。図中の記号において、Y:1, Y:2 はそれぞれ第1, 第2目標変数を表わし、この例では、 $Y:1=Y(1)=S(1)=a$, $Y:2=Y(2)=S(2)=b$ となっている。左端の T はシミュレーションの期を示し、また右端の D は目標値、Y は目標変数の最適値、そして ERROR は Y と D の誤差率を表わし、つぎの式で計算される。

$$\begin{aligned} \text{ERROR} &= 100 \times \frac{Y-D}{|D|} && (D \neq 0 \text{ のとき}) \\ &= Y-D && (D=0 \text{ のとき}) \end{aligned}$$

6. お わ り に

OPCON が使用している制御技術は、非線形制御理論でなく、非線形モデルの線形近似モデルに対する線形制御理論である。その意味において、計算された最適径路は、厳密には、準最適径路というべきであろう。非線形連立モデルに対する実用的な非線形制御理論が完成されていない現状においては、準最適制御理論を用いるしか方途がないと思われる。したがって、少しでも良好な分析結果を得るためには、モデリングに際して、十分な注意が払われなければならない。各種検定の中で、少なくともファイナル・テストに合格したモデルを用いるべきである。OPCON においても、ファイナル・テストに相当する最初の状態変数の試験的径路を計算しているが、この径路が予想径路とかけ離れて

おれば、良い制御結果を望むことは不可能である。幸いにも、モデリングには、STEP FOIL（杉浦〔4〕）等の言語形式のプログラム・システムが開発されているので、それらを有効に利用すべきである。十分なモデル・テストに合格しておれば、OPCON はかなり大きなモデルでも制御シミュレーションが可能であるようにプログラムされているので、今後、経営・経済システムの新しい分析が期待される。

引用文献

- [1] Bensoussan, A., E. G. Hurst, Jr. and B. Näslund, *Management Application of Modern Control Theory*, North-Holland Publishing Co., 1974.
- [2] Chow, G. C., *Analysis and Control of Dynamic Economic Systems*, John Wiley & Sons, 1975.
- [3] Chow, G. C., "An Approach to the Feedback Control of Nonlinear Econometric Systems," *Annals of Economic and Social Measurement*, Vol. 5, pp. 297-309, 1976.
- [4] 杉浦一平, 「構文記述言語——FOIL V-10——」, 経営機械化シリーズ 18 (神戸大), 1978.

構文記述言語

— FOIL V-10 —

杉浦一平

はじめに

この研究叢書の前号（経営機械化シリーズ No.17）に、STEPS FOIL について、その構造を説明した。STEPS FOIL は、中小規模の計算機による一括処理を目的に設計された、高水準言語システムであり、これによって開発したシステムで、その後、大規模な研究を実施し、効率性を実証しえた。

STEPS FOIL は、公開を前提に開発したものである。多くの大学・研究機関からの要望に応じようとして、IBM370、HITAC8000シリーズ系統ならば変換は比較的容易であったが、他機種については、プログラミング仕様上、書替えが極めて困難で、全面的な変更を必要とすることがわかった。加えて、最近になって、大学における計算機設備の大幅な改善の可能性が生じ、記憶装置の大容量化、および特に、時分割処理又は実時間処理の、一般化のきざしが見えはじめた。これらの事情から、STEPS FOIL を、他機種（ACOSシリーズ、および FACOM）の仕様に、変換を行う代りに、大型機用FOILのもつ性能を再検討して、コンパクトなシステムを新しく設計し直すこととした。

FOIL V-10 は、共通領域(COMMON)の大きさをどの程度に設計するかによるが、128 KB 程度で充分実用的なシステムを編成しうる、高水準言語モニター・システムである。現在実稼動中の FOIL ファミリのうち、最大のものは、V-6 (“ASTRO”, IBM 370/168, 三和銀行, 東洋情報システム) であり、最新のものは、V-8 (“CHARM”, ACOS 77, 日本電気) である。両者とも、一括処理、時分割会話処理のいずれをも選択できるが、コンパイラはかなり龐大なものになった。V-10は、もともとV-9として一応設計を完了、プログラミ

ングを開始したが、途中で現在の構想に到達し、それまでの成果を破棄して再出発し、1977年3月に、モニター部およびコンパイラ部の作製に成功した。このシステムはコンパクトであるのみならず、過去の FOIL 開発によって得た成果のほとんどを包摂しえた上に、次のような諸特性をもっている。

1. 構文記述言語をもち、その記法を、“ASTRO”で採用していた1命令文を1行に連続して書く自由型式から、アセンブラ言語的な固定書式(将来は、絶対固定書式を半固定書式に改める)に変更した。この結果、命令文の構文規約を決定するに当たって、より大きな自由度が保証されるようになった。
2. 従来の FOIL は、汎用コンパイラが一括処理を前提に設計されていた。時分割処理時に、会話型の誘導をシステムから与えられながら、命令文を入力する場合には、全然別個の会話ルーティンが機能して、命令文が編集され終ってから、コンパイラに引渡たされる。V-10では、汎用コンパイラ自体が会話機能をもち、時分割時に命令文の一括入力を行っている途中で中断すれば、直ちに会話が始まって誘導を開始し、会話途中でそれ以上の誘導を必要としなくなれば、命令文の終りまで一括入力することによって、会話は中断される。利用者には常に必要且十分な情報が提供されるよう改良された。
3. 構文記述言語は、会話処理用と一括処理用とを区別していない。したがって一括処理専用のシステムを編集したいときに、会話処理用に開発されたソースを、ほとんど変更しないで用いれば、システムが自動的に会話のための情報を無視して編集するし、逆に、一括処理用のソースで、会話処理用のシステムを編集または更改しても、誘導の際システムの与える情報が不十分になるだけで、実用に充分耐える程度の会話誘導が与えられる。
4. 従来の FOIL では、インタプリタティブなシステムの特性から来る弱点として、演算式を多数組合せて、一定の処理を行いたい場合(たとえば、非線型モデルの記述)に、一命令文ごとにファイルのアクセスを行なわねばならず、処理時間がそのために極端に長くなった。この弱点を回避する

ため“ASTRO”では、命令文をブロッキングする命令文をつくったが、長大な場合はカバーし切れなかった。V-10では、マッピングと呼ぶ手法によって、この弱点を克服することに成功し、これをBASIC-サブシステムの標準機能に組込んだ。この改良によって、実用的な規模の非線型計量経済模型のシミュレーションも効率的に実行できるようになった。

FOIL-V10は完成したばかりで、まだ若干の欠点をもっていると考えるが、致命的な欠陥はない。すでに配布済のシステムも、近い将来、これに置替えて、統一をはかりたいと考えている。

このシステムは、数本のサブルーティンを入れ替えることによって、IBM 370, HITAC8000, HITACM-シリーズ、およびACOS77(600以上)に対して提供される。FACOM 系統については、まだ変換を実施した経験をもたないが、重大な困難はない筈と予想している。

なお、現状において、FOIL V-10の機能は、必ずしもこのマニュアルに記述した機能をすべて満たしているとは言えず、多くの欠陥を残していると言わざるを得ない。利用者各位の御指摘を得て、出来るだけ早く、欠陥を除去し、マニュアル通りの機能をもたせたいと希望している。懇切なる叱正を乞いたい。

汎用コンパイラ FOIL V-10

構文記述プログラムと構文解析ルーティン

FOIL V-10は高水準応用言語の翻訳およびタイム・シェアリング時における会話処理を、単一の言語・会話処理ルーティンである“汎用コンパイラ”によって一括処理する。このルーティンは、単一のルーティンで理論的には無制限多数の応用言語の命令文の翻訳を行なうことができる。各命令文の構文規約は表の形でシステム・ファイルに与えられる。この入力のためには、アッセンブラー形式の構文記述言語が準備されてシステムに内蔵されており、これによって組まれた構文記述プログラムと、3, 4のFOIL命令文とを書くだけで新しい命令文のための翻訳ルーティンがシステム内に生成される。

会話ルーティンは、構文記述プログラムを作製すれば殆んど自動的に生成される。必要に応じて詳細説明のためのメッセージを付記するだけでよい。会話のための特殊機能に関する命令も構文記述言語に含めているが、通常はこれらを用いなくてもよい。会話ルーティンは、利用者が十分な情報を与えている間は作動せず、それ以上の処理に情報が不足していたり、または、誤まった情報が入力された場合には、自動的に作動する。作動途中においても会話を継続する必要がないくらい十分な情報が一括して利用者から与えられれば、直ちに作動を中止する。これらの機能によって、かなり複雑な形で指定すべき要素も多い命令文についても、十分に有効で効率的な誘導を行なうことができる性能をもつ。

演算式翻訳ルーティン

演算式（算術および論理）の翻訳および実行は、高水準の応用言語の形成にあたって、工数増大の最大の原因となっていた。たとえば、BMDP は、データの加工を行いたいときは FORTRAN プログラムを利用者側で組んで個々の手法のプログラムに挿入するという便法をとっている。

FOIL は、いわゆる“棚積み法”によらない独自の演算式翻訳ロジックを発見し、このロジックによって汎用的に演算式を翻訳している。このため、単に FORTRAN 的なスカラー演算だけに止まらず、配列間や場合によっては行列間の演算を行う命令文までシステムの必要に応じて組込めるようになった。

FOIL の演算式翻訳ロジックは、高水準応用言語の性質上、システムの構造を INTERPRETATIVE に設計しなければならない場合が殆んどであるという制約に適合するもので、しかも、翻訳効率、実行効率ともに充分満足すべきものである。すでに、ノン・リニア・エコノメトリック・モデルのシミュレーション・言語システムの開発にも適用され、大規模な非線型連立方程式よりなる経済モデルのシミュレーションに使用して、単にプログラミングが従来と比較にもならぬ位簡単化されただけでなく、時間的にも実用に耐える程度のものであることが実証されている。

翻訳ルーティンと実行ルーティンとは独立しているので翻訳結果に対応させ

て、新らしく開発されるシステムの特性に適合した実行ルーティンだけを作製することができ、このためのプログラミングは簡単である。

命令の登録

既存のプログラム・パッケージから1つのプログラムを選び、それを FOIL システムに組込みたいという場合、どのように命令文規約を定め、それを登録して翻訳可能な状態にしてゆくかを説明しよう。

例として BMDP の多変量線型回帰分析 (PIR) について FOIL の命令文規約を考え、この翻訳ルーティンおよび命令文入力誘導用の会話ルーティンをシステムに組込む手順を説明する。

まず、命令文の名称を定める。これを命令文の命令部とよぶ。命令部は3語以内(1語8字以内、8字以上の場合は、後から8字で識別される)で示すものとするが、これを FULL NAME とよび、別に1語の簡略命令を指定することとし、これを SHORT NAME とよぶ。FULL NAME と SHORT NAME が同じであっても差支えない(たとえば、IF, DO, GO 等)。また、利用者はプログラミングするとき、このどちらを用いてもよい。この命令文については

(SHORT NAME) REGRESSION

(FULL NAME) LINEAR REGRESSION

としよう。REGRESSION は後8字 (GRESSION) で識別される。この命令部をはじめにシステムに登録するが、登録ルーティンは命令文としてシステムに組込まれているので、次のような FOIL プログラムを実行させればよい。

```
FOIL START, ' NEW STATEMENT REGISTRATION ' :
LIBRARIAN START, TSBMD :
REGISTER STATEMENT (REGRESSION)
      LINEAR REGRESSION :
      * * * * *
LIBRARIAN END #1 :
FOIL END :
```


これで、命令部の登録と、システム・ライブラリへのプログラム登録位置の割当て、このルーティンのシステム内部名の割当てが行なわれ、この命令を含む既登録命令の一覧表が出力される。*****の位置には、後で説明する構文記述プログラムの翻訳・登録命令文が入る。

命令部の次には、制御番号部が書かれる規約である。この部分は特殊記号#と整数とからなる。制御番号部は、標準的な処理のときは省略することができる、省略されたときは#0が記入されているものとみなされる。

制御番号は命令文によって種々の目的に利用することができ、各命令文ごとと文法規約の一部としてその機能を定める。REGRESSIONの命令文では、中間結果（共分散行列、相関係数行列、等）のリスト制御に用いることとする。

制御番号部の処理ルーティンは、コンパイラに組込まれている。実行時における処理は、インタフェイス・プログラムで行う。

命令文の構文規約

制御番号部の後に、この命令文で行ないたい処理の具体的内容が書かれる。これを、命令文の内容記述部（又は単に記述部）とよぶ。この命令文の場合、まず推定すべき方程式を与えねばならない。

変量Vを、他の諸変量X, Y, Z,, の線型関数として示す場合、係数ベクトルをBとすれば、

$$V = (X, Y, Z, \dots) B$$

と書くことができる。この一般的な表現をそのまま使うこととする。

これらの変量は、当然、この命令文が翻訳・実行される前に、それぞれシステム・データ・ファイルに何の何かの命令文によって入力され、名称が定義されている筈である。これらの変量の名称を一般に RE_i で示すこととする。Bは一次元の実配列であり、その変数名称を一般的にFで示す。この実配列名は、この命令文以外では必要ないかも知れず、この命令文ではじめて現われ、後は使用されないものであっても差支えない。

以上の一般的な変数名をつかうと、推定すべき方程式を与えるための構文規約は、

RE 1 = (RE 2, RE 3,, RE n) F

で示される。nはこの命令文で用いうる変数の最大個数である。

BMDPIR のプログラムでは、この他にいくつかのオプションを指定できる。一つは入力データの種類を選択できるオプションであり、他の一つは、結果を各種のグラフで表示するためのグラフ種類を指定するオプションである。これらのオプションは、書かなくてもよく、その場合はシステム・データ・ファイルから各変数のデータが入力される。オプションを利用したいときは、おのおの次のように書くものとしよう。

入力オプションとしては、システム・データ・ファイルからの入力を標準とし、他にあらかじめ計算された共分散行列または相関係数行列を入力してもよい。これを指定するキーワードを

共分散行列	COVA
相関係数行列	CORR

とし、一般的に **KEY 1** で示す。オプション指定は

INPUT = (KEY 1)

と書くことと定める。

グラフ出力としては次のものがある。付けられている単語はキーワードであり、一般的に **KEY 2_i** で示す。

被説明変数の原データ・推定値比較	VARIABLE
残差の確率分布図	PROBIT
残差および残差平方	RESIDUALS
全グラフの出力	ALL
グラフを出力しない	NONE

オプションが書かれていないときは原データ・推定値比較 **VARIABLE** が出力されるものとする。

オプション指定は、次のように書くものと定める。

PLOT = (KEY 2_i, KEY 2_J)

命令文の終りには必ず特殊記号 [: または ;] を書く。この後に同じ行に次

の命令文を書いてもよく、また、[: または ;] がなければその命令文は次のカードに続いて書かれているものとみなされる。命令文が [;] で終わっているときは、次の命令文も同じ手法に関する命令文であって、命令部の記入は省略されているものとみなす。

以上をまとめると、多変量線型回帰分析の命令文は、次のような構文であると規定される。[] で囲まれた要素は、書く必要がないときは省略してもよいことを示す。

```

LINEAR REGRESSION [#n],
  RE 1 = ( RE 2 [, RE 3 [, …… [, RE m ] ] ] ) F
  [, INPUT = ( KEY 1 ) ] [, PLOT = ( KEY 2 i
    [, KEY 2 j [, KEY 2 k ] ] ) ] :

```

構文の登録

一つの応用言語システムについて、そのシステム内で認められる命令文の構成要素の種類を定め、それぞれに、RF, F, KEY 等のコードを割当てておけば、上記のように記号化された構文規約をシステムに与えることによって、コンパイラはそれを入力された命令文と対比しつつ解釈し、翻訳することができる。記号化された構文規約を“構文表”(syntax table)と呼ぶこととすれば、構文表をシステム・ファイルに登録するだけで、1つの汎用コンパイラが多様な命令文の翻訳を一括して処理できる。応用言語システムの設計段階においては、単にそのシステムの命令文の中で使用することのできる要素の種類だけを考えればよく、個々の命令文の構文を設計段階で詳細に規定する必要はない。処理ルーティンを組込む段階で既存の処理ルーティンの機能に適した命令文を考えて、構文表を作製すればよく、試用して見た結果、命令文が書きにくかったり、機能的に不都合があったりした場合は構文表だけを修正すればよい。

パッチ処理・会話処理を一括して必要且十分な翻訳情報を与えるために、FMCP (FOIL MONITOR COMPILING PROCESSOR) とよぶルーティンがモニタ・システムに内蔵されている。FMCP はそれ自身が特有の言語をもち、この言語によって構文を記述し汎用コンパイラのための構文表を出力す

ると共に、会話処理に必要なメッセージのファイリングを行う。命令文に含めてよい構成要素は 234頁に示したが、必要があれば取扱う要素を追加しうる。

構文記述言語およびそれによるプログラミングについては、後に説明する。

LINEAR REGRESSION に関する上述の構文規約を、FMCP 言語によってプログラムした例を、示しておく。

このプログラムを翻訳して内部形式の構文表を作製し、システム・ファイルに登録するための命令文は、

REGISTER SYNTAX (short-name) :

であり、この場合、'REGRESSION' の構文表を登録するのだから

REGISTER SYNTAX (REGRESSION) :

である。この命令文を前に示した命令部登録プログラムの中で

を入れてある部分に挿入し、データとして FMCP プログラムのカードをつけて実行させればよい。

構文表が登録されると、汎用コンパイラは、'REGRESSION' (内部においては 'GRESSION') 命令文の翻訳および会話処理を実行することができる。いま利用者が、

LINEAR REGRESSION,

**WATER = (HOUSEHLD, FACTORY, OFFICE,
TIME) C1,**

PLOT = (VARIABLE) :

という形の処理を希望しているとしよう。

**WATER, HOUSEHLD, FACTORY, OFFICE,
TIME**

等のデータは、すでにシステムの標準データ・ファイルに格納されており、この命令文は、ある地域の水の需要量をその地区の家族数、工場数、事務所および時間的趨勢によって説明するモデルをつくるためのものと考えよう。

バッチ処理の場合は、この命令文はカードに穿孔されてよみこまれ、汎用コ

F M C P PROGRAM EXAMPLE

L I O N S O N P X T C S E T O					PARAM	MESSAGE
	PG				GRESSION	LINEAR REGRESSION
*	GT	M				RE1 = (RE2, RE3, , RE16) F ;
MM	RE	1	EX	1	-----	GIVE THE MODEL FUNCTION ELEMENTS -----
	SP	=	E1			DEPENDENT VARIABLE
NN	RE	1	EX	2		INDEPENDENT VARIABLE, IF THE LAST ATTACH ')'
*						
	DO	II	3	16		
	SP	,	J1			
	RE	1	EX	II		#NN
II	DT				-----	COEFFICIENTS VECTOR -----
*						
J1	SP)	E1			COEFFICIENTS VECTOR
	F	2	EX	17		
	NP			II		
	SP	,	ED		-----	INPUT DATA OPTION -----
*						DO YOU REQUEST NON-STANDARD TYPE INPUT ?
B		J3				KEY IN 'COVA' IF COVARIANCE MTX IS GIVEN
MS	2					OR 'CORR' IF CORRELATION MTX IS GIVEN
					INPUT	
	W	ED				
	SP	=	E1			
	SP	(
	K	K1	E2	18		
	SP)			-----	GRAPHIC DISPLAY OPTION -----
*						DO YOU REQUEST GRAPHIC DISPLAY(S) ?
J3	B	ED				3-KINDS OF GRAPHS ARE AVAILABLE
MS	5	J4				' VARIABLE' COMPARES ORIGINAL & ESTIMATED
						' PROBIT' PROBABILITY DISTRIBUT. OF ERR
						' RESIDUAL' RESIDUALS & SQUARED RESIDUALS
						' ALL' ALL 3 GRAPHS MENTIONED ABOVE
J4	W	ED			PLOT	
	SP	=	E1			
	SP	(
LL	K	K2	E2	19		IF THE LAST ATTACH ')'
*						
	DO	JJ	20	21		
	SP	,	J5			
	K	K2	E2	JJ		#LL
JJ	DT					
J5	SP)				
*						
ED	ID	:	E1			
EX	//					

ンパイラは命令部を判別して、REGRESSION 命令文の構文表をファイルから取り出し、それを解析しつつ翻訳する。この過程は外見上、通常の翻訳と同じである。

タイム・シェアリングの場合は、前命令文の処理が終了すると

TYPE IN STATEMENT, IF STATEMENTS
TABLE IS REQUESTED KEY IN ?

というメッセージが出される。ここで、命令文の全文をタイプしたとすれば、一時的にバッチ・モードとなり、バッチ処理の場合と同じ過程の翻訳が実行される。命令文の書き方がわからないときには‘?’を入力すれば、命令文一覧表が数頁にわたって出力されるので、所望の命令をさがすことができる。いま、命令部だけを入力したとすれば、それを判別して構文表を取り出してから、自動的に会話モードに移って処理を開始する。

システムは、まず、記述部の書き方をメッセージとして示してから、より詳細な説明を必要とするか否かを尋ねてくる。‘YES’ と応答すれば、その命令文の処理内容の詳細説明を書き出す。次いでこの場合には、最初の要素たる独立変数名の入力を求め、応答がなされると、つづいて従属変数名の入力が求められる。ここで、記述部の書き方が理解できたので残りの部分を一括して

FACTORY, OFFICE, TIME

と入力したとすれば、直ちに仮バッチ・モードに移り、入力された要素をすべて翻訳し終るまでメッセージは出さない。この状態では、まだ命令文は終了していないのであるから、自動的に、ふたたび会話モードにもどり、次の係数を格納すべきベクトル名の入力が要求される。以下同様にして会話は命令文が終るまで進行し、必要な要素がすべて入力されると完成した形の命令文が出力され、その命令文の実行に移る。

汎用コンパイラは、構文表の解析に当って、現在どのモードで処理が行なわれているか、未処理の入力要素が残っているかどうかを常に判別し、両者の状態判定結果により、自動的に次に採るべき処理モードを選択する。利用者は、会話型処理の場合に自分の理解している所まで入力すれば常にそれ以後につい

では会話的な誘導が得られ、わかれば一括入力に移って会話モードのわずらわしさを避けられるので、最も効率的に命令文入力を実行できる。

次に、上述の LINEAR REGRESSION 命令文の会話モードによる入力例を示しておく。

EXAMPLES OF CONVERSATIONAL INPUT

```
TYPE IN STATEMENT, IF STATEMENTS TABLE IS REQUESTED KEY IN ?
=linear regression, water=(househld,factory,office,time) cl:
```

```
TYPE IN STATEMENT, IF STATEMENTS TABLE IS REQUESTED KEY IN ?
=linear regression
```

```
LINEAR REGRESSION (GRESSION)
```

```
REL = (RE2, RE3, ..... , RE16) F :
```

```
IF MORE INFORMATION IS NECESSARY KEY IN ?, OTHERWISE BLANK
=
```

```
KEY IN REAL
```

```
DEPENDENT VARIABLE NAME
```

```
=water
```

```
KEY IN REAL
```

```
INDEPENDENT VARIABLE NAME, IF THE LAST ATTACH ')'
=househld
```

```
KEY IN REAL
```

```
INDEPENDENT VARIABLE NAME, IF THE LAST ATTACH ')'
=factory, office, time )
```

```
KEY IN REAL ARRAY NAME
```

```
COEFFICIENTS VECTOR
```

```
=cl
```

```
KEY IN 'YES' OR 'NO'
```

```
DO YOU REQUEST NON-STANDARD TYPE INPUT ?
```

```
=no
```

```
KEY IN 'YES' OR 'NO'
```

```
DO YOU REQUEST GRAPHIC DISPLAY(S) ?
```

```
=
```

```
3 KINDS OF GRAPHS ARE AVAILABLE
```

```
'VARIABLE' COMPARES ORIGINAL & ESTIMATED DATA
```

```
'PROBIT' PROBABILITY DISTRIBUTION OF ERRORS
```

```
'RESIDUAL' RESIDUALS & SQUARED RESIDUALS
```

```
'ALL' ALL 3 GRAPHS MENTIONED ABOVE
```

```
KEY IN KEY-WORD
```

```
IF THE LAST ATTACH ')'
=variable )
```

```
LINEAR REGRESSION, WATER=(HOUSEHLD,FACTORY,OFFICE,TIME) C1
```

```
PLOT=(VARIABLE) :
```


FOIL システムの構造

FOIL V-10 モニター

FOIL V-10は、高水準応用計算言語システムのための汎用モニターである。応用言語で書かれたプログラムの一括〔バッチ〕入力、または、会話型式の誘導による入力と、その翻訳、およびプログラム内容の実行とを、統一的に管理する。このモニターは、システムに含まれる命令文の数が増大し、それに含まれるプログラムの長さ和本数とが、量的に尠大なものになっても、設備規模が充分であるかぎり、効率を低下させることなく処理できるように設計されている。

一括処理と時間割処理とでは、利用者の希望する処理様式に差があることを考慮しなければならない。例えば、一括処理の場合には、プログラムの翻訳とその実行とを、おのおの別のフェーズとする方があらゆる点から見て有利である。しかし、時分割の場合には、1命令文の翻訳が終れば、直ちにそれを実行して、利用者が試行錯誤的に分析を進めうようにした方がよいといえる。

FOIL V-10 モニターは、その処理経路を MODE とよぶスイッチで制御できるように設計している。MODE は4個のスイッチより成り、そのおのおの値を適宜に設定することによって、数値の処理方法を選択することができる。このスイッチは、外部から FOIL のプログラムによって設定できるようにして、利用者に処理方法を選択させるだけでなく、システムのプログラミングに当って、適切な処理を行うことによって、必要な処理方法の変更を自動的に行なわせ、使いやすいシステムを作ることができる。

MODE (1) 処理状態指示スイッチ

現行処理形態（翻訳、実行、終了）を指示し、その処理の終了時には、次の段階で行うべき処理形態を指定する。

MODE (2) 処理状態判別スイッチ

入力端末の種類を判別して自動的に設定される。

MODE (3) 処理様式指定スイッチ

命令文による指定により翻訳・実行の組合せ様式を指定す

る。

内 容	MODE(1)			MODE(2)	MODE(3)
	処 理 時	翻 訳 終 了 時	実 行 終 了 時		
0	F M C P に よる 翻 訳	エ ラ ー が な け れ ば 実 行	翻 訳	T S 時 会 話 処 理	1 命 令 文 づ つ 翻 訳 ・ 実 行
1	特 定 ル ー テ ィ ン に よる 翻 訳	強 制 翻 訳	—	T S 時 一 括 処 理	翻 訳 後 実 行
2	実 行	強 制 実 行	強 制 実 行	一 括 処 理	翻 訳 の み
3	終 了	E N D 検 出	E N D 検 出	T S 端 末 に よ る 一 括 処 理	実 行 の み

スイッチの値に対応して、どのような経路が選択されるかを、上の表に一括した。MODE(4)は予備であって現在は用いていない。

また、MODE(3)によって、処理の基本様式が指定される。FOILシステムでは、1命令文ずつ翻訳しては直ちにその内容を実行する処理をくりかえすのを、基本としており、プログラム上に命令文による指定がなければ、この方式をとるものとする。この方式には制約が伴うので、通常のコンパイラの方式に従って、まずプログラムを一括翻訳し、エラーのない場合のみ一括実行することも、または、翻訳のみを行うことも可能である。既存の翻訳結果にもとづいて実行のみを行うことも可能であるが、この部分は現行システムではプログラムの整備がなされていない。MODE(3)は、通常0にセットされ、命令文によって、1または2が設定される。

MODE(2)には、システムが最初に、アクセスしている端末の種類を識別して、0〔時分割処理〕または、2〔一括処理〕を設定する。時分割端末から一括処理を要求したときは、当然、2に設定される。時分割処理時には、会話による誘導が原則となり、各命令文ごとに、それを構成している要素を、システムの誘導に従って、一つずつ入力してゆくのであるが、強いて誘導を必要としない場合には、1つの命令文の全文を入力してもよく、途中まで、または、途中から残りを全部入力してもよい。会話処理の過程で、次に処理すべき要素が

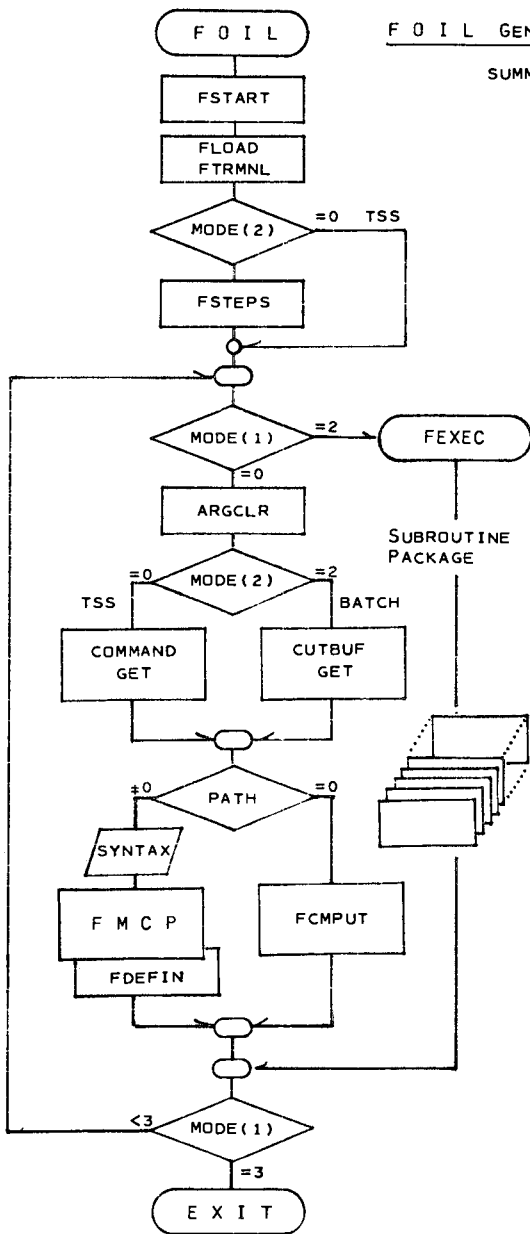
システムの誘導以前に既に入力済であったときには、MODE(2)の内容は1に設定され、一括処理とおなじ経路をたどって、不必要な会話は行なわれない。処理すべき要素が、未入力であることを発見すると、MODE(2)は0となって、会話による誘導が再開される。

MODE(1)は2種の機能をもつ。1つは、処理の進行中に現行の処理形態、すなわち、翻訳中であるか、実行中であるか、またはEND命令文を検出して、終了処理を行いつつあるか、を指定する。第2に、1つの形態の処理が終了した段階、たとえば、一命令文の翻訳が終った段階で、他のMODEスイッチの状態、および、いままでの処理の性格を考慮して、次に行なうべき処理を定めてそれに対応する値を設定する。この値によって、制御が主ループの先頭に帰されたとき、次に辿るべき経路が決定されることになる。

FOIL V-10は、次頁の図に示すような構造をしている。コンパイラ部分は、所要のルーティンをすべて内蔵しているので、個々の命令文の追加には、後に説明する構文解析言語FMCPによる命令文解析プログラムを追加すればよい。図の右端に、主要サブルーティンの機能を略記したが、この詳細については、次節以下で説明する。これらのうちFEXECは、実行ルーティン用のモニターである。FOILにおいて、翻訳結果は、クローズド・タイプに構成されたサブルーティン・サブプログラム形式の実行ルーティンに対して引渡される引数パラメーターに他ならない。命令文のセットに対応して、その実行のためのサブルーティン・パッケージがあり、FEXECは両者の中間に介在して、必要なパラメータを授受し、制御を引渡してゆくためのルーティンである。

F O I L GENERAL PURPOSE COMPILER

SUMMARY FLOW CHART



MAIN SUBROUTINES

1. FSTART
COMMON TABLES
INITIALIZE
2. FLOAD
STATEMENTS TABLE
INITIAL LOAD
3. FTRMNL
IDENTIFY
ACCESSING TERMINAL
4. FSTEPS
SOURCE STATEMENTS
DECOMPOSITION
5. ARGCLR
ARGUMENTS TABLE
INITIALIZE
6. COMMAND GET
CONVERSATIONAL
COMMAND INPUT
7. CUTBUF GET
DECOMPOSED STATE-
MENTS READ IN
8. FMCP
GENERAL PURPOSE
TRANSLATOR
9. FDEFIN
VARIABLE NAME
TRANSLATION
ROUTINES
10. FCMPUT
GENERAL PURPOSE
ARITHMETIC
EXPRESSIONS
TRANSLATOR
11. FEXEC
EXECUTION MONITOR

FOIL モニターの主要サブルーティン

FOIL V-10 モニターを構成する主要サブルーティンは、それぞれ、次に示すような機能をもつ。複雑な機構のものについては、次節以下に詳説することとし、ここでは、FOIL V-10 モニターの機能説明をかねて、各サブルーティンについて略述する。

1. FSTART

システム共通領域の初期化ルーティン。基本共通領域についてはプログラムは共通であるが、新しくサブシステムを構成する場合には、そのサブシステムの共通領域の初期化ルーティンを、このルーティンに別途追加しなければならない。

2. FLOAD

命令文表初期入力ルーティン。BASIC サブ・システムの命令文表を読みこむ。他のサブ・システムの命令文表は、COMPILE BY 命令文により読込まれる。

3. FTRMNL

端末番号識別ルーティン。現在 FOIL をアクセスしている端末の番号を識別し、その種類を判定して登録する。時分割処理と一括処理は、このルーティンで識別される。

4. FSTEPS

原始命令文分解ルーティン。一括処理の場合は、始めに FOIL のソース・プログラムを一命令文ごとに区分して、各命令文を構成要素別に分解する。この処理は V10 CUT ルーティンの機能に拠る。命令文にレーベルのついているときは、その定義処理を行い、命令文の種類を識別して、結果をファイルに出力する。時分割時には、この前処理は行なわれない。

5. ARGCLR

引数表初期化ルーティン。各命令文ごとの翻訳開始に先立って、翻訳結果を格納すべき引数表 [ARGTBL] の内容を初期化する。

6. COMMAND GET

時分割処理時における命令文名称入力ルーティン。会話誘導によって命令名称の入力を求め、それを識別して、命令文の種類を決定する。なお、これにはレーベル定義ルーティン、および、制御番号設定ルーティンを含む。

7. CUTBUF GET

分解済命令文入力ルーティン。一括処理の場合、FSTEPS ルーティンによってファイルに出力された、命令文分解結果を、ここで再入力する。

なお、6. COMMAND GET と 7. CUTBUF GET とは、2つで単一のルーティン CMDGET にまとめられている。

8. FMCP

汎用翻訳ルーティン。演算命令文を除く一般の命令文の汎用翻訳ルーティンであり、SYNTAX 表をファイルから入力して、それと命令文を対比・解析することにより、翻訳を実行する。時分割処理時には、会話による命令文入力の誘導をも併せて処理する。

9. FDEFIN

変数名翻訳処理ルーティン。FMCP に含まれており、命令文中のデータ要素の変数名・定数等の、個々の翻訳処理ルーティンの集合である。

10. FCMPUT

汎用演算命令文翻訳ルーティン。算術代入文、IF 文等、演算式を含む命令文の汎用翻訳ルーティンである。演算式を含む命令文は、多数の一定範囲の命令文を編集して、一命令文に編集することが可能であり、その編集ルーティンをも含んでいる。

11. FEEXEC

実行ルーティン・モニター。各命令文に対応して、それを実行するための諸ルーティンは、サブルーティン・パッケージを構成して、システム・ライブラリに収納されている。このルーティンは、翻訳結果にもとづいて、所要の実行ルーティンを探り出すと共に、翻訳結果から、この実行ルーティンの処理に必要な諸引数を、探し出された実行ルーティンに転送し、しかる後、これに制御

を渡すモニター機能を果している。

バッチ処理用命令分解ルーティン FSTEPS

バッチ処理の場合、命令文のソース・カードは、翻訳に入るに先立って命令文分解ルーティン FSTEPS によって一括前処理される。FSTEPSは入力されたプログラムを、終了記号を判別して個々の命令文に分け、文法の一般規約にしたがって、構成要素別に分解する。さらに、個々の命令文につけられているレーベルを定義し、命令文の種類を判別して、対応する内部コマンドをもとめる。これらの結果は、命令文分解表 CUTBUFに編集され、ファイルCUTFILに書き出される。

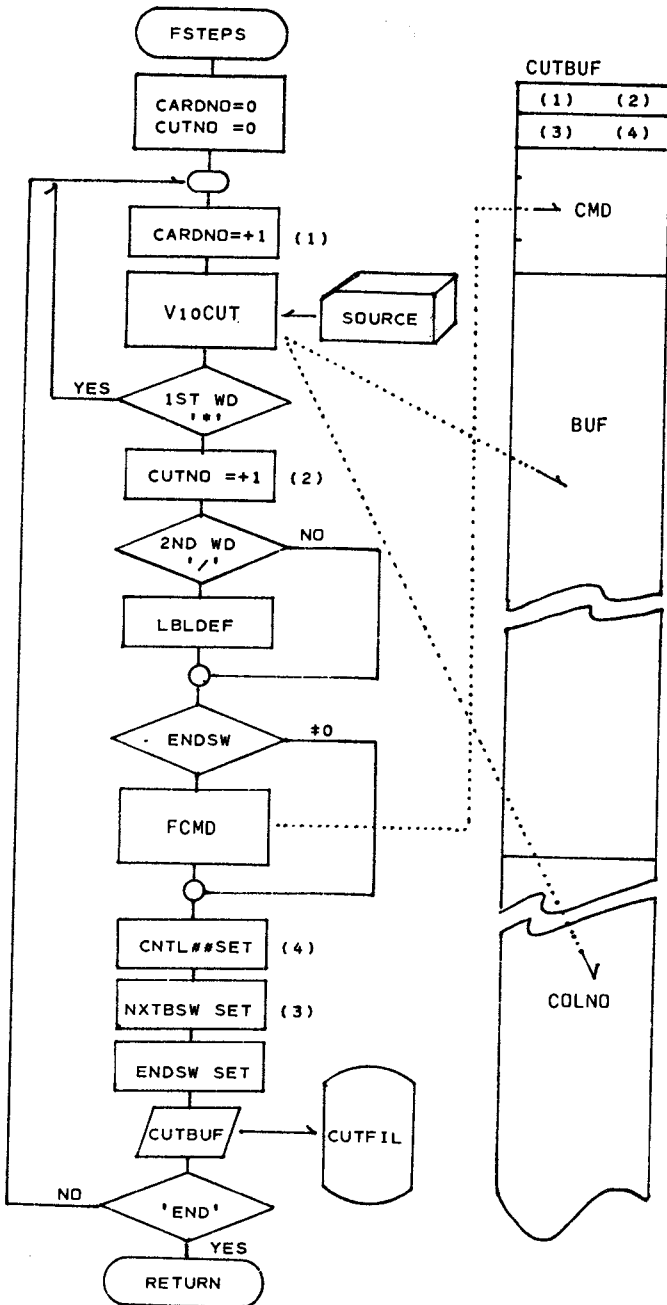
このルーティンを経ることによって、命令文の翻訳は、自由書式の高水準言語を対象とするにもかかわらず、実質的には固定書式のパラメータ・カードを処理するのと大差ないことになる。

FSTEPS ルーティンのフロー・チャートおよび CUTBUF テーブルの構造図を次に示した。ルーティンが CALL され、初期化の手続きが終ると、主ループに入って分解処理を END 文のくるまでくりかえす。

主ループ このルーティンの流れは比較的簡単で、分解ルーティン V10CUT レーベル定義ルーティン LBLDEF, 命令種類判別ルーティン FCMD, 制御番号処理, 分解表書出し, 等から成立っている。

V10CUT 主ループに入ると始めに、V10CUT とよぶ分解ルーティンに制御がわたる。分解ルーティンそのものについては、後に詳述する。このルーティンは、ソース・ステートメント入力を内蔵しており、また、タイム・シェア時の会話型、不連続入力も処理できるよう、多目的に設計されている。バッチ処理の場合、1 命令文が複数レコードにまたがっているときでも、逆に1レコードに数命令文が含まれている場合でも、はじめの1命令文を判別して処理する。出力は分解バッファ BUF の内容となる。命令文は構成要素ごとに分解され、一要素は常に8バイト1語（ただし、文字列は除く）に編集されて、順に BUF に格納される。

FOIL の一般規約によって、命令文の構成要素は、特定の形式に分類され編



集される。ブランクは無視されず、要素間の区切りとみなされる、複数個連続していても1個であっても同じに取扱われる。分解された各要素がカード上で占めていた最終文字位の置欄番号は、COLNOの対応欄に保存される。

分解が終ると、命令文の属性判別に入る。

COMMENT FOILの文法規約では、1つの命令文が終わってのち最初に書かれているブランク以外の文字が、スター〔*〕である。

CONTINUE : * END OF LOOP

* STEP NO. 5 START

とき、以下そのカードの終わりまでをコメントとみなす。上の例で

END OF LOOP

STEP NO. 5 START

等はコメントである。ブランクは何個続いていても1個とみなされるので、ブランク以外のものが最初に命令文カードにあれば、それはBUF(1)に入れられる。したがって (BUF(1).EQ.'*') であれば、それ以下カードの終わりまですべてコメントであり、次に翻訳すべき欄番号を0にして新しくカード・イメージをよみ第1欄から分解を始めることをV10 CUTに指定し、主ループの先頭にかえる。コメントはファイルに出力されない。

LBLDEF 命令文はその先頭にレベルを付して、他の命令文で参照することができる。レーベルとしては、単語および整定数(文番号)を用いることができる。これらは、いづれにしても、1要素として編集され、BUF(1)に格納される筈である。

自由書式であるから、レーベルを他の要素と区別するために、命令文の主文との間にスラッシュ〔/〕を入れる。この特殊記号は、レーベルの書き方にまちがいが無い限り、必ずBUF(2)に入る。したがってBUF(2)を調べ、それが〔/〕であれば、BUF(1)はレーベルである。

レーベルは、その名称または番号を、レーベル定義表に登録することによって定義される。

レーベル定義表はレーベル名表LBLNMと、位置登録表LBLLCとからなる。

BUF(1) の内容と同じものが、LBLNMにすでに登録されていないかを調べる。もし同じものがあれば2重定義となるのでエラーである。なければ、それをLBLNMの空番に格納し、有効命令文の順番号 CUTNO の値を、対応する LBLLC に入れて定義を終る。

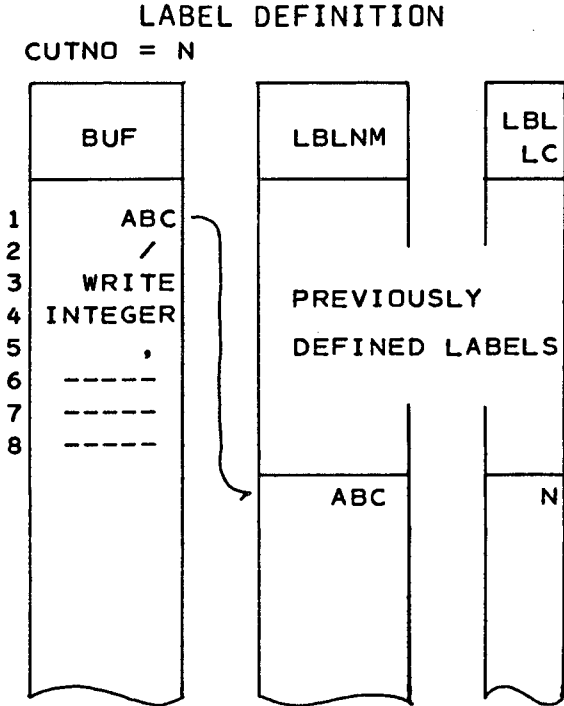
ENDSW レーベルの処理につづいて、命令名の処理をおこなう。

レーベルの次に、もしくは、レーベルの付いていない場合は文の先頭に命令名がくる。ただし、FOIL の文法の一般規約で、先行する命令文の終了記号がコロン [:] でなく、セミコロン [;] であるときは、次の命令文も同じ命令名であり、その記入は省略されているものとして処理する。

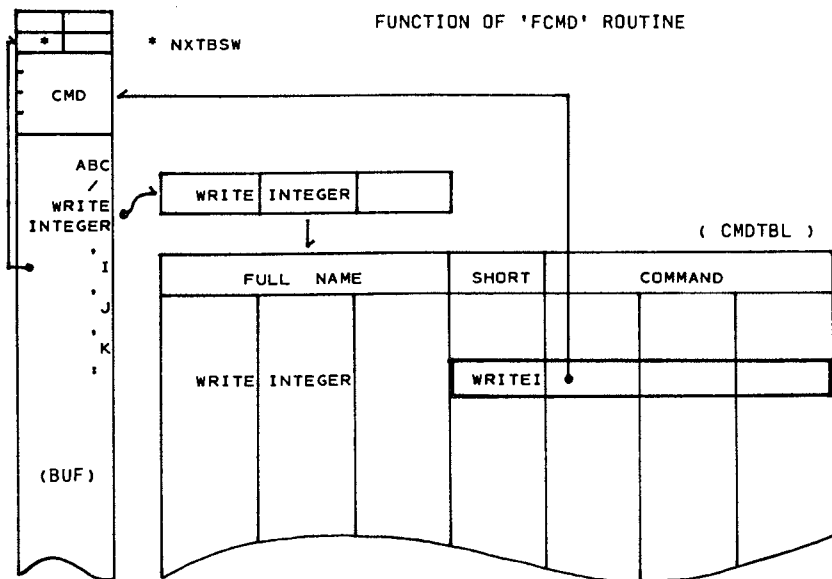
先行する命令文の終了記号が何であったかは、V10 CUT が判別して登録している。そのレジスタ ENDSW が0でなければ [;] であり、この命令文も同じ命令名をもっとして処理し、次の FCMD の処理は行なわれない。0であれば [:] であって、先行したのとはちがう命令文であるとみなす。

FCMD 命令名の判別は SUBROUTINE FCMD によって行なわれる。

FCMD はまず、命令名の部分を BUF から取り出し、それで命令表 CMDTBL を走査する。命令表は、1命令分が8バイト7語からなり、指定さ



れたサブ・システムで使用可能な全命令を登録している。7語のうち、前3語に命令の名称 (FULL NAME)が、次の1語に簡略名 (SHORT NAME)が、そして残り3語に内部コマンドが格納されている。



命令名は3語以内の単語からなっており、別に1語の簡略名がある。命令文にはこのいずれを用いてもよく、命令名が1語のときは FCMD はまず、SHORT NAME を走査する。

命令表上に合致するものがあれば、SHORT NAME と内部コマンドを含めて4語が、CUTBUF 中の CMD の部分に転送される。これで CUTBUF には、翻訳ルーティンおよび実行ルーティンの所在に関する情報が記録されたことになる。

命令名の終りには、必ず、特殊記号、区切語、または終了記号が来る規約である。

CNTL 命令名の終りが特殊記号〔#〕でないときは、このステップは実行されない。〔#〕であるときは、次に整数で、この命令文の制御番号が書かれて

いるものとみなし、その値を CNTL に入れる。FOIL は命令文に任意に制御番号を付しうる規約であり、制御番号の機能は、命令文ごとに定められ、マニュアルに規定されるものとする。通常、出力リストのレベル指定や、入力型式の指定等に用いる。

NXTBSW 命令名の終り、制御番号の付けられているときにはその次、には必ず特殊記号、区切り語または終り記号が来ることはすでにのべた。FSTEPS では翻訳はこれ以上行なわれないが、翻訳ルーティンに対する情報として翻訳を開始すべき BUF 上の位置を NXTBSW に記録しておく。

終りが [,] または区切り語 [%] であるときは、NXTBSW にはその次の位置を入れる。その他の特殊記号のとき、および、終了記号のときは、その位置が NXTBSW に入れられる。

ENDSW SET この命令文の終了記号が BUF 上のどこにあるかは V10CUT の出力として引数に与えられているので、終了記号の種類を調べる。[:] であれば 0、[;] であれば 1、が ENDSW にセットされる。

以上で、この命令文に関する FSTEPS の処理は終了する。命令文が、END 命令文でなければ、主ループの先頭にかえて、次の命令文について同様の処理がくりかえされる。END 命令文であれば、メッセージを出して、FSTEPS の全処理を終了し、制御をシステム・モニターにかえて、次の段階に移行する。

翻訳用モニター FCMPIL

翻訳過程は、FMCP と FCMPUT の二つのルーティンによって処理される。このうち、FCMPUT は、COMPUTE、IF 等、算術式、論理式、関係式を含む命令文の解析ルーティンであり、FMCP はこれらを除く一般の命令文の翻訳ルーティンである。

TSS の場合は、処理開始とともに初期入力ルーティンが働き、基本的にどのプログラムにも必要な、三、四の命令文入力を誘導する。それ以後、利用者は COMMAND GET ルーティンによって、命令名称の入力を行うことになる。このルーティンには、利用者の要求によって、命令文一覧表をディスプレ

イする機能があり、利用者は一覧表を調べて使用したい命令文の番号を入力するだけで、次の誘導をうけられる。

バッチ処理の場合、命令名称はすでに解析されており、CUTFIL から、入力される。

TSS の場合でも、命令文の書き方がわかっているときは、1 命令文全部を入力すればよい。この場合は、ルーティンが、入力状況を判別して、自動的にバッチ的な処理に切りかえる。

命令名称によって、内部コマンドが決定されるので、その中の翻訳コマンドを調べて翻訳が進められる。翻訳コマンドは 8 バイト 1 語であるが、その最終 2 バイトが PATH レジスタに入る。この PATH の内容は、対応する構文表のレコード番号を示す。これが 0 の場合は、FCMPUT ルーティンに、0 でない場合はそのレコード番号によって構文表を入力して FMCP ルーティンに制御がわたる。FMCP ルーティンの機能については次の第 3 章において詳述する。0 の場合は、翻訳コマンドの他の部分を解析して、処理ルーティンを選択し、ECMPUT ルーティンに入る。FCMPUT については、第 4 章において詳述する。

FMCP の出力は引数表 ARG_TBL であり、FCMPUT ルーティンの出力は演算定義表 CMPTBL である。MODE(2) をセットして次の処理方式を指定し主ルーティンにかえる。

会話処理時の初期命令入力としては、

START 命令

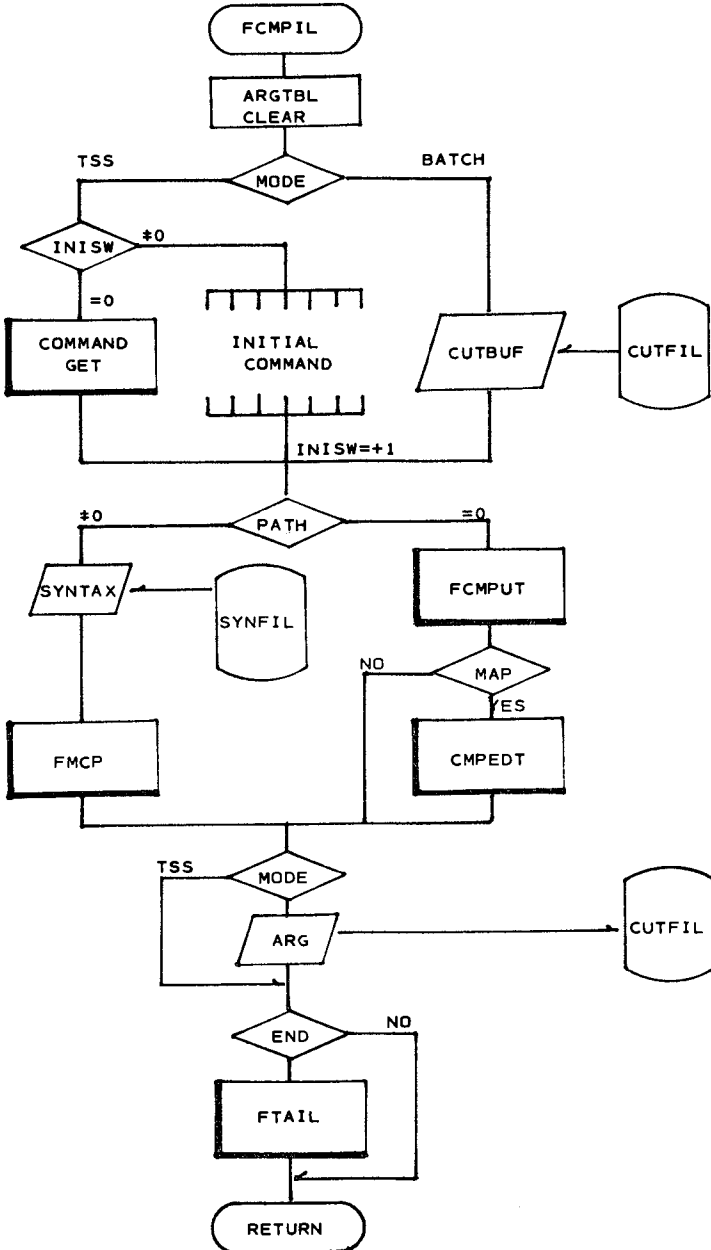
SUB-SYSTEM 名 指定命令

最大標本範囲設定命令

等がある。

FSTART による共通領域の初期化のとき、INISW には 1 が入れられる。したがって、最初 FCMPIL に入るとき、初期命令入力ルーティンに分岐し、その第 1 のものを処理する。この処理内容は、内部に対応する命令文を編集することである。その後は、通常の命令文として処理される。終ると、INISW

F O I L MULTI PURPOSE COMPILER FLOW CHART



に1を加える。

次の命令も同様の処理を受けるが、最後のルーティンで、INISW に-1を入れる。これによって、この命令文の処理が終ると INISW = 0 になり、一般処理が行なわれることになる。

COMMAND GET ルーティンは、命令文の入力ルーティンであり、命令文のうち、レーベルと命令名称を処理する。

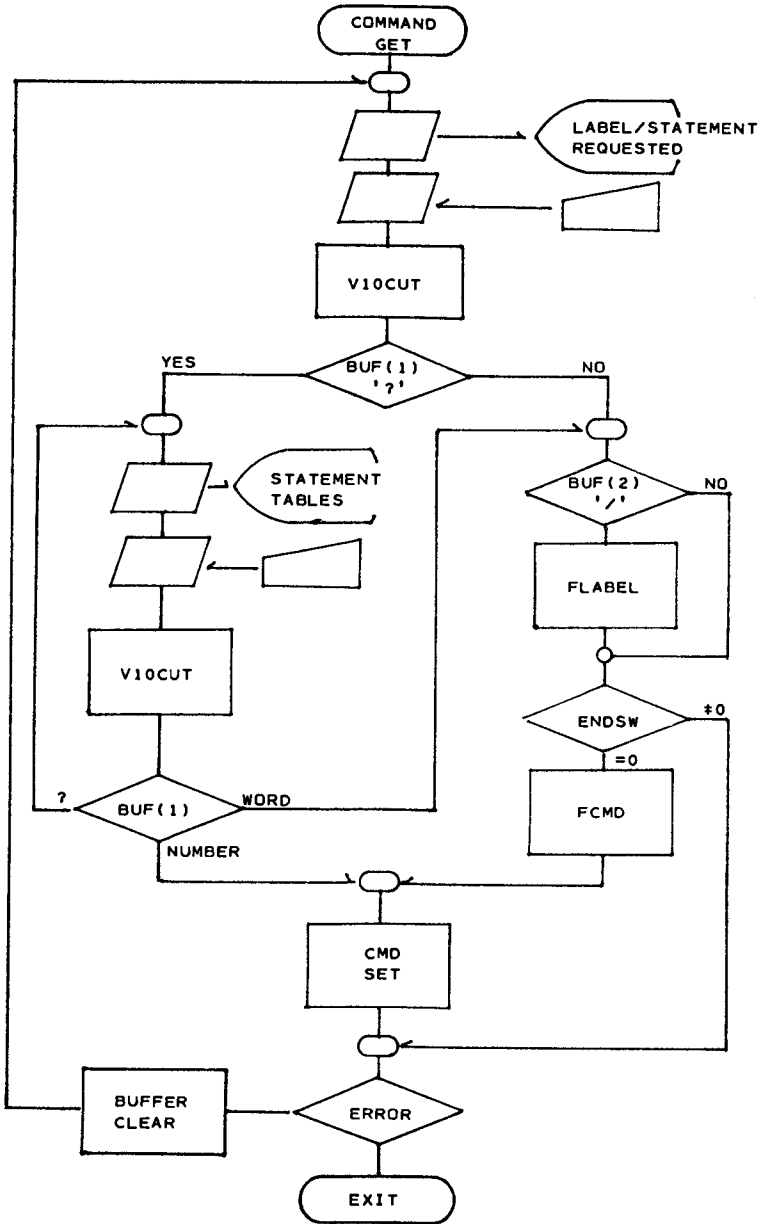
最初に、命令文の入力を要求するメッセージが出される。この際、レーベルは必ず命令名称とともに入力しなければならず、レーベルのみを独立で入力できるのは、先行する命令文が ' ; ' で終わっている場合に限られる。

入力された命令文は直ちに分解され、BUF に格納される。入力内容が ' ? ' であれば、命令表がディスプレイされる。さがしている命令文があれば、その番号、または命令名称あるいは全命令文のいずれでも入力すればよい。命令文入力の場合は、' ? ' を入力しなかった場合と同じである。番号を入力すれば、対応するコマンドを CMD に入れる。' ? ' の場合は、命令表の次の頁をディスプレイする。

命令文入力が行なわれた場合は、レーベルおよび命令名称の処理を行う。この処理は、FSTEPS で行うものと同様である。以上の処理で、CMD の内容が定まり、翻訳過程に制御が渡る。

この間にエラーが発見されたときは、エラーメッセージが出され、一括処理の場合はその命令文の翻訳を中断して次の命令文に移るが、TSS の場合は、BUF が初期化されて、ルーティンの先頭にかえる。このエラー処理は、次のように行なわれる。V10 CUT によって、入力命令文が分解されるとき、BUF への分解結果の格納開始番地（この場合は1）が BUF5 に、格納終了番地+1、すなわち、次回の格納開始番地が BUFE に書きこまれる。

エラーが発生すると、この情報によって、入力された部分は消去されて何も入力されなかった状態にもどる。命令名称の入力ミスであっても、同時に行なわれたすべての入力を消去するので、レーベルを書いた場合は、もう一度はじめから入力し直さねばならない。全命令文を入力している場合は入力したもの



全部が無効である。

実行用モニター FEXEC

翻訳結果の実行は、実行用モニターFEXECによって制御される。翻訳の結果、コマンドがえられるが、その中には実処理ルーティンの PHASE 名が、CMD(2)に与えられている。FEXECはこの PHASE 名を用いて対応するルーティンをライブラリから取りこみ、それに制御を引きわたす。

各実処理ルーティンは、独立のプログラム、またはサブルーティン・サブプログラムである。FOIL システムとこれらのルーティンを結合するためには若干のインター・フェイス用プログラムを必要とする。一般に、独立プログラムならば、機能をコントロールするための、パラメータ・カードのセットをもっており、サブルーティン・サブプログラムの場合ならば仮引数のセットをもつ。インター・フェイス用プログラムは、翻訳の結果として、共通領域、特に引数表 ARG TBL、に格納されている各種のパラメータの中から、必要なものを選択して、パラメータ・セット、または、仮引数のセット、に引渡すためのものである。

サブルーティン・サブプログラムについて言えば、そのサブルーティンに対する CALL 命令文を、共通領域にある変数名を実引数に用いて書けば終る。

実処理プログラムによる実行が終了すると、制御はモニターにもどり、同様の処理をくりかえしてゆく。END または STOP 命令文が来ると、オペレーティング・システムにかえる。

FOIL は、各種の型のデータに対して、広汎に対応しう様に設計されているが、特にクロス・セクショナルなデータ型式のシステムには、独特の処理を行い、高い自由度を保持しつつ分析を遂行できる。このための処理を含めた実処理ルーティンの作製基準については、後章で詳述する。

命令文分解ルーティン V10 CUT

入力された命令文は、一旦かならず、分解ルーティン V10 CUT によって構成要素別に分解される。V10 CUT の識別する要素は次のようなものである。

1. 単語 アルファベット、カナで始まり、アルファベット、カナ、

数字よりなる8字以内の文字列、但し8字をこえる単語を書いたときは、前の字は脱落する。8字より少ないときは後に詰められ、前にはブランクが入る。

2. 文字列 ダッシュ〔'〕で囲まれ、間に〔'〕を含まない、任意の文字・記号からなる文字列。
3. 終了記号 1つの命令文の終りを示す記号〔:または;〕
4. 特殊記号 EBCDIC で認められているすべての特殊記号、
但し〔:; '〕を除く。* *および'が連続して2個ならんでいる場合には例外として2個で1個の特殊記号とみなす。
5. 区切り語 特定の単語で命令文中では区切り記号とみなされる。

A N D A R E A S A T B E B Y F R O M
F O R I S I N O F O N T H R U
T H R O U G H T O W I T H

 の16種が予約されている。追加可能。特殊記号%と翻訳される。
6. 論理演算子 .AND. .OR. .NOT.
7. 関係演算子 .LT. .LE. .EQ. .NE. .GE. .GT.
8. 論理定数 .T. .TRUE. .F. .FALSE.
9. 整定数 符号、小数点を含まない9桁以内の数字のみの列。
10. 実定数 符号をもたず、必ず小数点を間に1個のみもった数字の列、小数点以上0～9桁、小数点以下0～9桁、指数部を付けることができそのときは、文字Eと符号±および2桁以内の整数で示す。

実定数は間に一切ブランクを入れてはならない。

他に、倍精度実定数、複素定数、倍精度複素定数が処理しうるが、現在までの所、FOILシステムでは、これらは使用されていない。

1 2 3 4 5 6 7 8 9 0 ¹ 1 2 3 4 5 6 7 8 9 0 ² 1 2 3 4 5 6 7 8 9 0 ³ 1 2 3 4 5 6 7 8

WRITE INTEGER #1, A, B<5>, C<10> :

CUT BUFFER CONTENTS

	CUTNO =	15	CARDNO =	12
	CUNT =	17	ENDMK =	0
1	7			WRITE
2	15			INTEGER
3	17			#
4	18		1	I0
5	19			,
6	21			A
7	22			,
8	24			B
9	25			<
10	26		5	I0
11	27			>
12	28			,
13	30			C
14	31			<
15	33		10	I0
16	34			>
17	36			:

ブランクは要素間の区切りとして有効であり、必ず書かねばならない。

DO 100 I=1, 10 DO 100 I=1, 10

の2つの文を FORTRAN は、2つとも DO 文で、同じ内容のものと識別するが、V10 CUT は、後者の最初の要素を DO 100 I という単語として識別してしまう。但し、ブランクが複数個つづいても2番目以後は無視する。

命令文の最初にあらわれたブランク以外のコードが[*]であるとき、そのコード以下、カードの終りまでをコメントとみなして無視する。ブランク・カ

ードが挿入されていた場合も無視される。

BUF 内において、各要素は次のような特性をもち、互に他から識別可能である。この識別のために、ユーティリティ・ルーティン SEP が、システムに組みこまれており、システム保守には、これを自由に利用できる。BUF (*) は、IBUF (2, *) [整数], EBUF (2, *) [実数], および, LBUF (2, *) [論理数] と、EQUIVALENCE がとられており、8 バイトのうち、後 4 バイトを用いて識別が行なわれる。

単 語 8 バイトで後づめになっており、最終 1 バイトが空白ではない。(他の諸要素はすべて、最終 1 バイトが空白である)

文 字 列 先頭の IBUF (2, *) の内容が '△C0 △' であり、IBUF (1, *) に文字列の字数が入る。文字列そのものは、これに続いて BUF に格納されている。語数 M は

$$M = (\text{IBUF}(1, *) + 7) / 8$$

で得られる。

特殊記号・終了記号 記号がそのまま、第 6 バイト、2 個連続したもの [* その他] は、第 5, 6 バイトに入る。他はすべて空白である。

@ IBUF (2, *) = '△@△△'

** IBUF (2, *) = '**△△'

区切り語 IBUF (2, *) の第 2 バイト [BUF (*) の第 6 バイト] に % が入り、他は空白である。区切り語の内容の後 4 バイトが、IBUF (1, *) に入れられている。

BY | BY | △%△△ |

THROUGH | OUGH | △%△△ |

論理演算子・関係演算子 IBUF (1, *) はすべて空白であり、後の 4 バイトのうち、最後の 2 バイトは空白で、前 2 バイトに記号のうち、後の 2 字が入る。

.AND. | | ND |

.OR. | | OR |

.NOT.	O T
.LT.	L T
.EQ.	E Q
.GE.	G E

論理定数 IBUF (2, *) に 'LG 0 Δ' が入り, LBUF (1, *) にその値 .T. 又は .F. が入る。

整定数 IBUF(2, *) に 'Δ I 0 Δ' が入り, IBUF(1, *) にその値が, 2進変換されている。符号は別に, 特殊記号として1つ前に入っており, 利用者がプログラムで処理するものとする。IBUF (1, *) の内容は常に [正] である。

実定数 IBUF(2, *) に 'Δ E 0 Δ' が入り, EBUF (1, *) にその値が, 2進変換されて入る。符号は別に特殊記号として1つ前に入っており利用者がプログラムで処理するものとする。EBUF (1, *) の内容は常に [正] である。

識別の一般規約として, 特殊記号は, 終了記号, 区切り語, 演算子を含めて後2バイトが 'Δ Δ' であり, 定数は, 文字列を含めて, 後2バイトが '0 Δ' である。

COLNO (*) に, 各要素の最終の文字のカード上の欄番号が格納されている。

```

SUBROUTINE V I O C U T
1 (LIST, SW, TITLE, PAGE, LINE, CMD, CMDSIEQ
2 , HEAD, TAIL, CARD, COL, COLNO, CODE, CARDNO
3 , CUNT, LIMIT, BUF, EBUF, IBUF, ENDRG, ERR
4 , MODE, INPUT, OUTPUT)

```

°を付したものは定数を与えてもよい。型記入なきものは14

仮 引 数	型	寸法	初期値	内 容
°LIST				プリント制御=0のときは、ソース・プログラムをリストしない。
°SW				処理方式制御 + のときBUFを初期化して、その始めから分解結果を格納する。 - のときBUFを初期化せず、前回の結果の次から続けて格納する。 0 CARDの第COL欄から分解をはじめ第TAIL欄まで来れば、次のカードを読んで分解をつづける、[:]または[;]を検出すれば終了する。 1 必ず第HEAD欄から始め、第TAIL欄まで来るか、または[:]または[;]を検出すれば終了する。 10, 11 下1桁の機能は上述の通り、BUFに格納された分解結果をリストする。
TITLE	A 4	1 6		64字でソース・リストの各頁先頭にプリントするための題字を与える。
PAGE			0	出力頁数カウンタ、初期値を与え自動カウント
LINE			0	各頁ごとの行数カウンタ、自動カウント
CMD				システム用、通常ダミーを入れておく。
CMDSEQ			0	命令文順位番号、初期値を与え自動カウント
°HEAD				カード有効範囲の先頭欄番号[通常1]
°TAIL				カード有効範囲の最終欄番号[通常72/80]
CARD	A 4	2 0		カード読込バッファ
COL			0	CARD上の処理中欄番号、自動カウント
COLNO	1 4	LIMIT		分解された各要素の終りの欄番号、BUF(K)の要素に対応するものがCOLNO(K)にある。
CODE		2 0		FORMAT文で許される型コード、1015, 8A4等、数字で始まり途中にアルファベットを含む要素はエラーとなるが、CODEに登録された文字については型指定コードとみなす。 110, A4等は、許されず1110, 1A4と書く。'△△△1', '△△△A'等1字で与える。
CARDNO			0	読込済カード枚数カウンタ、自動カウント

仮 引 数	型	寸法	初期値	内 容
CUNT °LIMIT BUF IBUF EBUF ENDRG ERR °MODE INPUT OUTPUT			1	<p>分解済要素格納位置カウンタ, 自動カウント BUF, COLNOの次元 (LBUFを含め主プログラムでEQUIVALENCEをかけておく) 分解バッファ, 寸法はLIMITである。単語のときはBUFに, 整数, 論理定数のときはIBUF(1,*)に, 実定数のときはEBUF(1,*)に入り, (2,*)に識別コードが入る。</p> <p>特殊コードはIBUF(2,*)に入る。 命令文の終了記号が[]のとき0 [;]のとき1が入る。</p> <p>1回の分解でエラーが発生のときエラー番号が入る。</p> <p>通常1, 0を入れたときはソース・カードのINPUTは行なわず, 外部でよみてむ。 ソース入力ファイルの参照番号[通常 5] 分解結果出力ファイルの参照番号 0とすれば出力しない。</p>

整 定 数	-I0-	複 素 数	CE0-	文 字 列	-C0-
実 定 数	-E0-	倍精度複素数	CD0-	レ ー ベ ル	-L0-
論 理 定 数	LG0-	倍 精 度 実 数	-D0-	特 殊 記 号	-#--

構文解析ルーティン FMCP

構文記述言語の基礎構造

応用プログラム言語の命令文は, 一般にその文法規約を, あらかじめ種類の定められた有限個の要素をつかって, バックス・ノルムまたはコボル記法を用いて, 定義することができる。たとえば, 次のような, 整数名の内容をプリンタ出力する命令文を考える。

WRITE INTEGER, I, J, K :

ここに、I, J, Kは、出力すべき変数名であるから、当然、既定義でなければならぬ。

この命令文の書き方を、コボル記法で定義する。Iはこのシステムで利用可能な要素としての、整数を示す記号とし、複数個を用いるときは、I 1, I 2 I 3, ……等と示すこととする。不必要なら省略してもよい部分を〔 〕で囲って示すこととすれば、

WRITE INTEGER, I 1 [, I 2 [, …… [, I 8]]]

と書くことができる。I 8は、この命令文では、整数名を最大8個まで指定できることを意味する。

この命令文は、はじめに V10CUT によって構成要素に分解すれば、2段階に分けて容易に翻訳できる。

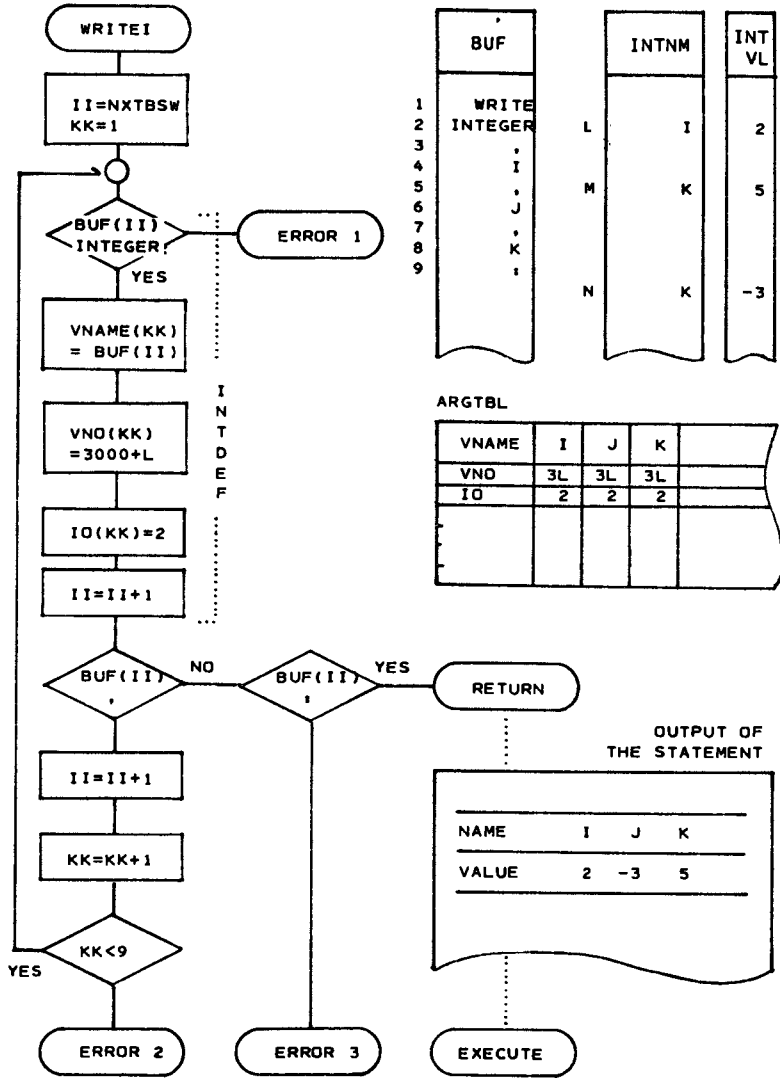
まず、処理すべき命令文が、どの命令文であるかを判別しなければならぬ。この判別は、FCMD ルーティンの任務である。判別が出来ると、その情報にもとづいて、特化された命令文、ここではWRITE INTEGER 文の、翻訳ルーティンに分岐し、それを実行することになる。

翻訳ルーティンは、次に掲げたフロー・チャートのようになる。命令文を分解した結果は BUF に入れられている。このうち命令自体は、第1語、第2語にあって、命令種類の判別に用いられた。翻訳はこの場合、第4語から始められるが、その開始番地は、命令判別時に、NXTBSWに納められている。これをIIとする。

BUF (II) には、整変数名が書かれているものと期待される。この名称は既定義でなければならぬ。整変数名の登録表はINTNMテーブルである。BUF(II)を用いて INTNM 表を走査し、第L語目に同一の名称を得る。もし同一名称がなかった場合はエラーである。

翻訳結果は、一旦、引数表 ARG TBL に転送される。このカウンタを KK とする。引数表の第1列には、変数名 I に関する情報が入る。この場合には、変数名と、その変数の所在を示す変数番号と、入出力指定、つまり、この変数は入力すべき変数か、出力すべき変数か、の3つの情報で充分である。

TRANSLATION OF 'WRITE INTEGER'



変数名は BUF(II) または、INTNM(L) に入っている。これを、引数表の VNAME(KK)に入れる。変数番号は、整数表の L 番目であるが、整数表であることを示すため 3000を加え、それをVNO(KK)へ入れる。入出力指定は、後に詳述するが、共通領域から実行ルーティンへの出力であって、2を入れることになる。I/O指定は、IORG(KK) = 2となる。これで、変数名 I の翻訳は終る。

添数がついている場合には、続いて添数の処理が行なわれるが、ここでは困るを避けるために、添数はないものとし、添数処理の説明は後にゆずる。

II に 1 を加えて次の要素に移る。次の要素は特殊コード [,] の筈であるが、もし 1 変数だけが指定されていたのなら、命令文の終了コード [:] が書かれているかも知れない。[,], [:] のいづれでもなければ、文法で指定する書き方に違反しているのであるからエラーである。この場合は [,] であって、次にまだ変数名が書かれている。II に 1 を加え次の要素に進む、次には ARG_TBL の格納位置もちがってくるので KK にも 1 を加える。ただし KK は、文法規約により 8 以上であってはならない。

こうしてふたたび整数名の翻訳にかえる。この過程をくりかえせば ARG_TBL の内容が、最後には 図に示したようになることは、容易に理解されよう。この ARG_TBL の内容によって、I, J, K の内容をプリントするプログラムを作製することは容易であろう。

さて、変数名の以上の翻訳プログラム、つまり、与えられた変数名を、整数表と対比して、必要なパラメーターを求め、引数表の所定位置に格納するルーティンは、

```
SUBROUTINE INTDEF (—, —, —, ..., *)
```

として与えられている。この引数については今の所は説明を省略しておこう。エラーの場合の飛先は、引数に含まれている。

特殊記号の判別も、サブルーティンで行うものとし、それを

```
SUBROUTINE SEP ('CODE', *)
```

とする。'CODE'は、文字定数4文字で、第2字目に判別すべき特殊記号を書くものとする。[,]の判別は、

```
CALL SEP ('A, AA', &150)
```

によってなされる。対象要素が[,]であれば、先の例ではIIを1上げて、次の命令文に移り、ちがえば文番号150の命令に移る。

エラー・メッセージも多用するのでサブルーティン化しておこう。

```
SUBROUTINE ERROR (*, 'MESSAGE')
```

'MESSAGE'には、60字以内で必要なメッセージを書き、文の終りに [@を書くこととする。終れば [*]で示された番地にもどる。

以上のサブルーティンを用いて、翻訳過程をFORTRANで書けば、下に示すようなものとなるであろう。これに加えて、特定の命令文とこのプログラムとを連結する処理が要る。

下の右に示したのは、左側のFORTRANプログラムと同じ機能をもつように設計された、インタプリタ・システムのためのプログラムである。翻訳に必要な各種の標準化されたルーティンが整備されたら、一々FORTRANその他の言語をつかってプログラムを書かなくても、インタプリタによって、順にそれらのルーティンを稼動させ、同じ機能を果させることができる。

FORTRAN PROGRAM

```

SUBROUTINE WRITEI
C
DO 100 KK = 1,8
CALL INTDEF (&300)
CALL SEP (' , ', &150)
100 CONTINUE
GO TO 310
C
150 CALL SEP (' : ', &320)
200 RETURN
C
-----
300 CALL ERROR (&200, 'MESSAGE')
310 CALL ERROR (&200, 'MESSAGE')
320 CALL ERROR (&200, 'MESSAGE')
END

```

FMCP PROGRAM

	PG				WRITEI
*	DO	KK	1	8	
	I	2	E1	KK	
	SP	,	LL		
KK	DT				
	J	E0			
*					
LL	ID	:	E2		
ED	//				
*					
E0	ER	ED			MESSAGE
E1	ER	ED			MESSAGE
E2	ER	ED			MESSAGE
	\$\$				

構文記述・解析言語 FMCP は、このような観点から設計された、コンパイラ創出用のインタプリタ言語である。

FMCP プログラムの第 2 行目は、命令コードを示している。

PG 命令によって、このプログラムは、命令表に登録された WRITEI という命令文に連絡し、そのコンパイル・ルーティンを提供する。

DO 命令は、DT 命令と結んで、DO ループを形成する。

I 命令は、対象となる要素を整数名とみなし翻訳処理を行う。

SP 命令は、対象となる特殊記号の種類を判別する。

J は分岐命令である。

ID は、特殊な意味をもつ要素（ここでは終了記号）を判別し、必要な処理を行う。

ER は、エラーメッセージを出し、システムにエラー発生を登録する。

// は、FORTRAN の RETURN 文に相当する。

\$\$ は、FORTRAN の END 文に相当する。

左右のプログラムを対照すれば、FMCP プログラムの内容も容易に理解できると思う。

以上の限りでは、FMCP 言語は、単なる FORTRAN の置きかえであってさして簡易化されたものとも言えない。しかし、左側の FORTRAN プログラムで、WRITE INTEGER の命令文入力について、会話的に利用者を誘導して、全くの初心者でも、容易に FOIL のプログラムが書ける様に、豊富な説明用メッセージの出力と、段階的な入力を認めるプログラムを書こうとしたら、極めて繁雑ものになることに、直ちに気付くであろう。FMCP は、バッチと会話モードの TS 処理と処理しうる翻訳ルーティンを、一つのインタプリタタイプな言語によるプログラムで、作製しうる便宜を提供する。システムの作製担当者は、この言語によって、会話のためのわずらわしいプログラムミングから解放される。

FMCP の命令

FMCP で構文の記述を行なうため命令は、次の四群にわかれる。

1. FMCP翻訳制御命令
2. プログラム制御命令
3. 基本要素翻訳命令
4. 変数名翻訳命令

FMCP 翻訳制御命令は、FMCP プログラムの始めと終り、命令表に記載された命令名称と、FMCP によって作製される構文表との連結等を行うもので

P G G T // \$ \$

の4命令からなる。定数、パラメータを設定するための命令も、便宜的にこの群に含まれる。

プログラム制御命令は、分岐したり、処理をくりかえしたり、ユーザの判断を求める質問を発したりするための命令よりなる。これらは、命令文の翻訳過程を制御したり、誘導したりするための命令で、これらの命令の実行結果は、コンパイラの出力にならない。

基本要素翻訳命令は、BASIC サブ・システムに属し、どのサブ・システムにとっても、命令文体系を構成するのに必要な諸要素

レーベル, 整数, 実数, 単語, 文字列 および キー・ワード

等の翻訳を行う命令群である、翻訳結果は引数表に出力される。

変数名翻訳命令は、各サブ・システムごとに、特有の命令セットが準備される。そのサブ・システムのデータ構造に対応して含まれている各種の変数の変数名を翻訳する。

以上の四群の命令をつかって、命令文の構文を記述する。プログラムは次に掲げる、FMCPコーディング・シートに記入する。コメントを含めて95ステップ、実効65ステップ以内でなければならない。このプログラムはカードに穿孔されて、FOILプログラムのライブラリアン命令

R E G I S T E R S Y N T A X (short-name) :

のデータとして与えられる。翻訳された構文表は、構文表ファイル(SYNFIL)に格納される。

FMCPプログラムは次掲のプログラム・シートに記入される。各欄の記入内

容を以下に説明する。

LOC この命令の相対番地を、英字2字以内（または英字と数字）で記入する。'△A'と'A△'とはちがうものとして処理されるから注意を要する。

INS FMCPの命令コードを記入する。命令コードは英字2字または1字より成るが、1字の場合は必ず'△I'のように、後につめて書く。

OPE 通常は命令のオペレータを書く、命令によっては他の要素が入ることもあり、各命令ごとに規定する。

NXT 通常はエラー時の飛先番地を書く。命令によって他の要素が入ることもあり、各命令ごとに規定する。

STO 翻訳結果を引数表に記入するときの格納位置を指定する。他の要素が入ることもあり、各命令毎に規定する。

PARAM 整定数、または、文字定数を書く。整定数を書いた場合は、STO欄に書くのと同じである。STO欄には2桁までの数なら書けるが、それ以上の桁数になるときは、この欄に書く。文字定数を書いたときは、NXT、STO欄に分けられて4字宛を格納する。

MESSAGE 会話時、メッセージを伴う命令の場合この欄に記入されたメッセージが、会話時にディスプレイされる。初め8字に何も記入しないと無視される。最初の4字が、'*△△△'の場合も同様である。

この欄の最初の1字が[#]であるときは、既出のメッセージと同じメッセージであることを示す。[#]に続いて英文字2字を書いた場合には、同一プログラム上の既出のメッセージとみなして、LOC欄を走査し、対応する位置に示されたメッセージを自命令のメッセージとする。[#]につづいて数字が書かれているときは、メッセージ・ファイル上のレコード番号とみなし、他のプログラムで既にファイルに登録されているメッセージを自命令のメッセージとする。既登録のすべてのメッセージとそのレコード番号とは、ライブラリアン命令文によってその一覧表をうることができる。

バッチ・システムの場合、ER命令文の場合を除きこの欄は無視される。

COMMENT LOC欄が'*△'であるとき、その一行はすべてコメントとみ

なされ、翻訳の際無視される。

この章の終りに、FMCP 命令の一覧表を掲げた。各命令についての詳細説明は続いて与えられるが、ここでは、記法の一般的な説明を行っておく。

INS欄に〔*〕が付してある命令は、時分割処理の場合にのみ有効で、一括処理時には機能しない。これらは、時分割時に会話処理を効果的に行なえるようにする補助命令である。同じくINS欄に〔°〕印を付した命令は、この命令の他に、追加的に引数を与える必要があることを示す。たとえば、実配列Fを指定する場合には、その実配列の次元が同時に与えられねばならないが、これを追加命令によって与えるものと規約される。引数追加にはNP命令を用いることとする。実配列について言えば、

F I O E R L
N P I

の2命令で与える。後述するように、IOに入出力指定、ERはエラー時の飛先〔飛先の命令のLOC欄に記入される相対番地名〕、Lは翻訳結果を格納すべき引数表上の位置を示す。Iは、この実配列の寸法を示し、整数名または整数で書く。与え方の詳細は各命令ごとに説明する。

OPE, NXT, STO欄に〔==〕がある場合には、記入してはならず、記入すれば無効となる。この位置は、FMCP 内部での目的のために使用される。

〔—〕がある場合には、FMCP内部では使用しない空番である。プログラムで自由に使用してよい。〔—〕がある場合、この位置には、書かれた文字コードがそのまま入ることを意味する。NXT, STO欄が共に〔—〕であるときは、PARAM欄の8字分の文字コードが、4字ずつに分割されて入れられる。

I, J, Kには、通常は整数を書くが、変数を書いてもよい。変数には相対番地名を書き、その名称をLOC欄に付けてある命令のSTO部の内容がとられる。

Lで示した個所には、その命令の翻訳結果を、引数表の何番目に格納するか

F M C P INSTRUCTIONS

NO.	CODE	INS	OPE	NXT	STD	CONTENTS
1	38	PG	N1	---	---	PROGRAM IDENTIFICATION
2	31	GT*	N1	N2	--	GET STATEMENT
3	49	//	--	--	--	RETURN
4	50	\$\$	==	==	==	END
5	03	C	K	---	---	PARAMETER CONSTANTS (NON-EXEC)
6	00	NP	--	--	--	NO OPERATION (NON-EXEC)
----- CONTROLE INSTRUCTIONS -----						
7	47	BT	M	N1	--	STORE BSW TO STO(M)
8	48	BL	M	N1	--	LOAD BSW FROM STO(M)
9	37	MD	I	N1	K	SET K ON MODE(I)
10	02	B*	N1	N2	--	BRANCH ON 'YES' OR 'NO'
11	17	O*	N1	N2	--	BRANCH ON ' ? ' OR '??'
12	33	MS*	K	N1	--	WRITE MESSAGES K-LINES
13	30	ER	N1	---	---	ERROR MESSAGE
14	10	J	N1	---	---	UNCONDITIONALY JUMP TO N1
15	28	DD	M	I	J	DO LOOP THROUGH M FROM I TO J
16	29	DT	==	==	==	TAIL OF THE DO LOOP
17	27	AD	M	N1	K	ADD K TO STO(M)
18	45	SB	M	N1	K	SUBTRACT K FROM STO(M)
19	32	ID	---	ER	--	IDENTIFY ELEMENT
20	46	SP	---	ER	--	IDENTIFY SPECIAL CODE
21	47	XD	ER	---	---	IDENTIFY SPECIAL WORD
----- BASIC / RECORD ELEMENTS -----						
22	11	K	M	ER	L	KEY WORD
23	12	L	--	ER	L	LABEL
24	20	T	K	ER	L	TIME PERIOD OR SAMPLE RANGE
25	51	RC	IO	ER	L	(51 RC) RECORD ELEMENT GENERAL
26	09	I	IO	ER	L	(40 RI) INTEGER SCALAR
27	14	N°	IO	ER	L	(41 RN°) INTEGER ARRAY
28	05	E	IO	ER	L	(42 RE) REAL SCALAR
29	06	F°	IO	ER	L	(43 RF°) REAL ARRAY
30	23	W	IO	ER	L	(44 RW) WORD
31	01	A°	IO	ER	L	(39 RA°) CHARACTER STRING
----- GROUPED DATA / MATRIX -----						
32	07	G°	IO	ER	L	GROUPED VARIABLE NAME
33	08	H°	IO	ER	L	GROUPED HEADER NAME
34	22	V°	IO	ER	L	VECTOR
35	24	X°	IO	ER	L	MATRIX
----- TIME SERIES VARIABLES -----						
36	19	S	IO	ER	L	TIME SERIES DATA
37	18	R°	IO	ER	L	TIME SERIES ARRAY
----- MODEL VARIABLES -----						
38	13	M	IO	ER	L	MODEL VARIABLE GENERAL
39	34	MN	IO	ER	L	ENDOGENOUS VARIABLE
40	35	MT	IO	ER	L	TEMPORALY VARIABLE
41	36	MX	IO	ER	L	EXOGENOUS VARIABLE
(*) THIS INSTRUCTION HAS MORE ARGUMENT(S) [NP -- J K]						
(*) EFFECTIVE WHEN TS-MODE I, J, K INTEGER PARAMETER						
== NO EFFECT / NOT USABLE N1, N2 NEXT ADDRESS						
-- NO EFFECT / USABLE M REFERING ADDRESS						
___ CHARACTER PARAMETER L ARGUMENT STORE LOC						
IO IN / OUT ASSIGNMENT ER ERROR RETURN						

を指定する。通常は整数を書くが、変数を書いてもよい。変数の書き方は前と同じである。

N 1, N 2には、その命令の機能に対応する飛先の相対番地名を書く。Mは、その命令で参照すべき番地名を示す。ERは、その命令によって翻訳される過程で、エラーが発見された場合〔たとえば、既にレーベルとして用いられたものと同じ名称が、整数名として用いられた場合〕の飛先である。

I Oには、入出力指定を書く。たとえば、

READ INTEGER, I, J, K:

において、この命令文にとっては、変数 I, J, Kは、入力すべきものであり

WRITE INTEGER, I, J, K:

において、変数 I, J, Kは、この命令文が出力すべき変数である。I O欄には、整数で

入力の場合 1

出力の場合 2

を入れる。さらに、たとえば、READ INTEGER 命令文において、I, J, Kは、今まで一度もプログラムに現れておらず、ここで始めて現れてもよい。すなわち、いままで未定義であってもよく、未定義の場合は、ここで定義される。このような場合には、

未定義なら定義しうる場合 + 1 0

とする。

翻訳制御命令

(1) **PROGRAM IDENTIFICATION**

3 8 P G N 1 — — P A R A M M E S S A G E

FOIL の翻訳ルーティンと、特定の命令文についての FMCP プログラムとの連絡をつける命令である。FMCP プログラムの翻訳時には、この命令の PARAM 欄に書かれた命令文の簡略名によって、命令表が参照され、命令表

上の位置に対応して、翻訳された FMCP プログラムが、構文ファイルに格納される。命令文の翻訳時には、命令部の翻訳結果と、PARAM 欄の内容とが対照されて、妥当な構文表であるかどうかチェックされる。

N 1 OPE欄に相対番地名を書くと、この命令の実行後、その番地にとぶ。OPE欄が空白のままなら、次の命令に移る。

PARAM FOIL の命令表に登録されている、命令文の簡略名を、後につめて書く。これより、命令文と構文表との連絡が行なわれる。

MESSAGE 通常、命令文の名称(FULL NAME)を書く。TSモードの場合は、まず、この命令によって命令文の名称が示され、利用者の入力しようとする命令文であることを確認させる。

1つの FMCP プログラムの先頭には、必ず、この命令がなければならない。もし、同一の FMCP プログラムが、複数の命令文の翻訳を行う場合には、エントリの個数に等しいPG命令を、エントリ番号の順に、連続して書くこととし、N1で各エントリに対応する翻訳プログラムの開始番地を指定するものとする。この場合、PARAM, MESSAGE は、重複して書かねばならない。

いま、1つの FMCP プログラムに、整数の入出力命令文と実数の入出力文の4命令文の翻訳構文を一括して格納したいとき、プログラムの先頭は次のようになる。

LOC	INS	OPE	NXT	STO	PARAW	MESSAGE
	PG	RI	--	--	READ I	READ INTEGER
	PG	WI	--	--	WRITE I	WRITE INTEGER
	PG	RR	--	--	REARD	READ REAL
	PG	WR	--	--	WRITER	WRITE REAL
					
					
	*				ENTRY TO READ INTEGER	

```

R I      .....
          .....

*   ENTRY TO WRITE INTEGER
W I      .....
          .....

*   ENTRY TO READ REAL
R R      .....
          .....

*   ENTRY TO WRITE REAL
W R      .....
          .....
          .....
          .....

```

(2) GET STATEMENT

```
* 3 1      G T N 1 N 2  --      MESSAGE
```

この命令はPG命令の次に書くものとする。TSモードの場合にのみ有効でBATCHモードの場合は無視される。MESSAGE 欄には、命令文の命令名称を除いた残り、すなわち、命令記述部の書き方を示すものとする。この命令が実行されると、まず、MESSAGE 欄の内容を端末に出し、つづいて命令文の入力を要求する。PG命令とGT命令とで、命令文の書き方が利用者に示されることになる。

入力内容の先頭が、'?'(以外ブランクを含む)であるときはN1へ飛ぶ。N1, N2が空白であるときは、次番地を示すものとみなす。

'?'でN1に移る場合、N1番地以降にはMS命令を用いて、この命令文についての詳細な説明文を与えて後、N2にかえる規約とする。

MESSAGE 欄の内容を出力して後、次のメッセージを出して入力を要求する。

IF YOU REQUEST DETAILED EXPLANATION
KEY IN '?'
OTHERWISE TYPE IN STATEMENT
DIRECTLY OR ``

(3) RETURN TO FMCP MAIN

49 // --- -- --

FORTRANにおけるRETURNに相当する、FMCPプログラムの理論的な終点を示すもので、1つのプログラム中にいくつ書いてもよい。この命令が実行されると、FMCPによる処理を終了して、汎用翻訳プログラムの主ルーティンにかえる。

(4) FMCP PROGRAM END

50 \$\$ == == ==

FORTRANのEND文に相当する。FMCPプログラムの終りに、必ず1つ、この命令を書く。

(5) PARAMETER CONSTANTS

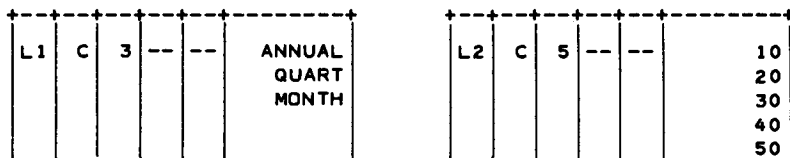
(NON-EXECUTABLE)

03 C K -- PARAM

パラメータ定数を設定する命令である。パラメータとしては、8字以内の単語（先頭は必ず英・カナ文字とする）、整数またはEタイプ実定数が書ける。整数または実定数の場合、正員の符号をつけてもよい。

K 複数の定数をまとめて設定する場合、その個数を書く。

PARAM 定数を記入する。2個以上のときは、次命令以下の同じくPARAM欄に記入する。この場合、必要なければ、Cは書かなくてもよい。



(6) NO OPERATION (NON-EXECUTABLE)

00 NP ---

この命令は機能をもたない。OPE, NXT, STOの各欄に、正の整数をパラメータとして与えるのに使用する。

プログラム制御命令

(7) STORE BSW TO STO (M)

47 BT M N1 --

基本要素翻訳命令、変数名翻命令、および、ID、SP、XWの諸命令は、命令文分解表 (BUF) 上におかれた要素について処理を行うが、処理すべき要素のBUF上の位置はBSW [BUFSW] に入っており、それを自動的に参照する。この命令は、BSWの内容を一時的に退避させるために用いる。

M BSWの内容の退避・格納先の相対番地名。この番地のSTO欄、すなわち、STO(M)に格納される。

N1 この命令実行後の飛び先。空白なら次番地となる。

なお、退避した内容を加工する必要があるときは、AD、SB命令を使用すればよい。

(8) LOAD BSW FROM STO (M)

48 BL M N1 --

BT命令で退避したBSWの内容を、ふたたびBSWにかえす命令である。

- M** BSW の内容の退避・格納先の相対番地名
N 1 この命令の実行後の飛び先

(9) SET K ON MODE (1)

37 MD I N I K

FOILの翻訳は、MODEと名付けられた4個のスイッチで、経路を制御されている。命令文によっては、これらのスイッチの内容を変更する場合が生じて来るので、その機能を果たすための命令である。MODEの機能についての詳細は'FOILV-10モニター'の説明を参照されたい。ここでは、機能の一覧表を示しておく。

- I** 変更すべきMODE スイッチ番号。MODE(I)の内容がこの命令により変更される。
N 1 この命令実行後の飛先。空白ならば次番地。
K MODE(I)に設定すべき値。0, 1~3

内容	MODE(1)			MODE(2)	MODE(3)
	処理時	翻訳終了時	実行終了時		
0	F M C Pによる翻訳	エラーがなければ実行	翻訳	T S時 会話処理	1 命令文づく 翻訳・実行
1	特定ルーティンによる翻訳	強制翻訳	—	T S時 一括処理	翻訳後実行
2	実行	強制実行	強制実行	一括処理	翻訳のみ
3	終了	END 検出	END 検出	T S端末による一括処理	実行のみ

(10) BRANCH ON 'YES' OR 'NO'

* 02 B N 1 N 2 -- MESSAGE

この命令は、T Sモードの場合にのみ機能する。まず、MESSAGE 欄の内

容を出力し、ついで、

KEY IN 'YES' OR 'NO'

とメッセージを出して入力进行を要求する。

入力が ' ' , 'YES' または 'Y' の場合は N1 へ、'NO' または 'N' の場合は N2 へ飛ぶ。N1, N2 は、空白の場合は次番地を意味する。

通常の命令文の入力誘導に必要な会話機能は、基本要素および変数名の翻訳命令のもつ機能で充分満されるよう設計されているので、FMCPプログラムの作製に当って、特に会話についての配慮を行う必要はないと言える。しかし、特にくわしい誘導をプログラムしたい場合には、この B 命令および次に述べる Q 命令によって、多様な会話を実行することができる。

(11) **BRANCH ON '?' OR '??'**

* 17 Q N1 N2 -- MESSAGE

この命令は TS モードの場合のみ機能する。まず、MESSAGE 欄の内容を出力し、ついで、

KEY IN 'NO', '?' OR '??'

とメッセージを出して入力进行を要求する。

入力が ' ' , 'NO' または 'N' の場合は次番地へ、'?' の場合は N1 へ、'??' の場合には N2 へ飛ぶ。N1, N2 は、空白の場合は次番地を意味する。

(12) **WRITE MESSAGES K LINES**

* 33 M S K N1 -- MESSAGE

MESSAGE 欄に書かれた内容を、この命令以下 K 行分出力し、終れば N1 に飛ぶ。第 2 行以下は必要ない限り、命令は書かず、MESSAGE 欄のみを書けばよい。この命令は、TS モードの場合のみ機能する。

P番地とすれば、 $OPE(M)=P$ となる。

DO命令の用法については、次のDT命令を参照。

(16) TAIL OF THE DO LOOP

29 DT == == ==

DO LOOPの終りを示す。この命令のOPE欄、NXT欄、STO欄は、先行するDO命令によって書き込まれる。

この命令が実行されると、STO欄の内容に1が加えられ、それとNXT欄の内容とが比較される。もし、STO欄の内容が、NXT欄の内容より小さいか、又は等しければ、OPE欄の示す番地の次の番地に飛び、STO欄の内容がNXT欄の内容より大きければ、この命令の次の番地の命令に飛ぶ。

いま、例として、実数名を8個以内書きならべる命令文があるとしよう。実数名と実数名と間にはコンマ〔,〕を書く規約とし、これらの翻訳結果は引数表の第3例から第10例までに格納されるものとする。この構文は右に示すようになる。

繰返し回数はJJでカウントすることとし、3から10まで繰返す。JJにはDT命令を書く。実数の翻訳命令はEであり、結果は引数表の第JJ列に入れられる。実数の次にはコンマが来るはずで、これはSP命令によってチェックされる。コンマでない場合は終了記号〔:〕が来るべきであり、そのチェックはID命令によってなされる。これらの順序に命令文が構成されていない場合はエラーであり、ER命令によって、エラー・メッセージが出力されることになる。

E1, E2, , E8 :

	DO	JJ	3	10
	E	11	E1	JJ
	SP	,	ED	
JJ	DT			
ED	ID	:	E2	
ZZ	//			
E1	ER	ZZ		
E2	ER	ZZ		
	\$\$			

(17) ADD K TO STO (M)

27 AD M N I K

M番地のSTO欄, STO(M)にKを加える。Kは整数または、相対番地名で示された番地の STO欄の内容である。すなわち、

$$K \text{ が定数のとき} \quad \text{STO (M)} = \text{STO (M)} + K$$

$$K \text{ 数が変数のとき} \quad \text{STO (M)} = \text{STO (M)} + \text{STO (K)}$$

実行後、N1番地に飛ぶ。Kが定数のときは、負の数を書くことはできない。

(18) SUBTRACT K FROM STO (M)

45 SB M N I K

M番地のSTO欄の内容, STO(M) からKを引く。Kは整数または、相対番地名で示された番地の STO欄の内容である。すなわち、

$$K \text{ が定数のとき} \quad \text{STO (M)} = \text{STO (M)} - K$$

$$K \text{ が変数のとき} \quad \text{STO (M)} = \text{STO (M)} - \text{STO (K)}$$

実行数, N1番地に飛ぶ。Kが定数の場合、負の数を書くことはできない。

(19) IDENTIFY ELEMENT

32 ID -- ER --

BUF(BSW) におかれている要素が、特定の種類のものであるかどうかを判別する。指定した種類のものであれば、次番地の命令に移り、種類がちがえば NXT欄に指定した番地 (ER) に飛ぶ。BSWの内容は変わらない。要素種類の指定は、OPE部に、次のいずれかの記号を、後1字に書く。

: 終了記号 BUF(BSW) の内容が、終了記号 [: 又は ;] であるか否かを判別する。

C 文字列 BUF(BSW) 以下に文字列があるか否かを判別。

I 整数 BUF(BSW) の内容が整数であるか否かを判別

E	実定数	BUF(BSW) の内容が実定数であるか否かを判別
W	単語	BUF(BSW) の内容が単語であるか否かを判別。

(20) IDENTIFY SPAECIAL CODE

46 SP _ _ ER _ _

BUF(BSW) におかれている要素が、指定された特殊記号であるかどうかを判別する。指定された特殊記号と同じものであれば、BSW の内容に 1 を加えて、次の命令に移り、指定されたものでなければ、BSW の内容はそのままにして NTX 欄に書かれた番地 (ER) に飛ぶ。判別すべき特殊記号の指定は、OPE 部の後 1 字に書く。OPE 部がブランクの場合は、特殊記号一般を意味することになり、種類はとわずとにかく特殊記号であれば、BSW の内容に 1 を加えて次の命令に移る。

(21) IDENTIFY SPECIAL WORD

47 X D ER _ _ _ _ P A R A M

BUF(BSW)にある要素が、PARAM 欄に書かれた単語であるかどうかを判別する。PARAM 欄には 8 字以内の単語を後につめて書く。先頭の 1 字は、英カナでなければならない。BUF(BSW) の内容が、PARAM 欄の内容と一致したときは次番地の命令に移り、ちがっている場合は、ER で指定された番地に飛ぶ。実行後、一致していた場合には、BSW に 1 が加えられる。

命令文基本構成要素処理命令

各サブ・システムについて命令文の体系を考える場合に、若干種類の要素は如何なる計算言語体系にでも、含まれている必要がある。これらを基本構成要素と呼び、その翻訳処理のために次の命令が準備されている。

これらのうち、I, N, E, F, W, A については、後に変数として一括して説明することとし、まず、K, L, T について説明する。

INSTRUCTION CODE	KIND OF ELEMENTS
K	KEY WORD
L	STATEMENT LABEL
T	SAMPLE RANGE OR TIME PERIOD
I	INTEGER SCALAR
N	INTEGER ARRAY
E	REAL SCALAR
F	REAL ARRAY
W	WORD
A	CHARACTER STRING

これら命令は、TSモード場合、実行されるとまず、この命令に書かれたメッセージがあれば、それを出力し、続いて、

KEY IN XXXXX

と出力して、対応する入力を要求する。たとえば、キイ・ワードの場合なら、

KEY IN KEY WORD

と、打ち出して、入力が求められる。

説明には、命令コードに続いて、 $\square\square$ で囲んで1, 2桁の数があり、それから書き方になるが、この数は、その要素の種類の識別コードであり、以下種類番号と呼ぶ。要素の種類別に、共通領域にそれぞれの定義表があり、翻訳処理は結局は、これらの諸要素ごとの定義表を参照し、処理対象がその何番目に定義されているかを走査して、そこに書かれている情報を取り出したり、または必要に応じて定義表に書込みを行ったりする操作に他ならない。得られた情報の中、命令文に固有の部分は、引数表に書きこまれ、1つの命令文の含む情報は引数表の完成という形にまとられる。実行ルーティンは、この引数表に入れられた情報と、共通領域にある他の各種情報とを参照して作動することになる。

引数表には、32個までの要素に関する情報を格納しうる。1要素に関する情

ARGUMENT TABLE ARGTBL(32,8)

	1	2	3					30	31	32
VNAME										
VNO										
IORG										
LAG										
SUF										
SUF2										
VAL										

報は、7つの欄に分解されて格納される。

VNAME欄 通常、BUF(BSW)の内容がそのまま転送される。実行ルーティンにおいて、変数名の出力に利用できる。

VNO欄 定義表上の位置をN、翻訳対象の種類番号をMとすると、
 $M * 1000 + N$

が書きこまれる。対象が定数である場合は、 $N = 0$

IORG欄 この要素が、実行ルーティンに対する入力であるか、出力であるかを指定する。FMCP命令の入出力指定の内容がそのまま転送される。

LAG欄 時系列データ名の場合、タイム・ラグを示す添数が入る。

SUF欄 第1添数の情報が格納される。

SUF2欄 第2添数の情報が格納される。

VAL欄 予備。実数定義がなされている。整数の格納に用いたいときは、IVALとして用いればよい。

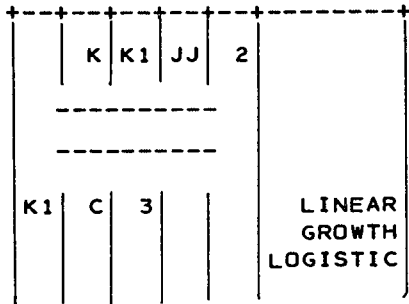
以下の命令において、STO欄には、1~32を書くことによって、対象要素の翻訳結果を、引数表のどの位置に入れるかを自由に指定できる。この格納を、命令文に書かれる順番に従って行う必要はない。さらに、1命令文の翻訳開始に先立って、引数表は初期化されるが、前の命令文が〔;〕で終わった場合、すなわち、同種の命令文が連続して書かれている場合には、この初期化を、1から25までについてだけ行い、26から32までは、前命令文のものが、そ

のまま保存される。したがって同一命令文が連続して書かれることが予想される場合、各命令文に共通して指定しなければならない要素が予想されるときは、その要素の翻訳結果格納位置を、26以下にしておけば、記法の省略ができるようになる。もちろん、連続した命令文で26番目以降に対応する要素について新しい情報が与えられたら、同じ欄へ書き込まれ、前命令文で与えたものは消去されるから、命令文の書き方の自由度は損なわれない。

(22) KEY WORD

11/9/ K M E R L MESSAGE

BUF(BSW)の内容が、M番地以降に格納されているキー・ワードのいずれかと同じであるかを調べる。同じものがあれば引数表の第L列にその情報を格納し、BSWを1上げて次の命令に進む。同じものがなければ、ER番地に飛ぶ。



いま、右の例で、K1番地以降にはパラメータ定数設定命令Cを用いて、3個のキー・ワードが入れているとする。BUF(BSW)には、単語'GROWTH'が入っていたとしよう。

K命令が実行されると、K1番地のOPE部の内容=3、によって、K1番地以降3個のPARAN部の内容がBUF(BSW)の内容と比較され、2番目の

ものが一致する。単語'GROWTH'は、引数表第2例のVNAME欄に入り、VNO欄には、9000に2を加えた値が入る。引数表の他の欄は、この命令によ

ARGTBL(2)

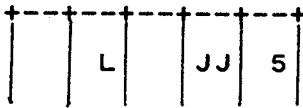
VNAME	GROWTH
VNO	9002

る翻訳には無関係である。

以上の過程でエラーが発見されなければ次命令に移り、途中で何等かのエラーが発見されたときは、ER番地の命令に移る。

(2 3) L A B E L

1 2 / 1 / L - - E R L M E S S A G E



ARGTBL (5)

VNAME

VNO

BACK
1012

BUF(BSW) の内容が、単語か又は整数であることを確認し、整数である場合は文番号であるものとみなす。これ以外の要素であればエラーとなる。BUF(BSW) の内容でレーベル定義表を走査して、引数表への書込みに必要な情報をうる。BSW の内容には 1 が加えられる。

いま、BUF(BSW) の内容が、`BACK` であり、右の命令によって、レーベル定義表を走査して、その12番目に登録されていたとすると、引数表の第5列の内容は左図のようになる。

対応するレーベルが定義表に登録されていないときは、一括処理の場合にはエラーとなり、TSモードの場合は警告メッセージを出して、そのレーベル名または文番号を定義表に仮登録する。

エラーの場合には、OPE 欄に指定された番地 (ER, 例示した命令では、JJ) へ飛ぶ。

(2 4) T I M E P E R I O D O R S A M P L E R A N G E

2 0 / 2 / T K E R L

この命令は標本範囲の指定に用いるもので、有効範囲 (ACTUAL SIZE) を変更する機能をもっている。BUF(BSW) が整定数であることを確認する。整定数でなければエラーとなる。

K = 1 サンプルの始期 (または、処理開始番号) を設定する。

K = 2 サンプルの終期 (または、処理終了番号) を設定する。

L 引数表上への格納位置

TSモードの場合は、標本範囲を制御する BASIC 表は、この命令によって書きかえられる。その際、時系列データであって、年次以外の場合には下2桁を期または月、3桁目以上を年とみなして変換する。入力内容と BASIC 表の内容とが斉合性を欠く場合にはエラーとなる。

BUF(BSW) の内容は、そのまま、引数表の VNAME 欄に入る。IVAL 欄には、入力内容に対応する添数の値が計算して入れられる。すなわち、全観測期間が、61年1期から始まる四半期データと指定されているとき、BUF (BSW) の内容が、6202であり、K = 1 であれば、有効期間は62年2期からであると判別され、VNAME 欄には、整数6202が識別コードを伴って入れられ、IVAL の値は6となる。期または月はSUF欄に、年はLAG欄に入る。時系列でないかまたは年次系列であるときは、対象である整定数の値が、そのまま SUF 欄に入る。

エラーの場合は、メッセージを出してERに飛ぶ。

変数名処理命令

変数名処理命令は、BUF(BSW) にある単語 (または定数) を、その命令の指定する種類の変数の名称 (又は定数) とみなして翻訳を行う。TSモードの場合は、命令が実行されると、まず、その命令にメッセージがつけられているときは、そのメッセージを出力し、続いて

```

または      KEY   IN   XXXXX
              TYPE  IN   XXXXX NAME
  
```

の出力を行って、その種類の変数名または定数の入力を要求する。

たとえば、I、X命令の場合、それぞれ

KEY IN INTEGER

TYPE IN MATRIX NAME

と出力され、入力が要求される。

変数名処理命令は、命令コードを一般にXXで示すこととすれば、右に示すいずれかの形になる。

既に説明したように、IOは入出力指定で、1、2、11のいずれかである。ERはエラー時の飛先で、エラー・メッセージはすべての変数名処理命令に内蔵されているので、エラー・メ

	XX	IO	ER	L
	XX NP	IO	ER I	L J
	XX NP	IO	ER	L J

ッセージ出力の処理は不要である。Lは引数表への格納位置指定で、1～32の値をとる。I、Jは配列の寸法を指定する値で、1次元配列の場合は、Jのみを書く。3次元以上の配列は現在の所認められていない。

説明は、まず、番号および名称につづいて、内部コードが書かれるが、それに続いて、//で囲んで種類番号が与えられる。種類番号は、引数表の中のVNO(L)に、その変数の格納位置が与えられているとき、付加される種類識別番号で、格納位置にこの種類番号に1000を掛けた値を加えて、VNO(L)の位とする。たとえば、VNO(L)が、3005であるとするれば、対応要素は整数名であり、整数名定義表の第5番目に登録されており、実行時には、その整数の値がそこに与えられているか、または、その位置に転送される。

スカラーを指定する命令に対して配列名が翻訳対象であった場合には、その配列の中の1つの要素を指すものとみなす。この場合、当然添数がついていなければならないが、添数の翻訳結果はSUF(L)に入るが、添数のついていない場合は、添数が<1>であるものとみなす。この場合、配列の種類は、指定されたスカラーと同じ性格のものでなければならないが、VNO(L)に入る種類番号としては、スカラー種類番号がつけられる。

配列の場合は、その寸法を検査する必要があるとき、次番地にNP命令を書

いて、STO部とNXT部とを用いて示す規約とする。次にNP命令が書かれていなければ、寸法を検査する必要がなく、その配列名が変数の種類に対応する定義表上に定義されていればよく、そこに指定されている配列の全体の寸法を翻訳結果の寸法とする。

BASIC サブ・システムに属する基本変数として、I, N, E, F, W, Aがあるが、レコード・データを取扱う場合の変数、RI, RN, RE, RF, RW, RAと処理は同じであり便宜上、同様の説明を行っておく。基本変数以外は、原則として、1つのサブ・システムの言語体系の中で、データの性質のちがうサブ・システムを対象とする要素の処理命令を、混用することは出来ない。基本変数に関する定義表は、どのサブ・システムにも必ずつけられるが、その他の変数の定義表は、システム・シェネレーション時に、用いる変数の範囲が決定され、特定の変数種類に関する定義表だけが、システム共通領域に含まれている。

基本変数名処理命令を除いて、各サブ・システムごとの処理命令の機能詳細は、定義表の構造、および、作業領域へのデータ・アクセスの一般原則と密接は関係しているので、これを各サブ・システムの構造説明にゆずることとする。

基本変数およびレコード変数の処理命令

(25) RECORD ELEMENT (GENERAL) RECORD

51 RC IO ER L MESSAGE

BUF(BSW)にある要素が単語であることを確認する。単語でなければエラーとなる。レコード・データ名定義表を走査して対応するデータ名を見出し、引数表に登録する。対象要素には添数がついていてもよい。未定義の場合にはエラーとなる。したがって、IOは1、または、2でなければならない。

引数表の第L列には、VNAME欄に要素名が、VNO欄にその要素の定義表上の位置が入る。文字列以外の配列名で添数を伴う場合は、同種のスカラー

とみなされ、その種類番号に1000を掛けた値が VNO 欄の値に加えられる。添数は SUF 欄に入る。添数を伴わないときは、配列の種類番号に1000を掛けた値がVNO 欄に加えられる。I O欄には、I Oの値をそのまま転送する。

エラーの場合はNXT部に示された番地に飛ぶ。

(26) I N T E G E R S C A L A R B A S I C / R E C O R D

09 / 3 /	I O E R L	MESSAGE
40 / 13 /	R I O E R L	MESSAGE

BUF(BSW)の要素が、単語であることを確認する。(ただし、Iの場合は整数であってもよい)その他の場合はエラーとなる。定義表上の対応するデータ名を見出し引数表上のSTO部で指定された列Lに格納する。VNAME欄には、BUF(BSW)の内容がそのまま入り、VNO欄には定義表上の位置が入る。添数のついているときは、SUF欄に入れられる。定数の場合はVNO欄は0となる。

定義表上で配列である場合。添数がなければ、添数を<1>とみなす。種類番号(3または13)に1000を掛けた値が、VNO欄に加えられる。I O欄にはI Oの値がそのまま転送される。未定義の場合、I O>10なら、定義がなされ定義表上の症欄にBUF(BSW)の内容が登録される。

エラーの場合は、NXT部に示された番地に飛ぶ。

(27) I N T E G E R A R R A Y B A S I C / R E C O R D

14 / 4 /	N I O E R L	MESSAGE
	NP	K
41 / 14 /	R N I O E R L	MESSAGE
	NP	K

BUF(BSW)にある単語を整配列名として処理する。Kに必要個数を書く。I Oが11の場合は、未定義ならば、K個の配列を定義する。既定義の場合、指定された個数だけ、定義されていない場合はエラーとなる。Kの値はIVAL欄に入る。NP命令を欠く時は、既定義の配列の大きさが入る。

いま、BUF(BSW)に、'CITYNO'という変数が書かれ、その変数が整数定義表の25番目に書かれていたとする。個数指定が、K=30であったとすると、

VNAME (L) = 'CITYNO',

VNO (L) = 4025, 又は

VNO (L) = 14025,

I VAL (L) = 30

となる。

VNAME	CITYNO
VNO	4025
I VAL	30

(28) REAL SCALAR BASIC/RECORD

05 / 5 /	E I O E R L	MESSAGE
42 / 15 /	R E I O E R L	MESSAGE

BUF(BSW)にある単語を実スカラー名として処理する。(ただし、Eの場合は実定数であってもよい)。実配列の要素であっても差支えなく、その場合には、添数がついていなければ、添数が<1>であるものとみなす。

既述の整数の処理を参照されたい。

(29) REAL ARRAY BASIC/RECORD

06 / 6 /	F I O E R L	MESSAGE
	NP	K
43 / 16 /	R F I O E R L	MESSAGE
	NP	K

BUF(BSW)にある単語を実配列名とみなして処理する。Kにその個数を書く。IOが11の場合、未定義ならK個の配列を定義する。既定義の場合、定義されていない場合はエラーとなる。Kの値はIVAL欄に入る。NP命令を欠くときは、既定義でなければならず、その配列全体の大きさが入る。

既出の整配列の処理を参照されたい。

(30) WORD BASIC/RECORD

23 / 7 /	W I O E R L	MESSAGE
44 / 17 /	RW I O E R L	MESSAGE

単語は、基本変数の場合と、レコード・データ変数の場合とで取扱い方がちがう。

基本変数として処理するW命令です。BUF(BSW)の内容が単語であるか否かを調べ、単語であれば、そのまま、VNAME(L)に転送し、VNO(L)=7000として処理を終る。BSWの内容には1が加えられる。IOは参照されない。

レコード・データ変数として処理するRW命令では、単語名を意味し、他のスカラーと同様の取扱いを受ける。8字1語として取扱われる他は、取扱いは全く同様である。文字列の要素であってもよい。この場合、添数は8字1語で単語配列が形成されているものとして計算され転送される。

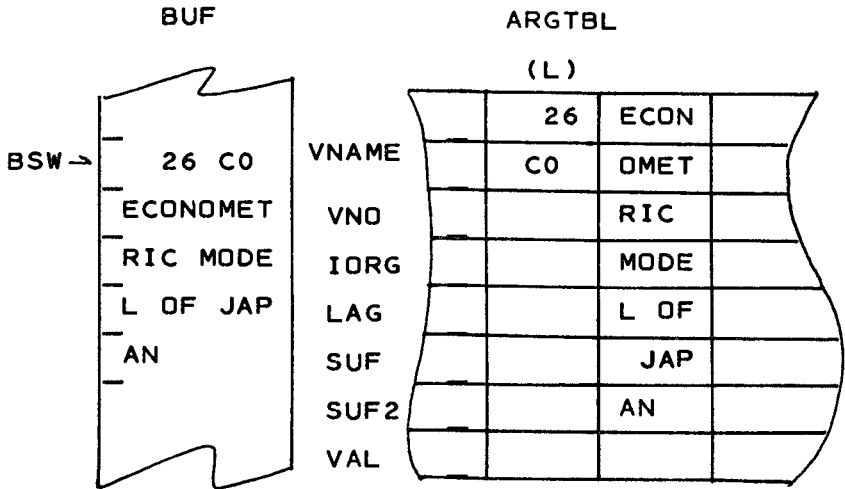
(31) CHARACTER STRING BASIC/RECORD

01 / 8 /	A I O E R L	MESSAGE
	NP	K
39 / 18 /	RA I O E R L	MESSAGE
	NP	K

文字列の場合は、基本変数と、レコード・データとで、その取扱いはちがったものになる。

基本変数Aの場合は、命令文に文字列が定数として書かれている場合に限られる。文字列は分解過程に於て、先頭に字数および識別コード〔'△C0△'〕があり、以下8字づつBUFの中に入められている。翻訳過程でA命令が実行されると、まず、BUF(BSW)に字数および文字列識別コードがあるか調べられ、あれば、BUF(BSW)の内容はそのままVNAME(L)に転送されて、VNO(L)は8000が入えられる。Kには最大字数が入り(NP命令を欠くときはK=64とする)、字数がこれより大きいときは、下の方から消される。与えられた文字列は、引数表の(L+1)列以下に、図に示すような形に転送され

'ECONOMETRIC MODEL OF JAPAN'



る。1列に32字づつなので、2列以上にわたることもあるから、STO部の操作に注意を要する。実行ルーティンでこれを使用するとは、VNAME欄は8バイト、他は4バイトで、各欄は相互に連続していないので、転送・編集に注意しなければならない。

レコード・データに関するRA命令は、他の数値変数の配列の場合と同じで、文字列名、または単語配列名として処理する。Kには字数を書く。レコード上には、8字を1語として、格納場所で定義される。入出力の取扱いは、数値の配列データと同様で、添数のつく場合は、8字1語で計算され、転送される。

グループ変数名，行列名処理命令

(3 2) G R O U P E D V A R I A B L E N A M E G R O U P

0 7 / 1 2 / G I O E R L M E S S A G E
 N P K

グループ変数は，レコード・データを取扱うとき，レコードの一部で，しかもかなり多数の変数を一括して取扱う場合が多い。繰返して個個の変数名をつかって命令文を書いてゆく代りに，グループ変数名を定め，その名称でレコード変数の部分集合を定義して，これを命令文中で使用する。

グループ変数は実行時には，異なる種類の構成要素をもつベクトルとして取扱われる。グループ変数定義表と対比して処理される他は，配列の取扱いと同じである。Kには，共通領域内の格納場所1，2，3のいずれかを指定する。Kの値はIVAL(L)に入る。

(3 3) G R O U P E D H E A D E R N A M E G R O U P

0 8 / 1 9 / H I O E R L M E S S A G E
 N P K

グループ・ヘッダーは，大きな行列や2元配置表を取扱うとき，その各行・各列の名称を一括して与える為のものである。1個当り16字で32個以内を1つのヘッダー名で定義できる。Kには共通領域上での格納場所，1，2，3のいずれかを書く。取扱いはグループ変数名と同じである。

(3 4) V E C T O R G R O U P

2 2 / 2 1 / V I O E R L M E S S A G E
 N P K

ベクトルは，行列名定義表で処理される。行列の特定の行または列を指定してもよいが，ベクトルは必ず，列ベクトルとみなされる。ベクトル名には添数をつけることはできない。要素はすべて実数である。

(37) TIME SERIES ARRAY

T. S.

```
18 / 32 /      R I O E R L      MESSAGE
                NP          K
```

BUF(BSW)にある要素を時系列配列名とみなし、時系列変数定義表によって処理する。Kには列数を示すものとし、値がIVAL(L)に入る。添数をつけてはならない。

IOの値が51または52のときは、システム・データ・ベース上との間の入出力を意味し、BUF(BSW+1)の内容が[@]のときも、BUF(BSW)の内容は、システム・データ・ベース上の変数を指すものとみなされる。種類番号は1511である。

モデル変数名処理命令

(38) MODEL VARIABLE (GENERAL) MODEL

```
13      M I O E R L      MESSAGE
```

BUF(BSW)の内容は何等かの種類の、モデル変数名であるとみなされ、モデル変数名定義表によって処理される。IO=11のとき、その変数名が既定義であれば、その種類番号をとり、未定義であれば TEMPORALY VARIABLE とみなし、種類番号は1431とみなされる。

添数がついている場合、第1添数はタイム・ラグとみなされ、LAG(L)にその処理結果が入り、第2添数はSUF(L)に入れられる。

(39) ENDOGENEOUS VARIABLE MODEL

```
34 / 41 /      MN I O E R L      MESSAGE
```

BUF(BSW)の内容を内生変数とみなし、定義表により処理する。IOの内容は1、または、2でなければならず、11を用いることはできない。

添数はN命令の場合と同じである。

(40) TEMPORARY VARIABLE MODEL**35/43/ MT IO ER L MESSAGE**

BUF(BSW)の内容を一時変数(モデルの計算途中結果その他を入れるための、モデルにとって内生・外生のいずれに属するか明確に規定しえない変数)とみなして処理する。

添数はM命令の場合と同じである。

(41) EXOGENEOUS VARIABLE MODEL**36/42/ MX IO ER L MESSAGE**

BUF(BSW)の内容を外生変数とみなして処理する。I, Oの内容は1, 2でなければならず, 11を用いることはできない。

添数はM命令の場合と同じである。

FMCP の構造と機能**FMCP 言語の翻訳ルーティン FMCPTR**

FMCP言語で書かれたソース・プログラム・カードを読んで翻訳し、構文表を創出するルーティンとして、FMCPTR が準備されている。このルーティンは、BASIC サブ・システムに組込まれており、次の命令文を書くことによって作動する。

REGISTER SYNTAX (short-name) :

構文表の登録は、ライブラリアンの機能に属する。従って、この命令を実行するには、まず、ライブラリアンが呼び出されねばならず、かつ、この命令で構文表を登録する前に、命令文表に命令の名称その他を登録しておかなければならない。これらの登録手順の詳細については、後節で説明する。

命令文の登録のためのFOIL プログラムの後に、そのプログラムのデータとして、FMCPのプログラム・カードをセットする。プログラムの実行過程で、

REGISTER SYNTAX 命令文が来ると、FMCP のプログラム・カードが読みこまれ、翻訳されて、構文表が創出される。

REGISTER SYNTAX 命令文によって、構文表の創出に加えて、命令表の一部も書きかえられる。書きかえられた命令表は、LIBRARIAN END: 命令文によって、命令表ファイルを更新するので、ライブラリアンの機能とは独立な命令として書いても、動かない場合もあるし、動いた場合でも、命令表の内容が一時的に書きかえられただけで、次のジョブ以後には無効となる。反対に単にテストしたい場合には、LIBRARIAN END : を書かずにおけば、命令表は変更されない。

FMCP 翻訳ルーティンは、次のような機能をもつ。

REGISTER SYNTAX 命令文の制御番号が、

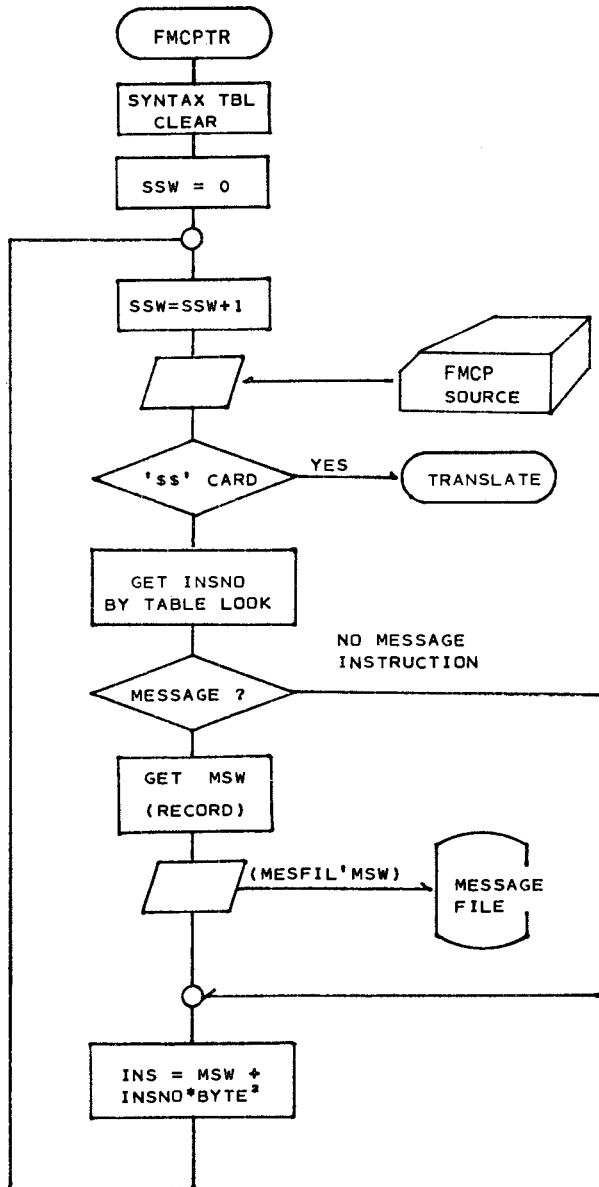
- 0 のとき、新規登録（既登録ならエラーとなる）
- 1 のとき、既登録の構文の修正
- 2 のとき、仮テスト、構文表および命令表は書きかえられない。
既登録であっても、なくてもよい。

STEP 1 最初に、SYNTBL テーブル、および LTBL テーブル、PARAM テーブルが初期化される。このうち SYNTBL テーブルは、共通領域には、SYNTAX に対応する。SSW でインデクシングを行う。

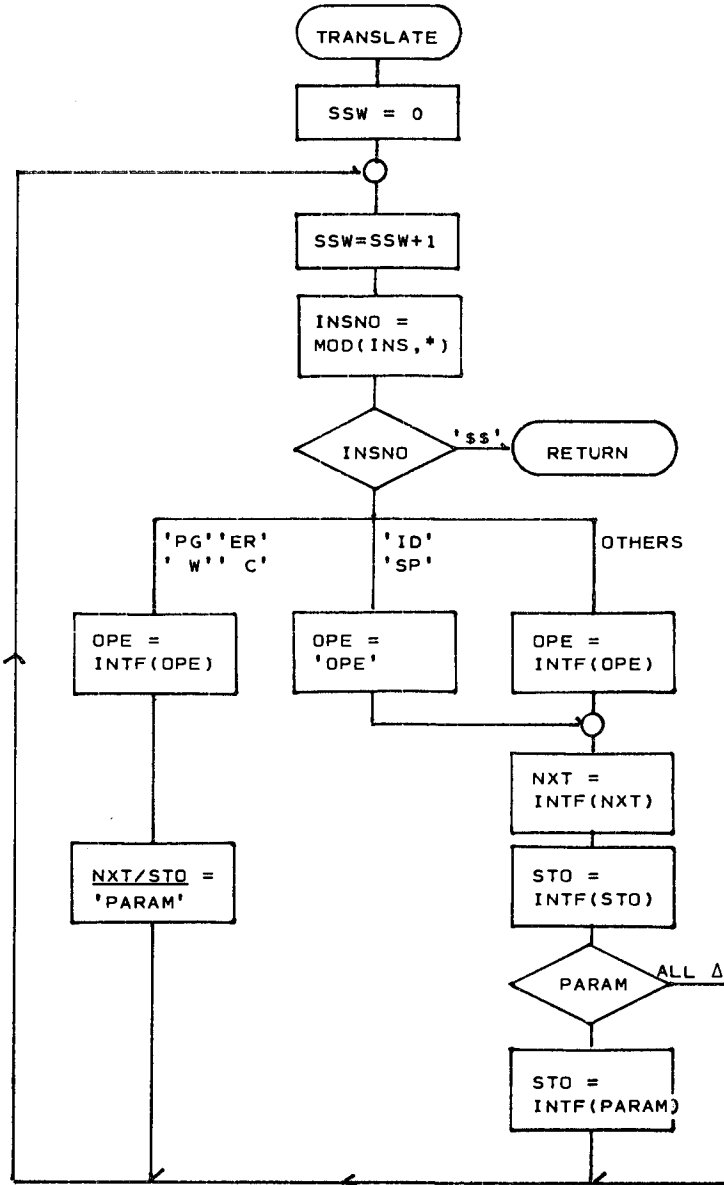
STEP 2 FMCP 命令文カードをよみ、各部分を対応するテーブル上の位置に、文字のまま収容する。全カードを一括してよみこむ。

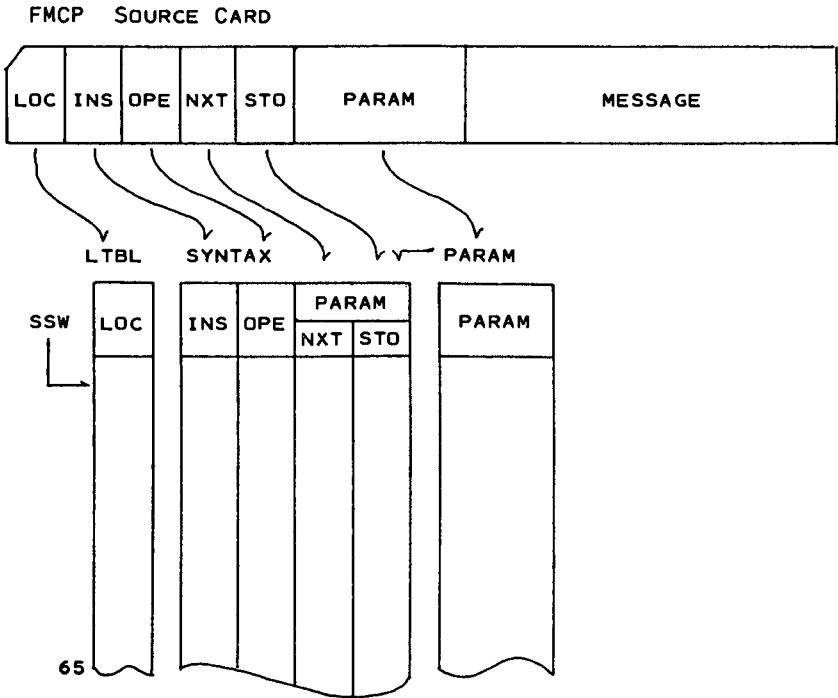
STEP 3 SSW をリセットして翻訳に入る、LOC 欄が '*△' であるときは無視する。INS 欄をしらべて命令の種類を判別する。メッセージを伴う命令であれば、MESSAG 欄を、メッセージ・ファイル(MESFIL)に書出す。メッセージ・ファイルはランダム・アクセス・ファイルであり、そのレコード・カウントにはMSW をつかう。ただし、制御番号が2つであるときはこの書き出しは行なわない。また、MESSAG 欄の最初の4文字が、'*△△△' であるときも、コメントとみなして書き出しを行なわない。

FMCP TRANSLATION ROUTINE 'FMCPTR'



FMCP TRANSLATION ROUTINE 'FMCPTR' (CONTINUED)





INS の前の 2 バイトに命令番号が，後バイトに MSW の値が入る。

メッセージを伴わない場合は，後 2 バイトに 0 が入る。

STEP 4 カードのINS欄が '\$ \$ ' であるときは，FMCPプログラムは終了したと判別される。FMCPプログラムは 1 命令文あたりの構文記述を，コメントを除き '\$ \$ ' を含めて 65 行まで書くことができる。

STEP 5 OPE, NXT, STO 欄の翻訳は，命令文の種類によってちがう。フロー・チャートについて承知されたい。翻訳は主として，次のルーティンを用いて行なわれる。

INTEGER FUNCTION INTG (COLMNN, LTBL)

EXAMPLE : OPE (SSW) = INTG (OPE (SSW), LTBL)

このルーティンはまず，指定された欄の内容が，アルファ・カナか整数

かを判別する。整数であれば2進変換を行い、その値を出力する。ブランクの場合は0となる。アルファ・カナであれば、その単語に対応する単語が、LTBL上にあるか否かをしらべる。なければエラーである。対応するものがあれば、その位置の値(添数)に4000を加えたものが出力される。

STEP 6 PARAM 欄は、もし単語であれば(最初の1字がアルファ・カナであれば)4バイト宛に分割して、NXT,STO欄に格納される。その他の場合は、整定数とみなして、2進変換を行い、ST 欄に入れられる。NXT欄は変らない。PARAM 欄がブランクのときは、このステップは実行されない。(フローチャート上では、便宜上この処理をINTHと名付けているが、サブルーティン化されていない)

以上で1行の翻訳処理を終り、STEP 3にかえつて'\$ \$'命令が来るまで翻訳をくりかえす。

翻訳が終了し、翻訳過程でエラーがなかったら、制御番号が0, 1であれば構文表がファイルに書き出される。さらに、プログラムの先頭の'PG'命令をしらべ、そのPARAM欄と、命令表の short-name 欄が比較され、同一のものがあれば、その翻訳コマンドの最後の2パイに、構文表のレコード番号を入れる。この処理は複数の'PG'命令が書かれているときは、すべてのものについて繰返えされる。

以上で全処理が完了し、入力・出力結果をリストして終る。

次に掲げたリストは、FMCPTR の出力である。プログラムに間違いがあり、エラー・メッセージのでているものと、それを訂正したものについて示した。右側に入力された FMCP プログラムが、左側に翻訳結果が示されている。翻訳結果中、OPE, NXT, STO欄に、4000台の数が示されているのは、相対番地[MOD(*, 4000)]である。たとえば、最初の命令の4003は、3番地、すなわちJ 0を意味している。

F M C P PROGRAM LIST

(LSM) LEAST SQUARES METHOD										SYSNO = 0 PRGRND = 0 SYMFL NO. = 0		
NO	INS	OPF	NXT	STO	LOC	INS	OPF	NXT	STO	PARAM	FILE	MESSAGE
1												
2	38140031					PG	J0			LSM1		LEAST SQUARES METHODD
3	31140001	01				PG				NCLSM1		LSM WITHOUT CONSTANT
4	461	(140071	01		J1	SP	(J2			(TEXT) YEST = Y(X1, ... , X16),COEF IN A :
5	01		1140351	11		I	01	E1	1			EXTRAPOLATE PERIOD
6	461)	140341	01		SP)	E0				
7	101		12140361	21		J2	S	12	E2	2		EXTIMATED RESULTS
8	461	=	140341	01		SP	=	E0				DEPENDENT VARIABLE
9	101		1140361	31		S	01	E2	3			
10	461	(140341	01		SP	(E0				
11	101		1140361	41		S	01	E2	4			INDEPENDENT VARIABLE
12	2R140151		51	701		DN	JJ	5	20			
13	461	,	140161	01		SP	,	J5				
14	101		114036140151			S	01	E2	JJ			INDEPENDENT VARIABLE IF FINISHED TYPE #
15	201	01	01	01		JJ	DT					
16	461)	140341	01		J3	SP)	E0			
17	45140151		01	31		SR	JJ			31		CALCULATE NUMBER OF COEFFICIENTS
18	2R140261		21	261		DD	11	21	26			
19	461	,	140271	01		SP	,	J4				
20	111403014037140261					K	K	E3	11			COEF TVAL PARAM. IF NO NEEDS TYPE #
21	27140261		01	11		AD	11			11		
22	461	%	140341	01		SP	%	E0				
ERRPR ##### 0, 0 UNOFFICED NAME												
23	61		12140391	01		F	12	E4	11			LOCATION TO STORE PARAMETERS
24	61		1214039140261			F	12	E4	11			LOCATION TO STORE PARAMETERS
25	0140151		01	01		NP	JJ					
26	201	01	01	01		11	DT					
27	321	:	140341	01		J4	1D	:	E0			
28	491	01	01	01		J5	//					
FRPR ##### 0, 0 NEXT INSTRUCTION IS ILLEGAL												
29	491	01	01	01		KY	K	E3	11			COEF TVAL PARAM IF NO NEEDS TYPE #
30	271	31	01	01		K	W	3				
31	01	01								COEF1		13
32	01	01								TVAL1		14
33	01	01								PARAM1		15
34	30140281					E0	EP	J5				ILLEGAL SEPARATOR
35	30140281					F1	ER	J5				
36	30140281					E2	ER	J5				
37	30140281					F3	ER	J5				
38	30140281					F4	ER	J5				
ERRPR ##### 0, 0 UNDEFINFD NAME												
39	501	01				F5	ER	J6				
40	31		2140041	01		M	MS	02	J1			LEAST SQUARES METHOD IS A TECHNIQUE TO
41	01	01	01	01								ESTIMATE FUNCTIONAL RELATIONSHIP BETWFFN

NO	INS	OPR	EXT	STP	LOG	INS	OPR	EXT	STP	PARAM	FILL	MESSAGE
1												
1	49140031					PG				LSM	1	LEAST SQUARES METHOD
2	391	01				PG				WLSM	2	LSM WITHOUT CONSTANT
3	31140361					J0	GT				3	(TEXT) YEST = Y(X1, ... , X16), COEF IN A :
4	461	(140071	01		J1	SP		J2		4	
5	91		1140321	11		I	01	F1	1		5	EXTRAPOLATE PERIOD
6	461)	140311	01		SP					6	
7	191		12140331	21		J2	S	12	E2	2	7	ESTIMATED RFSUL
8	461	=	140311	01		SP	=				8	
9	191		1140331	31		S	01	F2	3		9	DEPENDENT VARIABLE
10	461	(140311	01		SP	(10	
11	191		1140331	41		S	01	F2	4		11	INDEPENDENT VARIABLE
12	29140151			51	201	DD	JJ	5	20		12	
13	461	,	140161	01		SP	,		J3		13	
14	191		114033140151			S	01	E2	JJ		14	INDEPENDENT VARIABLE IF FINISHED TYPE *
15	291	01	01	01		JJ	DT				15	
16	461)	140311	01		J3	SP)			16	
17	45140151			31		SP	J1			31	17	CALCULATE NUMBER OF COEFFICIENTS
18	29140251			21	261	DD	II	21	26		18	
19	461	,	140261	01		SP	,		J4		19	
20	111402814033140251					K	K	E3	II		20	COEF TVAL PARAM IF NO NEEDS TYPE *
21	27140251			11		AD	II			11	21	
22	461	*	140311	01		SP	*				22	
23	61		1214035140251			F	12	E4	II		23	LOCATION TO STORE PARAMETERS
24	3140151			01		MP	JJ				24	
25	291	01	01	01		II	DT				25	
26	321	:	140311	01		J4	IO	:			26	
27	491	01	01	01		J5	//				27	
28	231	31				K	W	3			28	
29	01	01									29	
30	01	01									30	
31	30140271			LSM1		E0	FR	J5		LSM1	31	ILLEGAL SFPARATER
32	30140271			LSM1		E1	FR	J5		LSM1	32	THIS ELEMENT MUST BE INTFGR
33	30140271			LSM1		F2	EQ	J5		LSM1	33	THIS ELEMENT MUST BE TIME SERIES DATA
34	30140271			LSM1		F3	EQ	J5		LSM1	34	ILLEGAL KEY WORD
35	30140271			LSM1		E4	EQ	J5		LSM1	35	THIS ELEMENT MUST BE REAL ARRAY
36	531	3140041		01		M	MS	*	J1		36	LEAST SQUARES METHOD IS A TECHNIQUE TO
37	01	01	01	01							37	ESTIMATE FUNCTIONAL RELATIONSHIP BETWEEN
38	01	01	01	01							38	

FMCPの構造

FMCPは、SYNTAX 表をソース・プログラムとして作動する、インタプリタタイプ・サブシステムである。入力された命令文を判別して、それに対応する翻訳プログラム、すなわち SXNTAX 表が、ファイルからロードされ、FMCP にコントロールが引渡される。命令文の翻訳は、FMCP が、数字コードで表現された SYNTAX表の内容を、インタープリンターとして解析して、処理ルーティンを作動させることによって行なわれる。

次頁には、FMCPモニターのフロー・チャートを示した、各命令文の翻訳コマンドを解析して、そのうち、

SYNTAX 表レコード番号（処理型式番号PATH）によって、

SYNTAX 表はロードされるが、処理が FMCP モニターにわたると、さらに、翻訳コマンドから、エン트리番号 ENT を求めて、 $SSW = ENT - 1$ をセットする。SSWは、SYNTAX 表の処理順位カウンターである。

主ループに入ると、SSW に 1 が加えられる。FMCP の

プログラミング規約として先頭は必ず、PG命令であり、複数のエントリーをもつ場合は、連続してPG命令を書き OPE 部にその開始番地を書くこととしている。右の図において、 $ENT = 3$ の場合は、最初に PROG3 についての PG命令が実行されることとなる。

この処理は、INS (SSW) をとり出してそれを分解することからはじまる。INS (SSW) の前 2 バイトには、命令コード、後 2 バイトには、この命令にメッセージが伴っている場合、そのレコード番号が入っている。メッセージを

LOC	INS	OPE	NXT	STO	PARAM
	PG	N1			PROG1
	PG	N2			PROG2
	PG	N3			PROG3
	PG	N4			PROG4
N1					

N3		(ENTRY TO PROG3)			

伴わない場合は0である。一括処理時には、メッセージ・レコード番号は無視される。

命令コードINSTは、命令別処理ルーティンへの分岐パラメータとなり、

GO TO (1 , 2 , 3 , ……) , INST

によって、対応する処理ルーティンへと分岐する。

PG命令の主機能は、FMCPプログラムの翻訳に当って、対応する命令表の書きかえ（SYNTAXファイルのレコード番号の登録）を行なうことであり、実行時には、メッセージ・ファイルをディスプレイするだけである。この命令のOPE部には、次に実行すべき命令の番地が書かれており、0であるときは次番地を意味するので、SSWはそのままで、主ループの先頭にかえる。番地が設定されているときには、その絶対番地から1を引いた値をSSWにセットして主ループの先頭にかえることになる。

以上のような過程が、つぎつぎに指定されてゆく番地ごとの命令について実行され、その内容に対応する処理がなされる。終了命令（//）が検出されると、制御はFMCPをはなれて主プログラムに戻る。

FMCPの命令中には、翻訳制御用のものや、FMCPプログラムの経路を指定する制御命令、さらにはタイム・シェアリング時に、ユーザとの会話を構成するための命令等を含んでいるが、これは、最終翻訳結果の出力内容には残されない。これに対して、命令文の構成要素中、区切り記号または区切り語、および終了記号等を除く、他の構成要素、すなわち、

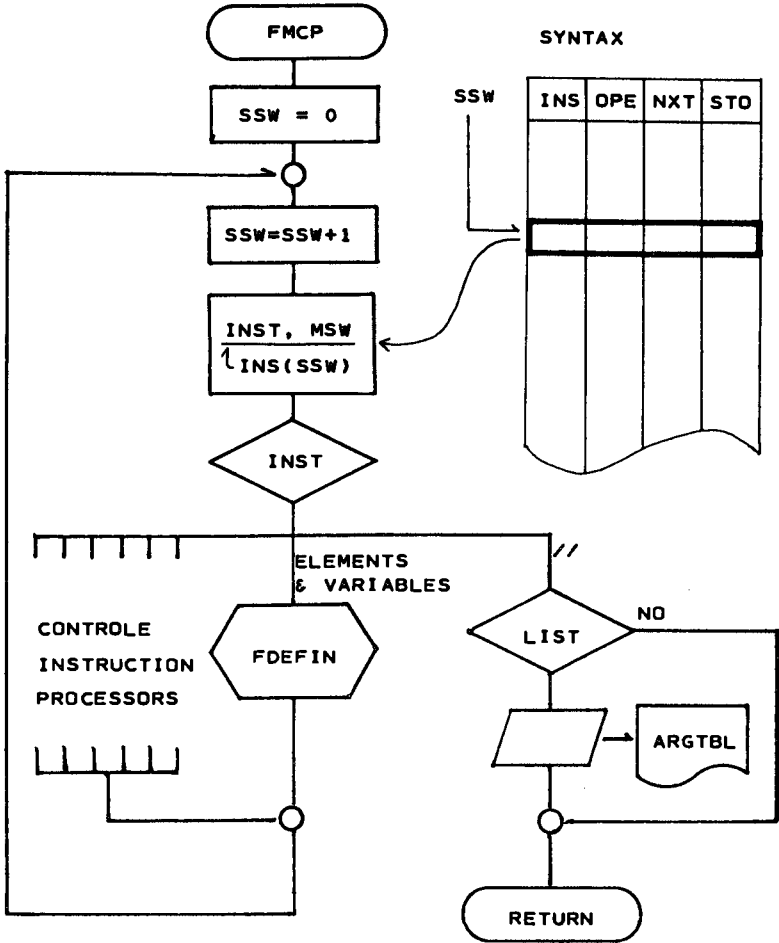
キイ・ワード

各種パラメータ、定数、特定単語

各種変数名

等々は、実行ルーティンへの入力情報となるものであって、最終翻訳結果の出力内容を構成する。かつ、これらを入力する作業は、会話型処理の場合には、システムによる誘導の対象となる。

この誘導を行うために、FDEFINルーティンがあり、上記諸要素の翻訳命令は、一括して一旦、FDEFINルーティンに入り、会話型処理をなすべきか



否かを判別して、もしそうなら、会話処理をして後に、各要素ごとの処理ルーティンに分岐する。その出力は、命令の STO 部に指定されている ARG_TBL 表上の位置に格納される。

FDEFIN については、後節「変数名処理命令実行ルーティン」において詳細に説明することとする。

終了命令 (//, 49) が検出されると、制御番号状況CNTL2 を調べて、CNTL2 \geq 6 (この値は、FOIL END #n: においてnの値として設定される) のときには、ARGTBL 表をリストする。

以上の処理が終ると主プログラムに制御をかえすこととなる。

会話誘導機能

FMCP の重要な特色は、会話形式による命令文入力誘導を、つねに、利用者にとって、必要で且つ充分な程度に行うよう設計されていることである。利用者が命令文の書き方に不明の点があったとき、直ちにシステムの側から、入力すべき要素の種類と順序とについて、明確な指示が与えられるよう設計されていなければならないが、同時に、利用者が必要としないにかかわらず、詳細なガイダンス・メッセージを出しつづけては、かえって冗長感を与え、不便なシステムになってしまう。

FMCP は、この問題を、入力された情報を一旦、命令文分解バッファ BUF に分解して格納し、FMCP の命令をつねに BUF に格納された要素と対象として作動させることによって解決し、常に最適な誘導が、利用者にも与えられるようになっている。

いま、利用者が、システムの利用過程において、重相関分析を行いたいと考えた場合を想定しよう。入力したい命令文は、次のようなものとする。

```

LINEAR REGRESSION,
WATER = (HOUSEHLD, FACTORY, OFFICE,
TIME) COEF:

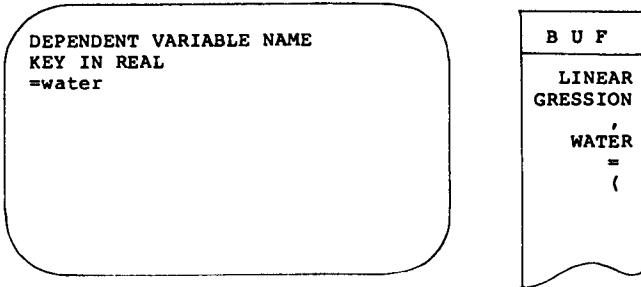
```

ただし、利用者は、自分の行いたい分析、および、分析につかう変数名については、当然のことながら承知しているが、このサブ・システムでの重相関分析命令文の書き方は知らないか、忘れてしまったとしよう。

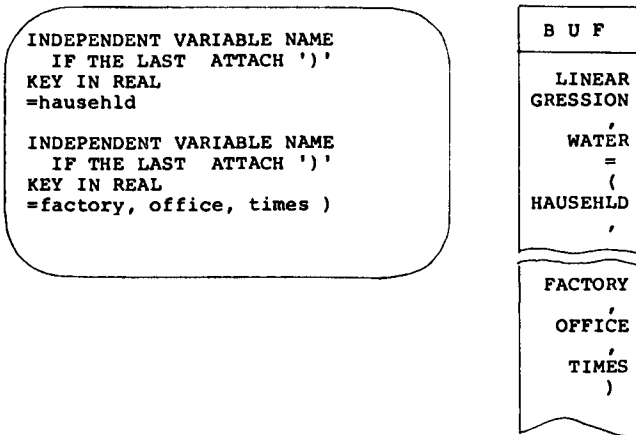
命令そのものについては、最初に入力を要求されたとき '?' を打てば、一覧表が出る。それによって必要な命令文をさがせばよい。命令名称がわかっていて、その後の書き方がわからないときは、命令名称だけをタイプすれば、システムは、PG命令文によって書き方を与えてくれる。

これで十分な知識が得られたなら、全文を入力すればよい。しかし、書き方は記号的に与えられるので、まだわからないというのなら、さらに'?'を入力すればよく、システムは詳細なガイダンスを開始する。

最初に、被説明変数名を与えねばならない。誘導は次のように行なわれる。



両面例示において、大文字はシステムからの誘導メッセージ、小文字は利用者の入力である。

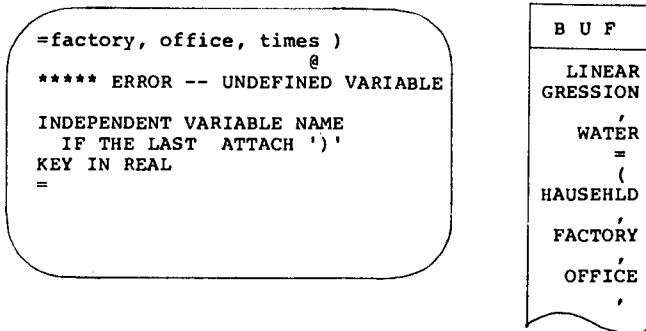


命令文分解バッファ BUF の内容は右図のようになる。まず WATER が BUF に入り、被説明変数名として翻訳処理される。これは妥当な入力なので処理は正常終了する。FMCP は次の要素の処理に移る。→の欄には、'='があもると期待されるが、まだ入力されておらず空白になっている。

FMCPは、この場合、必要な特殊記号を自動的に挿入する。'('についても同様である。次には説明変数名の入力がなければならない。しかし、対応するBUF の欄は空白である。この場合には、次のような誘導が行なわれる。

説明変数名の入力が要求され、終りには')'をつけることが求められる。最初の入力では、終りに')'がついていないから、説明変数名の入力を求めてくる。これに対して書き方はわかったのだから、残るすべての変数名を入力したとしよう。

FACTORY の処理を終り、BUF の次の欄を調べると空白ではないので誘導メッセージの出力も、入力要求もなしに処理は進行する。OFFICE の処理を終り、TIMES に進む。ここで、TIMES としたのは、利用者のタイプ・ミスで、正しい変数名は TIME であったとしよう。この場合は、エラー・メッセージが出され、説明変数名の再入力が必要される。画面は次のようになる。



'@' はエラーの検出された要素を示す。再入力を要求された時点で、BUF の内容は右図のように修正されている。

以上のように、最初の入力要求に対して、命令名称から始まって、終了記号':'に終る命令文の全部を入力してもよく、わからなければ、途中の任意の所

F H C P PROGRAM LIST

(READI)

SYSNO = 0 PRGNO = 0 SYNFIL NO. = 0

NO	INS	OPF	NXT	STO	LOC	INS	OPF	NXT	STO	PARAM	FILE	MESSAGE
1	3	0				PG				READI	0	
2	9	114009				I	OJ	E1	1		0	
3	2	14006	2	9		DO	JJ		2		0	
4	4	6	14007	0		SP		JJ			0	
5	9	114009	14006			I	OJ	E1	JJ		0	
6	2	9	0	0		JJ	DT				0	
7	4	6	14008	0		JL	SP		KK		0	
8	4	9	0	0		KK	//				0	
9	3	0	14008			E1	EP	KK		READI	3	THIS ELEMENT MUST BE INTEGER

CARDNO STAT-NO STATEMENTS
 1 0 READ INTEGER , I , J , K , L , M :

CUT BUFFER CONTENTS

CUTNO = 0 CARNO = 1
 CUNT = 14 ENDMK = 0

10 READ
 19 INTEGER
 21 ,
 23 , I
 25 , J
 27 , K
 31 , L
 33 , M
 35 ,
 37 ,
 39 ,
 42 :

ARGUMENTS TABLE LIST

	1	2	3	4	5
NAME	I	J	K	L	M
VNO	3016	3017	3018	3019	3020
I/O	1	1	1	1	1
LAG	0	0	0	0	0
SUF	0	0	0	0	0
SUF2	0	0	0	0	0
IVAL	0	-0	0	0	0

で入力を中止すればよい。つまり BUF 上に処理対象となる要素が、すでに入力されているかぎり、FMCP は誘導や入力要求を行なうことなく、処理を続行してゆくし、BUF 上に処理対象となる要素が、まだ入力されていないことを検出すると、FMCP は直ちに誘導を開始する、エラーを検出すると、その要素を含む以下の既存入力をすべて無効とし、エラーの検出された要素について誘導メッセージを出して入力を要求する。

変数名ルーティン処理ルーティン FDEFIN

次頁に、FMCPルーティンで処理可能な、変数の種類の一覧表を掲げる。これらの諸要素はFDEFIN ルーティンによって翻訳される。つづけて FDEFIN の流れ図も示しておく。

これらの諸要素は、命令文の実質的な構成要素となるものであって、会話型式で命令文の入力を行うときは、システムが必要に応じて入力すべき要素の種類を、命令文の文法規約に示されている順番によって指示し、正確な入力を誘導することになる。誘導処理のためのプログラムは、各要素を通じて同じ型式の処理でよいので、一旦このルーティンに入ってから、適宜の段階において各要素別のルーティンに分岐して処理を行う。

ルーティンに制御がわたる前に、SYNTAX表上の処理中の命令のINS 部から、コマンド番号 INST と、メッセージ・レコード番号 MSW とが得られている。ルーティンに入ると、まず、TSS であるが、BATCH 処理であるかが判別され、BATCH の場合は誘導のための処理はスキップされる。TSS の場合は、処理対象たるBUF(BSW)の内容をしらべ、処理すべき要素が入力されているかどうかを検する。既入力であれば、直接処理に移るし、BUF(BSW)が空白であれば、誘導処理に入ることになる。

MSW = 0 であれば、要素の性質を説明するためのメッセージはないことを意味する。> 0 の場合は、それがレコード番号であり、メッセージをファイルから入力してディスプレイする。つづいて、各要素に個有のメッセージをディスプレイする。このメッセージは、FMCPの命令の説明に付記してある。たとえば、整数の場合には

F O I L STATEMENTS ELEMENT CODE

CD	INS	CONTENTS	CD	INS	CONTENTS
----- BASIC ELEMENTS -----			RECORD ELEMENTS -----		
1	L	LABEL			
2	T	PERIOD / SAMPLE			
3	I	INTEGER SCALAR (10)	13	RI	
4	N	INTEGER ARRAY	14	RN	
5	E	REAL SCALAR (11)	15	RE	
6	F	REAL ARRAY	16	RF	
7	W	WORD	17	RW	
8	A	CHARACTER STRING	18	RA	
9	K	(KEY) KEY WORD		RC	RECORD ELEMENT GENERAL
----- GROUPED VARIABLES -----			VECTOR AND MATRIX -----		
12	G	GROUPED DATA NAME	21	V	VECTOR
19	H	GROUPED HEADER NAME	22	X	MATRIX
----- TIME SERIES VARIABLES -----					
31	S	TIME SERIES DATA			
32	R	TIME SERIES ARRAY			
----- MODEL VARIABLES -----					
41	MN	ENDGENEDUS VARIABLE	43	MT	TEMPORALY VARIABLE
42	MX	EXOGENEDUS VARIABLE		M	MODEL VARIABLE GENERAL

SEPARATOR WORDS (RESERVED, IDENTIFIER = %)

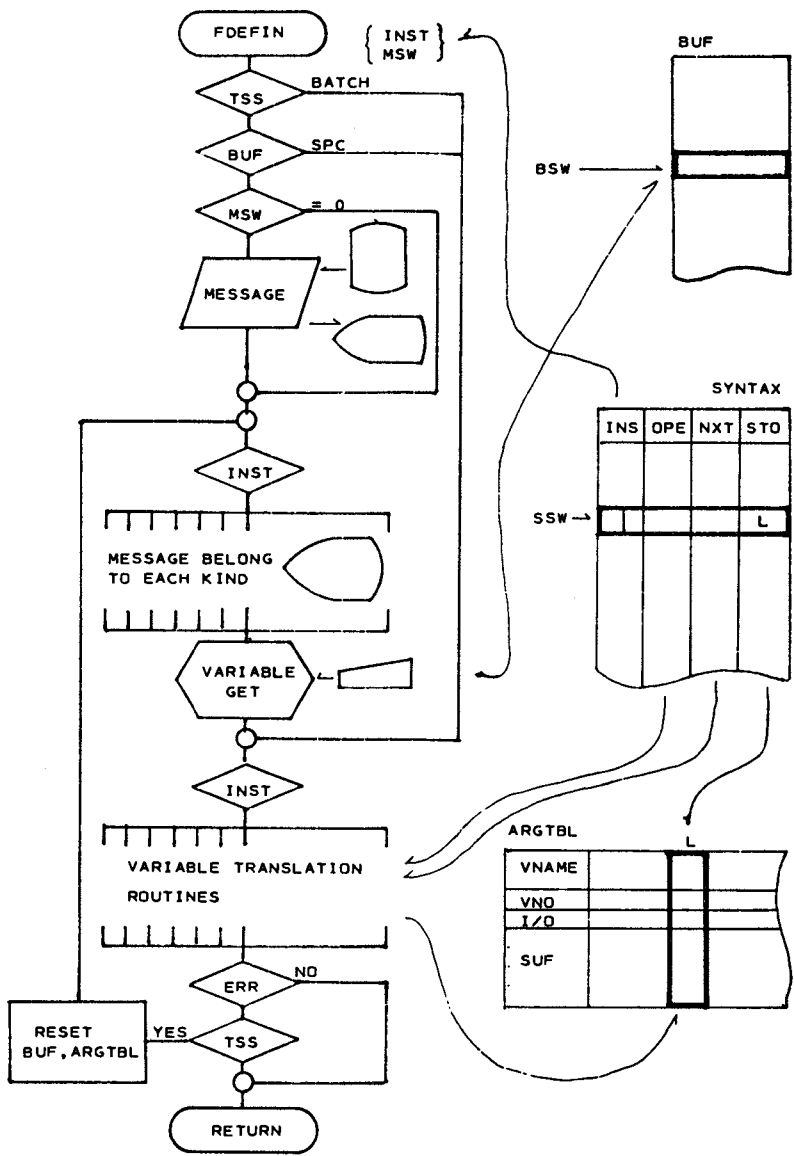
AND, ARE, AS, AT, BE, BY, FROM, IS,
IN, OF, ON, THROUGH, THRU, TO, WITH

RESERVED SPECIAL CODES

(PROHIBITED TO USE FOR ANOTHER PURPOSES)

CONTROLE NUMBER IDENTIFIER
' ' CHARACTER STRING IDENTIFIER
< > SUFFIX IDENTIFIER
: ; STATEMENT TERMINATION CODE

* CD = VARIABLE ID CODE IN THE FOIL OBJECT
 INS = F M C P INSTRUCTION CODE



KEY IN INTEGER

である。

これが終ると入力が必要される。これは VARIABLE GET ルーティンによって行なわれるが、後にくわしく説明する。

入力が行なわれると、BUF(BSW)の内容を対象に、INSTによって分岐して該当する要素として翻訳処理が行なわれる。このとき、SYNTAX表のINS部以外の内容その他がパラメータとして用いられる。翻訳結果はARGTBLのSTO部の内容で指定されている位置に格納される。

翻訳中にエラーが検出されたときは、エラー・メッセージが、各翻訳ルーティンによって出される。TSSであれば、エラーの検出されたBUF上の欄より以下の既入力情報をすべて無効とし、ARGTBLの対応する位置も初期状態にかえて、レジスター、スイッチ類をリセットした上、再入力を要求することとなる。

以上がEDFEINの機能であるが、このルーティンには、既入力部分をディスプレイして訂正する機能を含んでいる。以下この訂正機能を含めて、入力ルーティン VARIABLE GET について説明する。

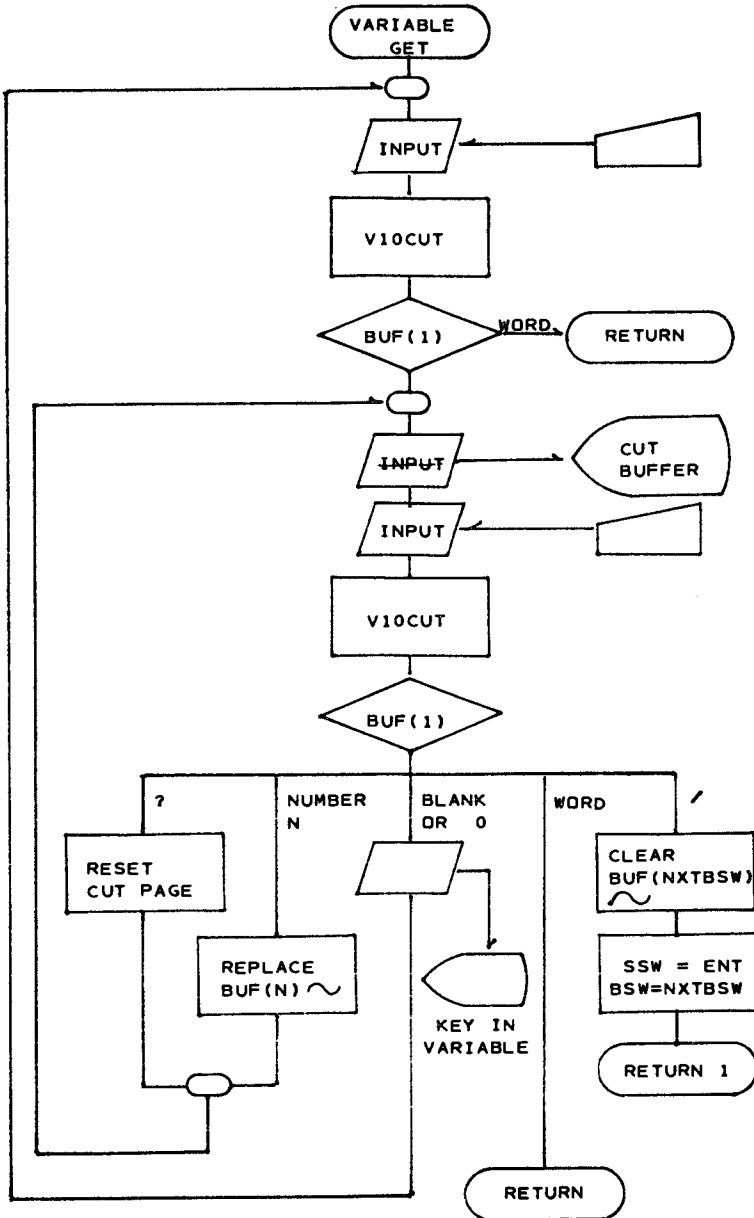
次頁に VARIABLE GET ルーティンのフロー・チャートを示した。このルーティンは、独立したサブルーティンではなく、FDEFINの一部であるが、説明の便宜上切はなした。

入力がなされると、命令文分解ルーティン VI0CUT がよばれて、入力情報を分解する。この際、BUFへの格納開始位置が、BUFSに、格納終了位置に1を加えたもの、つまり、次回の格納開始位置が BUFE に記録される。

入力が '?' でなければ、これで入力処理は終る。 '?' の場合は、BUFの内容を、数頁にわけてディスプレイし、訂正入力を要求する。

訂正入力がブランクまたは0のときは、訂正なしとして、はじめの入力要求にもどる。

整数Nをうち、つづいて任意の入力を行なうと BUF(N)の内容がその入力内容で入れかえられる。



13 TIME

とすれば、BUF(13)が、単語TIMEで入れかえられる。つづいて訂正入力が必要されるから、ひきつづき訂正を行ない、最後はblankを入力すればよい。

演算命令の翻訳と実行

会話型インタプリタティブ・システムと演算命令文

算術代入文、IF文等（以下一括して「演算命令文」とよぶ）の翻訳は、他の命令文とはちがったロジックによってなされる。これらは、命令文の中で記述されている順序以外の、所定の演算優先順位に従って翻訳される。本システムでは、演算優先順位は、論理式、関係式を含めて、FORTRAN 文法規約に準拠する。この翻訳ルーティンは、構文解析ルーティンとは別に、コンパイラに組込まれている。FORTRAN とちがって FOIL の演算命令文は、単にスカラ間の演算のみでなく、時系列間の演算その他の複数の演算式を処理する。時系列に対して適用された場合は、標本期間の全体について同一の演算を行うし、クロス・セクション型のデータ構造に対してアクセスする演算命令文は、レコード数だけ、同一の演算をくりかえす。これらは同一のルーティンで翻訳される。したがって、この演算式解析ルーティンもまた汎用であるといえる。

翻訳の最終結果自体は、実行可能なプログラムではなく、クローズド・タイプ・サブルーティンに対する引数であることを原則とするのが、インタリタティブ・システムである。単位演算プログラムを優先順にしたがって棚積みしてゆく正統的な方法は、インタプリタティブなシステムの構造となじまない。われわれはすでに、棚積み方式による解析法をとったシステムを作製・稼動しているが、翻訳プログラムが繁雑であり、処理速度が遅い上に、解析結果が大きくなり勝ちで、他の命令文の比較的少ない引数と調和しない。

さらに、このシステムは、会話型処理を基本とし、1つの命令文を翻訳して直ちにその内容を実行するという処理をくりかえす。クロス・セクション型のデータ構造とこの処理型式との間には重大な難点がある。いま、一例として、

レコード中に、A, B, C, D, ……等の諸項目があり、これらに、

$$X = A + B$$

$$Y = C / D$$

……………

……………

等の変換を加えて、これらの結果についてある統計方法を適用したいという場合を考えてみよう。

レコード数をmとすれば、これらの処理はm回くりかえされるが、1回ごとにファイルのアクセスを行なわねばならない。もし1命令宛、翻訳実行をくりかえせば、変換のための命令数をkとして、読み書きを含めて $m * k * 2$ 回のアクセスを行うことになる。バッチ処理的に、一回のアクセスで処理できる限り、各命令文ごとの引数を一括しておいて、連続的に変換する方が明かに効率的である。

これら2つの問題点は、会話型のインタプリティブ・システムで、クロス・セクショナルなデータ構造を対象とする場合には、特徴的なものである。

第1の問題のためには、棚積み法によらない演算命令文解析ロジックを開発した。演算命令文も他の命令文と同様に、記入された順序に引数表に格納されてゆく。本システムでは1命令文40個まで変数を格納できるように設計されている。式にあらわれる変数名または定数の個数、左カッコ〔 〕の個数および関数名(ALOG, SQRT等)の個数の合計が40個までの式を翻訳し処理できる。

第2の問題のためには、クロス・セクション・データ・ファイルのアクセスを前提とする命令文については、命令文編集オプションを付しうることにした。命令文編集オプションは、プログラムのメイン・フロー中に書いてよいが、それとは独立のセグメントを形成し、翻訳結果として数式表〔MCMPテーブル〕を生成する。これには個有名称「数式表名」をつける。このセグメントの中では

算術式

IF文

EJECT IF文

GOTO文

だけが用いられる規約とし、特定の命令文によってこのセグメントに入ると、セグメントの終りまでは翻訳のみが行なわれて、実行は行なわれない。翻訳結果は、1枚の数式表に構成され、ファイルに格納される。ある命令文がクロス・セクション・ファイルのアクセスを前提する場合は、その命令文のオプションとして、この数式表の名前を指定すれば、入力データは、所要の修正を受ける。オプションがついていなければ、入力データから必要な項目を取出して、データ変換を行うことなく、その命令文の処理を行うことを意味する。

演算命令文の解析

演算命令文の解析には、4バイト6列の演算解析表がつかわれる。次掲の表によって、棚積み法によらず、どの様に算術文の解析が行なわれるかを説明する。与えられた演算命令文は、あらかじめ分解ルーティンによって分解され、BUFに格納されているものとする。

分解に当っては、演算子、定数または変数名、添数の組が、一つの要素として取扱われ、演算解析表の1行に収納される。なお左括弧〔(〕は、1つの変数とみなして処理される。演算子は算術演算子〔* * * / + -〕、論理演算子〔.AND. .NOT. .OR〕および関係演算子〔.GT. .GE. .LT. .LE. .EQ. .NE.〕と符号〔=〕とが、FORTRANの規約する優先順位によって、その高い程大きい2桁の番号が割当てられている。但し、10以上の桁の値が等しいときは同じ優先順位にあるものとみなされる。左括弧が来ると、それ以後は演算子の番号に100が加えられる、左括弧が2回あれば、200を加える。以下同様である。反対に右括弧の場合は100づつが引かれる。この規則によって、解析終了時には、括弧を示す100以上の値が残っていないはならないということを判別して、括弧が左右対称につけられていたことを確認する。

関数〔SQRT, ALOG, EXP, SIN, COS等〕は、一旦変数名と

して処理されるが、1組の要素の処理が終わった後、関数名表が調べられ、同じ名前があれば、次の要素が左括弧であることを確認して、次の要素の演算子を、その関数とする。

次掲の表は

$$X = (-B + \text{SQRT}(B * B - 4.0 * A * C < K + I / J >)) / (2.0 * A) :$$

という例について、演算命令文の解析経過を示したものである。分解結果は FIG. 1の形で、BUF に格納されている。右側に番号で要素を示した。同一番号のものが集って、演算解析表の1要素(1行)となる。この分解された要素が、FIG.2の演算解析表に転送される。変数名は実際は演算解析表に転送されず、直接VNO欄の変数番号、および、TYPE欄の型番号として変換されてしまうが、ここでは、対照し易いように、一旦、変数名のまま転送するものとみなした、BUFに付した番号と、演算解析表の左側に付した要素番号とを比較しながら、2つの表を見れば、分解結果がどのようにまとめられるかを容易に理解しうるのであろう。適当に+が付加されていることに注意されたい。会話処理の場合、添数に変数名の翻訳過程で直ちに計算され、その翻訳時での値がSUF欄に入る。これは、翻訳して直ちにその内容を実行する会話型の処理に対応している。添数の処理についての詳細は次節で説明する。

転送が終わると、優先順位が調べられる。OPE欄には、優先順位の高い演算子ほど、大きな値が与えられており、かつ、1桁目は、同一優先順位での内容をあらわしている。例えば、+-は優先順位では同じであり、*/も、同様である。しかし、*/の方が+-より優先順位は高い。この場合

$$\begin{array}{ll} + & 101 \\ * & 111 \end{array} \qquad \begin{array}{ll} - & 102 \\ / & 112 \end{array}$$

というように演算子番号がつく。第5要素のように、先行して左括弧が2回あらわれているときは、それぞれ2000が加えられる。OPE欄の値を10で割って、値の大きいもの、同一優先順位の場合は先行するものから、その要素番号を、PRI欄の前から順に格納してゆく。最後に、OPE欄の値を、処理し易いよう

OPE	VNO	VAL									
		INITIAL VAL	6 8	9	7	5	4 12 2	3 11	10	1	
0	X										
1	=+ -										-0.5
2	- B	5.0					-5.0	-2.0	-0.5		---
3	+ -										---
4	y -						3.0				---
5	+ B	5.0	25.0	25.0	9.0	9.0					---
6	* B	5.0	---	---	---	---					---
7	- 4.0	4.0	8.0	16.0	---	---					---
8	* A	2.0	---	---	---	---					---
9	* C	2.0	---	---	---	---					---
10	/ -							4.0			---
11	+ 2.0	2.0					4.0				---
12	* A	2.0					---	---	---		---

```

L = 0
C
100 L = L+1
    M = PRI(L)
    K = OPE(M)
    IF (K .EQ. 0) RETURN
    N = M
110 N = N-1
    IF (VAL(N) .EQ. SPC4) GO TO 110
    GO TO ( 1, 2, 3, ..... ), K
    .....

C ----- ( + )
XXX VAL(N) = VAL(N) + VAL(M)
GO TO 500
    .....

C ----- (SQRT)
XXX IF (VAL(M) .LT. 0.0) GO TO 600
    VAL(N) = SQRT(VAL(M))
    GO TO 500
    .....

500 VAL(M) = SPC4
GO TO 100
    ISSUE ERROR
    MESSAGE & RETURN
    
```

に、順番号に変換する。以上で演算式の解析は終りで、実行可能な演算解析表が完成したことになる。演算子番号は変換されたことを示すために、肩に°を付けた。この段階では、優先順はすでに PRI 欄に与えられているので、括弧のことは考慮する必要はなく、消去されている。

この方式で演算式の解析が完了していることを示すために、実行ルーティンについて説明しなければならない。次表に、実行ルーティンの概要のプログラムと上掲の例題についての実行過程の説明とを掲げた。いま

$$A = 2.0, \quad B = 5.0, \quad C = 2.0,$$

であるとしよう。

翻訳が完了し、実行ルーティンに制御が引渡されると、実行ルーティンに演算解析表を参照しながら作動する。

まずVNO 欄を参照して、その時の各変数の値を VAL 欄に転送する。表中のINITIAL VAL に示したように値が転送され、VAL の他の部分は0が入る。

M=PRI (1) は6であり、演算解析表の第6行目が実行される。処理に当たっては、OPE(6)にある演算子[*]に対応する演算番号がKとなり、それによって、所定の演算ステートメントに分岐し実行することになる。演算対象はこの場合VAL(M-1)→VAL(5)である。答もまた、VAL(5)に入れられる。但し、VAL(M-1)が、blankであればVAL(M-2)が、さらにそれもblankであればVAL(M-3)が、というように、blankがあればスキップされる規則である。演算対象を確定して演算番号によって分岐し、1回の計算が行なわれる。この場合は

$$VAL(5) = VAL(5) * VAL(6)$$

であり、答えは25.0となる。計算が終ればVAL(6)にはblankが埋められる。

次いで、プログラムの先頭にかえて L=2とし、8=PRI(2)であるから、第8要素の演算

$$VAL(7) = VAL(7) * VAL(8)$$

を実行する。答えは8.0である。L = 3に進んで、9 = PRI(3)であるから第9要素の演算を行うことになる。この場合、VAL(8)はブランクであるから

$$\text{VAL}(7) = \text{VAL}(7) * \text{VAL}(9)$$

が実行され、答えは16.0となる。

L = 4 の場合に進む。7 = PRI(4)であるが、VAL(6)はブランクで

$$\text{VAL}(5) = \text{VAL}(5) - \text{VAL}(7)$$

となり、答えは9.0である。その次はL = 5となり、L = PRI(5),

$$\text{VAL}(4) = \text{VAL}(4) + \text{VAL}(5)$$

であるが、VAL(4)のもとの値は0.0であるから、これは、VAL(5)の値をVAL(4)に転送し、VAL(5)にブランクを入れることになる。

さて、L = 6に進んで、4 = PRI(6)で、この計算は、

$$\text{VAL}(3) = \text{SQRT}(\text{VAL}(4))$$

となるが、もし、VAL(4)が負であれば、平方根を求めようとすれば、アブノーマル・エンドになってしまう、必要なテストを事前に行うことは、このシステムの場合のようにタイム・シェアリングを前提し、しかも演算が必ずしもスカラー演算ではなく、極めて大量のデータの全レコードについて行なわれている場合には、あるいは、今まで長時間にわたって行なわれて来た作業を、1, 2の不良データのために無にしまうことになりかねない。

インタプリタティブな演算処理は、ここで大きな特色をもつことになり、ユーザーは、中間結果のダンプや、プログラムの変更を行う機会を与えられることになる。非線型連立方程式の解の場合について、CUP・タイムのみで数分を必要とする演算で、FORTRANで直接、式を与えた場合と、この方式によった場合とでは3~4倍の差がある。この差は一見大きい様にみえるが、この方式によって得られる諸便益——特にプログラム・デバッグ時間を考えたとき、この種の応用システムにとっては、この方式の方が、はるかに優れている。

数式表の機能

複数の演算命令文を一括して処理するために、数式表(MCMP テーブル)が

つくられる。数式表は演算解析表を集めて一表にしたものである。このために必要な処理は、優先順位PRI欄の番号を変更するだけである。

いま、例として

$$A = B + C / D :$$

$$X = Y * * 2 . 0 * Z :$$

数式表PRI欄の0の位置はに、その式のテーブル長が負の値になって挿入される。

という2つの式を考えてみよう。これらはおのおの右の演算定義表のように翻訳される。

演算定義

表数式表MCM P

	VNO	OPE	PRI
0	#A	=°	0
1	#B	=+°	3
2	#C	+°	2
3	#D	/°	1

	VNO	OPE	PRI
0	#X	=°	0
1	#Y	=+°	2
2	#2.	**°	3
3	#Z	*°	1

	VNO	OPE	PRI
0	#A	=°	0
1	#B	=+°	3
2	#C	+°	2
3	#D	/°	1
4	#X	=°	0
5	#Y	=+°	6
6	#2.	**°	7
7	#Z	*°	5

演算定義表は省略して説明に必要なVNO, OPE, PRI欄のみを掲げた。数式表は演算定義表と同じ構造で、長さだけが異なり、130要素まで格納することが

できる。右に掲げた数式表はやはり省略されている。

さて、2つの演算定義表を併合する場合には、単純に数式表上に順序に2つの演算定義表を転送してから、優先順位番号PRIを修正する。この修正規則は各式の先頭の優先順位は必ず0であるから、この位置に対応する数式表の番号を、もとの優先順位番号に加えればよい。

演算の実行が、演算定義表によって行なわれたときは、一表の計算が終れば処理を終了するが、数式表による場合は、同じ実行ルーティンを用いて、優先順位が0になるたびに、別の式とみなして、計算の実行をくりかえす。このシステムでは、原則として1命令文ごとに翻訳と実行とをくり返し、そのためのプログラムのオーバーレイが行なわれる。したがって、1命令文が、大規模で実行に長時間を要するプログラムの場合はよいが、数命令文が、短かい式等で構成されているときは、極端に非効率になってしまう。

数式表は、この欠点を補うためのものである。数式表を構成するには、特定の命令文を設け、これによって、数式表の構成を指定する。たとえば

MODIFY DATA (NAME) :

STOP MODIFY :

という2つの命令文を設け、この2命令文にはさまれた諸命令文は、数式表に一括されるものとする。数式表はシステム・ファイルに格納され、必要な場合に参照されるが、このために数式表名をつける。数式表名は、便宜上、行列名表MTXNMを用いて登録される。ただし、ここに掲げた命令文は、例としての便宜的なもので、附随機能と組合せた数式表作製命令が、複数組設定される。

MODIFY DATA命令文によって、数式表の初期化が行なわれ、名前が登録されると共に、共通領域のMAPRGが0に、MAPSW が1にセットされる。以下1命令文ごとにMAPRGに1を加え、何命令文を一表にしたかが勘定される。数式表に統合可能な命令文は

COMPUTE 文

IF 文

LOGICAL IF
 ARITHMETIC IF
 EJECT IF
 GO TO 文
 COMPUTED GO TO
 UNCONDITIONAL GO TO
 CONTINUE 文

である。これらは翻訳過程において MAPSW を参点し、0 であれば翻訳を行って、直ちに実行に移るが、≠0 であれば、翻訳を行って、結果を数式表に移し実行は行なわない。

STOP MODIFY 命令文が来ると、MAPSW を 0 にし、数式表の優先順位を修正し、ファイルに書き出す。数式表に統合された一群の命令文は、直ちに実行される場合もあるが、通常はファイルに収納され、他の命令文によってその命令文に対する入力データの加工のために用いられる。STOP MODIFY 命令文のかわりに、積率行列の計算その他の集計データの計算命令文に、数式表の編集機能を組合せたものも作製しうる。

これらの詳細は、各サブ・システム命令文の設計書において説明されるべき性質のものである。

添数の処理

FOIL の文法規約では、変数名に添数をつけるときは、不等号 [$<$ $>$] で囲むことになっている。認められる添数の次元は、変数の種類によって異なるが、最大 3 次元までである。添数としては、整定数、整変数、および 3 項までの整数間の算術演算式を用いることができる。ただし、添数付の整変数を用いてはならず、算術演算式には、加減乗除巾 [$+ - * / * *$] を任意の順序で用いてもよいが、括弧 [$()$] および関数を用いてはならない。2 元以上の場合、添数と添数との間はコンマ [,] で区切るものとする。以下に示した例は、いずれも妥当な添数の記法である。

$\langle 5 \rangle \langle 1 \rangle \langle 1 + J / 5 \rangle \langle 1, 3 \rangle \langle K, L * M * * 2 \rangle$

時系列変数の場合、添数は2次元まで認められているが、第1添数は通常の意味での添数ではなく、タイム・ラグを意味する。従って、この場合第1添数は通常「-」の値となる。その他の場合は、負値をとる場合は、エラーとみなされる。

添数の翻訳は、一般の算術代入文の翻訳と同じであり、前節までに説明したのと同じロジックによって翻訳されるが、処理ルーティンを簡単にするため、強い制約を置いている。2次元以上の場合は、各次元が別別に翻訳される。

1 命令文つつ翻訳に引つついて実行が行なわれる場合〔MODE(3) = 0 の場合〕には、翻訳と共にその値が計算されて、結果を引数表に入れる。第1添数の値は、引数表のLAG欄、第2添数の値は引数表のSUF欄に入り、第3添数は引数表のSUF 2欄に置かれる。

その他の場合〔MODE(3) ≠ 0 の場合〕は、翻訳が行なわれ、翻訳結果は、演算定義表と同じ形式の添数定義表に入れられる。添数定義表の作製手順は、演算定義表および数式表の作製手順と同じである。但し、1命令文に含まれるすべての添数が、1枚の添数定義表上に翻訳される。引数表には、1命令文つつの翻訳実行の場合と同様に、第1添数がLAG欄、第2添数がSUF欄、第3添数がSUF 2欄に入るが、その状態は次のようになる。

1. 添数が、1個の整数のみである場合

対応する欄にその値が入る。

2. 添数が、1個の整変数のみである場合

その変数の整数定義表上の位置を示す番号に、3000を加えたものが、対応する欄に入る。

なお、以上、2つの場合は、引数表のみにその情報が置かれ、添数定義表には添数翻訳の結果は何も入れられない。

3. 添数が、算術演算式である場合

添数定義表上の、その添数演算式の翻訳結果が格納されている先頭位置に対応する番号に、4000を加えた値が、引数表上の対応位置に入る。

4. 数式表に編集される演算命令文が添数をもつ場合

この場合には、MODE(3)の状態にかかわらず、添数定義表が作製され、その翻訳結果は、命令文自体を数式表に編集する前に、同じ形で数式表に入れられる。つまり、添数の計算式は、独立の演算命令文として取扱われる。添数計算式の左辺には、数式表上の対応する添数格納位置を示すよう修正される。

添数の処理ルーティンは、各変数の翻訳ルーティンに組込まれており、FMCP言語によって構文をプログラムする場合、添数を考慮する必要はない。翻訳結果をファイルに出力するとき、コマンド[CMD]および、引数表[ARGTBL]とともに添数定義表も一括出力される。単一の演算命令文の場合の処理は、引数表の代りに数式表が出力されるだけで、添数の処理については、一般の命令文とかわらない。

実行時においては、FEXECルーティンの先頭に、添数処理ルーティンがあり、このルーティンが、引数表の内容を解析して添数の処理を行う。BASICサブ・システムに含まれている、整数および実数については、VNO欄は、添数で修正した値が入る。レコード変数についても同様である。時系列変数およびモデル変数については、VNO欄の値は、第2添数によって修正され、第1添数[タイム・ラグ]は、実行時における値が計算されて入っている。その他の変数については、各サブ・システムのシステム編集マニュアルにおいて説明する。

FOIL 命令文の実行ルーティン接続

実行ルーティンの編集と接続

FOILシステムにおいては、命令文はそれに対応する実行ルーティンをもち命令文翻訳ルーティンは実行ルーティンに対して引数を準備するものと考えられている。この翻訳ルーティンと実行ルーティンとの対応は命令表CMDTBLを通じて取られ、命令表はソース・プログラムの命令文が入力されて、その命

令名称が識別されると、どのような翻訳が行われるべきかを指定し、翻訳後はどの実行ルーティンを選んで制御を引渡すべきか、についての情報を保存している。

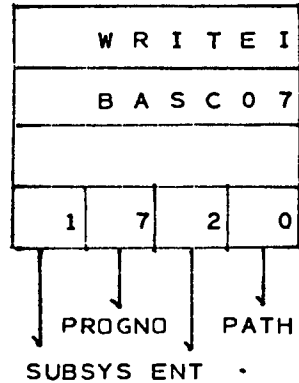
この情報は右図のような形で、CMD に入れられている。

第1語にはこの命令文の簡略名 SHORT NAME
 が入り、第2語にはフェーズ名 PHASE NAME
 が入る。第3語には翻訳ルーティン用の内部コマンドが、第4語には実行ルーティン用の内部コマンドが入る。

FEXEC ルーティンに制御がわたると、実行用コマンド CM

D(4)は直ちに2バイトとづつ4個に分解されて、前から順に、共通領域の

CMD(4)

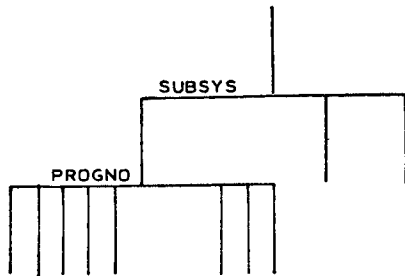


SUBSYS, PROGNO, ENT, PATH

にそれぞれ格納する。これらの情報によって所要の実行ルーティンをさがしてロードしそれへ分岐する。オーバーレイを行う方式は、オペレーティング・システムによって差があるので、どの情報を用いるかは対象機種によって異なる。

LLINK, OPSYS 等の連結ルーティンをプログラム中でCALLして、オーバーレイを行う形式をとるシステムの場合は、PHASE 名と、ENTおよびPATHの内容をパラメータとして分岐を行う。

リンケージ・エディタがオーバーレイ構造を編集する形式のシステムでは、実行コマンド・パラメータのうち、SUBSYS および PROGNO を用いて、分岐の木構造をつくり、そのプログラムがFEXEC に組込まれている。各命



FULL NAME			SHORT	PHASE	COMMAND		
XXX	XXX		XXX	BASC01	1 1 1 0	1 1 1 0	
XXX	XXX		XXX	BASC02	1 2 1 0	1 2 1 0	
XXX	XXX	XXX	XXX	BASC03	1 3 1 0	1 3 1 0	
XXX			XXX	BASC04	1 4 1 0	1 4 1 0	
XXX	XXX		XXX	BASC05	1 5 1 0	1 5 1 0	
XXX	XXX		XXX	BASC06	1 6 1 0	1 6 1 0	
				BASC07	1 7 1 0	1 7 1 0	
				BASC08	1 8 1 0	1 8 1 0	
				BASC09	1 9 1 0	1 9 1 0	
				BASC10	110 1 0	110 1 0	

命令に対応してダミー・サブルーティンが組込まれており、システムはいわば、完成した形に編集されている。以上を考慮して、各命令文に対応するサブルーティンを作製する場合には、その命令文を登録する命令表を参照して、その命令名称に対応するフェーズ名を、サブルーティン名とすることを規約する。フェーズ名を用いて分岐するときは、システムは、自動的にフェーズ名の後に、文字'P'を追加したものをフェーズ名として作り出し、それを用いて分岐するようになっている。

いま、命令表は上の表のような状態であるとしよう。BASC05 という PHASE 名をもつ命令文に対応する実行ルーティンのサブルーティン名は [BASC05] と名付ける。それをプログラム・ライブラリに登特するとき、フェーズ名は [BASC05P] とする。

例として、新しく命令文を登録し、その実行ルーティンを作製する手順について説明しておく。

命令の登録 まず、命令文の名称を命令表に登録しなければならない。

この例では、

```
READ INTEGER, I1, I2, ....., I8 :
```

```
WRITE INTEGER, I1, I2, ....., I8 :
```

という2つの命令文をとりあげる。この命令文の実行ルーティンは簡単で短いプログラムであるから、2命令文を1つのプログラムにまとめることとする。

命令表を調べると BASIC サブシステムには第 6 番目まで命令が登録済であるから、7 番目、8 番目を使用することとする。登録のためのプログラムは、

```
REGISTER STATEMENT, READ INTEGER
      ( READ I ) ;
      WRITE INTEGER ( WRITE I ) ,
      BASC07 ( 1, 7, 2, 0 ) , ( 1, 7, 2, 0 ) :
```

である。第 7 番目への登録はフェーズ名、内部コマンドを変更する必要がなく、かつ空番の先頭へ入れるので、命令文名称とその簡略名だけを書けばよい。第 8 番目への登録は、これに対応するルーティンがすべて第 7 番目の中に含まれており、フェーズ名、内部コマンド共に変更しておかねばならない。登録は空番の先頭に対して行なえばよい。フェーズ名は BASC07 を使用し、内部コマンドは第 7 番目のものと同じで ENT だけが 2 となる。

以上で命令文名称の登録は終る。

構文の登録 次に翻訳のための構文表の登録を行なわねばならない。FMCP のプログラムは、次頁に例示した。登録のための命令文は、

```
REGISTER SYNTAX ( READ I ) :
```

である。これによって、FMCP プログラムが翻訳され、構成された構文表が登録される。構文表は 2 命令文をエンタリーで区別して処理するようにプログラムされており、登録されると対応する内部翻訳用コマンドの ENT 欄に、そのレコード番号が記入される。

FMCP プログラムでは、命令表の ENT に対応して、2 つの PG 命令を書く。これは、上述の PATH 記入を行うためのものであると共に、会話処理時には、GT 命令と共に、ユーザに対して、この命令文の書き方を示す機能をもつ。

READ, WRITE 共、内容の記述法は同じで、整数名をならべて書けばよく、整数名 8 個までを指定できるものとする。8 個としたのは、出力時のブラウン管の 1 行の字数を考慮した結果である。ただし、同じ整数名でもそれに対する処理はちがって来る。READ の場合は入力であって、整数名自体、この

F M C P PROGRAM FOR READ INTEGER & WRITE INTEGER

LOC	INS	OPE	NXT	STO	PARAM	MESSAGE
	PG PG	N1 N2			READI WRITEI	READ INTEGER WRITE INTEGER
* N1 N2 N3 *	AD AD GT	F F	N3	11 1		
	I DO SP I	F K F	N5 2 N4 N5	1 8 K		11, 12, , 18 :
K * N4 N5 * F *	DT ID // NP \$\$ \$\$:		0		

```

FOIL START , ' EXAMPLE OF STATEMENT REGISTRATION ' ;
LIBRARIAN START , BASIC :
REGISTER STATEMENT , READ INTEGER ( READI ) ;
                    WRITE INTEGER ( WRITEI ) ,
                    BASC07 ( 1 , 7 , 2 , 0 ) , ( 1 , 7 , 2 , 0 ) ;
REGISTER SYNTAX ( READI ) :
LIBRARIAN END #1 :
FOIL END :
/*

```

F M C P PROGRAM

命令文ではじめて現われるものであっても差支えない。つまり未定義なら、この命令文で定義すればよいのである。これに対して WRITE の場合は出力であり、未定義なら出力の仕方がないから当然エラーである。プログラム中では

この機能の差を変数〔F〕で規定しているが、プログラムの先頭で、このFの値をセットしなければならない。

整数名1個は、少くとも指定されねばならない。後はくりかえして8個まで指定できる。命令文の終了は、変数名の後に〔,〕をうつか〔:]をうつかで区別される。

誘導メッセージ、エラー・メッセージは、内蔵されたものだけを出すこととし、以上でプログラムは終了する。

以上のプログラムを一括して示すと以下ようになる。

実行ルーティンの作製の登録

続いて実行ルーティンを作製しこれをシステムに組込まねばならない。翻訳結果はすべて共通領域に格納されているので、これを用いて必要なパラメータを得、実行ルーティンがそれによって作動するようプログラミングを行う。次頁に例示した。

サブルーティン名は〔BASC07〕とする。FOIL V-10用のCOMMON CARD DECKを付ける。エントリー番号はENTに入っているもので、それによって、2種のプログラムに分岐する。

READ INTEGERの場合は、わかり易いように簡易なプログラムとした。実際には自由型式の入力ルーティンが組みこまれている。この例の場合、固定型式で、まずカード1枚分のデータを、バッファースTOに入力する。命令文に何個かかれるかはわからないが、引数表ARGTBLのVNO欄に、命令文に書かれた個数に対応して変数番号が書かれている。0なら、その後は、変数名が指定されていない筈である。変数番号には、型番号(整数の場合3000)が付けられているので、それを除き、IVTVL表の添数に変えて、対応するSTO上の入力データを整数表に格納する。

WRITE INTEGERの場合、頁替を行って出力する場合と、行なわないで出力する場合とがある。この区分は制御番号を用いて行い、制御番号が0でなければ頁替を行うものとする。

頁を替えた場合は、見出しをつけなければならない。これはHEADERサブ

PROGRAMMING EXAMPLE NO. 1

```

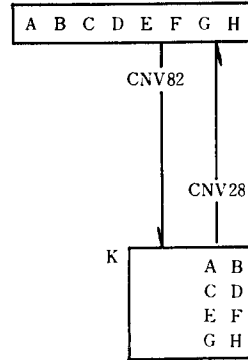
SUBROUTINE BASCO7
-----
C      ENT 1.  READ
C              INTEGER #N,  I1, I2, ... , I8 :
C      ENT 2.  WRITE
C      -----
C              FOIL V-10 COMMON DECK
C      -----
C      INTEGER*4  STO( 20 )
C      -----
C      GO TO (100, 200), ENT
C
C      ----- READ INTEGER -----
100 READ (5,105) STO
105 FORMAT ( 8I10 )
C
C      DO 120 I = 1,8
C          IF (VNO(I) .EQ. 0) GO TO 300
C          INO = MOD (VNO(I), 1000)
C          INTVL(INO) = STO(I)
120 CONTINUE
GO TO 300
C
C      ----- WRITE INTEGER -----
200 DO 210 I = 1,8
C          IF (VNO(I) .EQ. 0) GO TO 220
210 CONTINUE
220 I = I-1
C
C      IF (CNTL .EQ. 0) GO TO 230
C      CALL HEADER (' INTEGER DATA LIST @')
C      CALL LINE ( 0 )
C
C
C      230 CALL LINE ( 7 )
C          WRITE (6,245)
C          WRITE (6,246) (VNAME(K), K=1,I)
C          WRITE (6,247) (LAG (K), K=1,I)
C          WRITE (6,248) (IVAL (K), K=1,I)
C          WRITE (6,245)
C
C      245 FORMAT (1H , 8('-----'))
C      246 FORMAT (1H , 8(1X, A8))
C      247 FORMAT (1H , 8(2X, '( ',I3,' )'))
C      248 FORMAT (1H0, 8I9)
C
C      300 RETURN
C          END

```


5. 8-BYTE TO 2-BYTE

CALL CNV82 (X, K)

8バイト1語のデータを2バイト宛4個に分解し、先頭から順に、4バイト4語の領域に格納する。各4バイトの前2バイトは0



6. 2-BYTE TO 8-BYTE

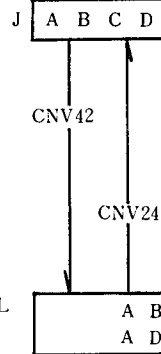
CALL CNV28 (K, X)

4バイト4語の領域に格納されたデータの各語の後2バイト宛をとり、8バイト1語に編集する。各語の前2バイトは無視する。

7. 4-BYTE TO 2-BYTE

CALL CNV42 (J, L)

4バイト1語のデータを2バイト宛2個に分解し、4バイト2語の領域に格納する。格納先の各4バイトの前2バイトは0になる。



8. 2-BYTE TO 4-BYTE

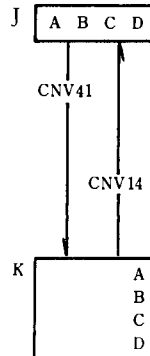
CALL CNV24 (U, J)

4バイト2語の領域に格納されたデータの各語の後2バイト宛をとり、4バイト1語に編集する。各語の前2バイトは無視する。

9. 4-BYTE TO 1-BYTE

CALL CNV41 (J, K)

4バイト1語を1バイトづつに分解し、これを4バイト4語の領域に格納する。格納された1バイトは各語の最後のバイトを占め、前3バイトは0になる。



10. 1-BYTE TO 4-BYTE

CALL CNV14 (K, J)

4バイト4語の領域に格納されたデータの各語の後1バイト宛をとり、4バイト1語に編集する。各語の前3バイトは無視する。

対話型書類管理システム

— SELF —

民 野 庄 造

目 次

はじめに

I SELF の構造

I・1 SELF のシステム構成

I・2 基本事項

I・3 ファイル

I・4 システム環境

II 適用例——刊行物発送管理(MAIL SYSTEM)——

II・1 データ定義

II・2 MAIL SYSTEM の処理パターン

II・3 操作例

II・4 各モードの操作例

は じ め に

年々増加していく学術情報を網羅的・組織的に収集・蓄積し利用者の要求に応じた的確な情報を提供する情報処理システム（情報検索システム，文献情報システム等）は，アメリカ国立医学図書館(NLM)のMEDLARS⁽⁴⁾，NASAのIRシステム⁽⁵⁾⁽⁶⁾，日本では日本科学技術情報センター（JICST）の理工学文献ファイル等に代表されるように科学技術，臨床医学等の発展に貢献し専門分野では高く評価されている。実験システムでは，Cornell大学のSMART情報検索システムに見られるように，自然語表現，自動テキスト処理という高い水準で研究がすすめられている。

これらの情報検索システムの主たる目的は，学術研究・開発・実験等に関し

て必要な一次情報，ないしその所在を提供することであり，情報要求者の求めに応じて提供される情報がどの程度要求を満たしたかが重要な課題である。そのため情報検索システムでは；

- (1) 網羅的データの蓄積，データの組織的管理，及び処理の効率化。
- (2) 主題分析・言語分析・テキスト処理の効率化，若しくは自動化。
- (3) 使用者の望むファシリティの提供，例えば対話方式による情報探索。

等の要件が満たされるものでなければならない。とくに情報検索のかなめである主題分析・言語分析に関しては研究が盛んに行われている。

SELF (Sort, Edit and List Files) は，上の(2)の機能を除きデータの蓄積・通常の検索機能に加えて，データ入力，検索データの分類・統合・編集，データ・ベースの保守，報告書等の情報操作機能を強化した情報処理システムである。一般にデータ・ベース型の事務処理への適用を想定している。コンピュータの操作は，表示画面による対話形式でノンプログラマ(エンド・ユーザー)向きに設計されている。

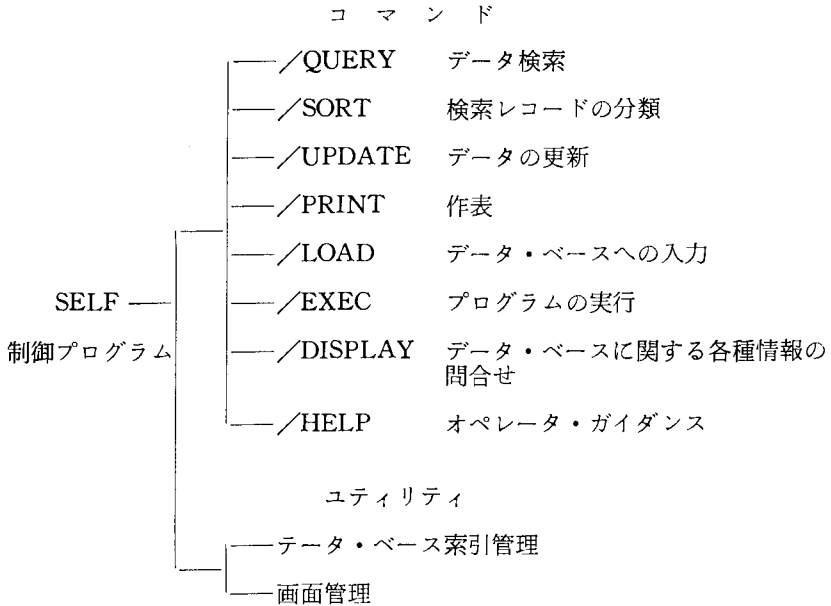
SELF で特に注目されるのはデータ・ベース開設の部分を除き，システム設計から，システム情報の問合せ・データ入力・蓄積データの保守・処理手続の作成・報告書出力等まで，適用業務の計画，管理運営の殆んどの部分をユーザー自身で行い得る情報処理システムであるということである。

適用業務は，アドレスの管理と資料配布システム (MAIL SYSTEM) ，文書管理，人事情報管理，文献情報システム等，ファイルを持つ業務に広範囲の適用が可能である。

本稿は SELF の基本構造と使用例を示したもので，“I・4 システム環境”で表わしたSELF をとりまくデータベース・システム，データコミュニケーション・システム等のサポート・システムに関しては別体系のシステムに依存するためここでは触れない。

I SELF の構造

I・1 SELFのシステム構成



(1) SELF 制御プログラム

- データ・ベース/データ・コミュニケーション・システム(DB/DC)とのインターフェース：メッセージ、及びデータの受渡しを行う。
- 語彙・構文解析。
- コマンドの実行制御：モードの切替、前処理、後処理。

(2) コマンド

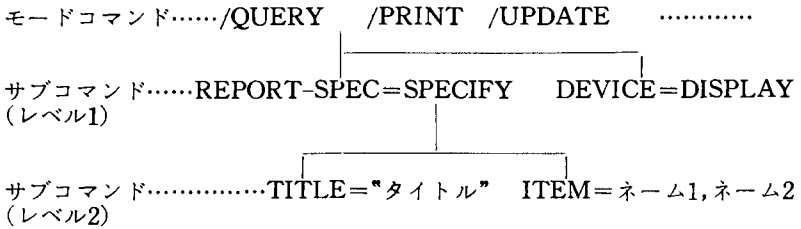
使用者と SELF システム間のコミュニケーション操作言語

(3) ユティリティ

- 画面管理：ディスプレイ画面のフォーマット定義、マッピング（表示装置の伝送データとユーザ・プログラムで扱うロジカル・データとの変換機能）。
- 索引管理：論理データと物理データの対応関係をとる各種索引の作成、並びに保守。

I・2 基本事項

I・2・1 コマンドの階層



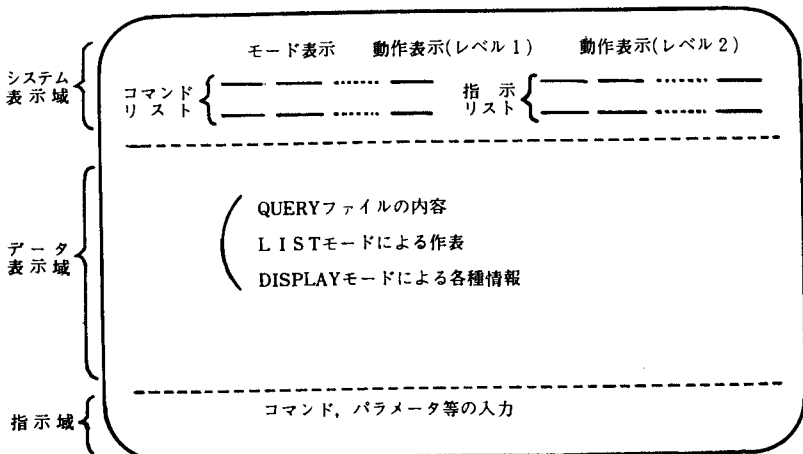
I・2・2 対話の一般的パターン

- (1) モードコマンドをキーイン。
- (2) 指定可能サブコマンド, および要応答メッセージ (画面の“動作表示部”に?…?) をシステムより表示。
- (3) サブコマンド入力。
- (4) システムより指定可能指示リスト (パラメータ) を表示。
- (5) パラメータの入力 (実行)。

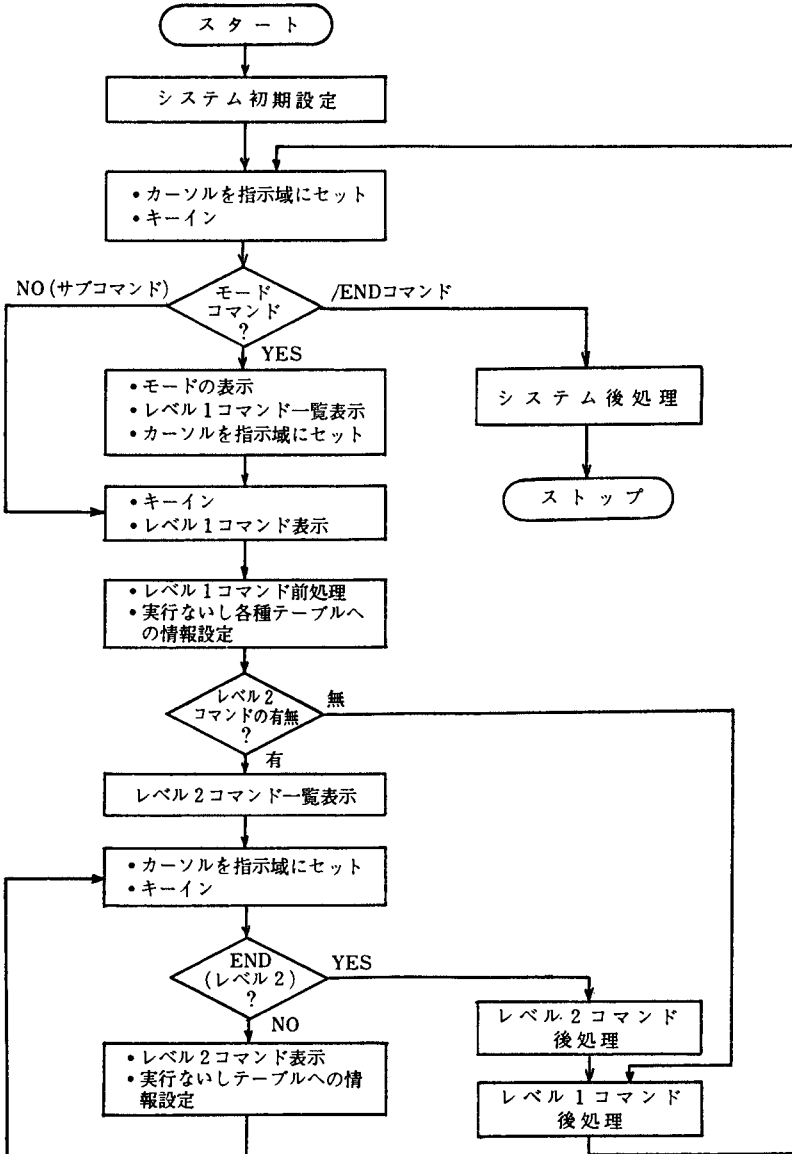
必要に応じて/HELPコマンドを用い, (2)・(4)項の列挙事項, 入力手順等のガイダンスを受ける。

I・2・3 表示画面レイアウト

図 I・2・3 表示画面



I・2・4 主制御フロー



I・3 ファイル

I・3・1 データ・ベース関係

(1) Text ファイル

タイトル, アブストラクト, 住所・氏名等データ・レコードのテキスト。

(2) Attribute ファイル

資料の属性を表わす情報, 文献情報管理の例では, 資料番号, 分類標目, 受入・登録・更新・発送等の日付, キーワード等。

(3) Text Index ファイル

Text ファイルの索引。

(4) Attribute Index ファイル

Attribute ファイルの索引。

(5) Inverted ファイル

キー(Key) 項目の転化索引。

I・3・2 システム・ファイル

(1) Query ファイル

● /QUERY コマンドによって検索されたデータ・ベースのレコードは, 当該ファイルに蓄積される。

● 収容項目: ファイル名, レコード長, 更新区分 (追加・削除・更新要不要・更新日付), リンケージ・パス等。

● SORT, UPDATE, PRINT, EXEC の各モードで参照されるデータは, すべてこのファイルである。

(2) Save ファイル

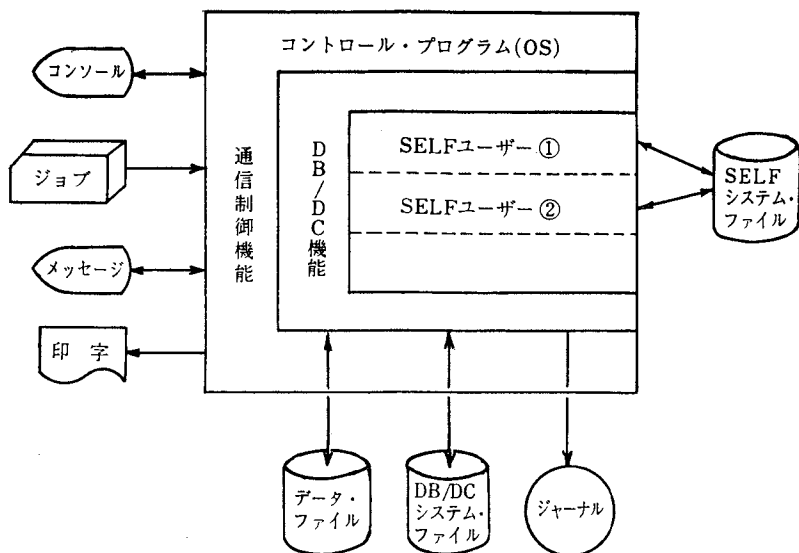
● Query ファイル上の指定レコードに名前 (パラグラフ名) をつけて蓄積・保存する。

● 収容項目: パラグラフ名, 登録日, 削除フラッグ, 削除日付, レコード数, レコード長等。

(3) 出力仕様ファイル

報告書の出力仕様を保存する。PRINT モードで呼出され出力様式の詳細仕

図 I・4 動作環境



様を当ファイルより参照する。

I・4 システム環境

SELFの動作環境を図I・4に示す。本資料では機能のみ列挙する。

(1) 通信制御機能

- 通信システムの定義：通信回線，リモート局（端末またはCPU，通信制御装置）等。
- 回線群とユーザ・プログラムとの結合・切離し。
- 計算機システムと局との送受信（定められた伝送制御手順にしたがう）。
- エラー処理：検出，回復，記録，通知。
- バッファの確保，解放。
- コード変換。

(2) データ・ベース/データ・コミュニケーション機能(DB/DC)

- データ・ベース定義：蓄積構造，データ定義，メッセージ定義，データ保護。

- データ・ベース・アクセス：データ・ベースをユーザー・プログラムより呼出す。
- データ・コミュニケーション：メッセージのルーチング（端末——ユーザー・プログラム相互間）、メッセージの待行列処理、メッセージ編集、機密保護リソースのスケジューリング、メッセージの呼出し等。
- チェック・ポイント/リスタート。

Ⅱ 適用例

—— 刊行物発送管理 (MAIL SYSTEM) ——

当経済経営研究所では部門研究のほか、専門委員会等所外の研究者も含めた共同研究の成果を当研究所が発行する経済経営研究年報、経済経営研究叢書、Kobe Economic & Business Review 等の刊行物に発表している。これらの刊行物は資料交換あるいは寄贈の形で発送されている。発送先地域は、国内・外、発送機関は国公立大学、官公庁、企業、団体、法人等にわたっている。

発送事務で特別の扱いを要するのは、

- (1) 刊行物によって発送先が異り発送部数もまちまちである。
- (2) 刊行物—発送先別、発送継続期間の掌握。
- (3) 発送先の新規登録、登録削除、住所・宛名変更等の台帳管理。
- (4) 受領・交換資料の受入状況確認。

など発送先に関する個別的な移動事項が頻繁に発生する作業であり的確かつ能率的な処理を行うシステムが要請される。

うえの業務を事務の機能あるいは流れとして整理すると次のように分けられる。

- (1) 発送先基本データのファイル化、移動事項の更改等発送台帳管理。
- (2) 刊行物の受領確認、交換資料の受入状況の記録、及び寄贈依頼に対する措置。
- (3) 発送作業と内部事務処理：住所・宛名印刷、発送伺のための業種別一地

域別発送先リスト。

刊行物の発送管理は、うえの事務分析から電算処理に恰好の業務である。当該業務をSELFにのせることによって次のような効果が期待できる。

(1) 第一に上げられるのは、概要でも触れたように電算化へのシステム設計、処理手続きのプログラム化、すなわち表示画面による操作、及びそのアプリケーション・システムの管理・保守の殆ど部分（ファイル設計とファイル開設を除き）をユーザー自身で行うことができることである。

システム使用にあたっての操作は、被選択項目の表示によるシステム主導の問合せに対する応答の形がとられる。また、/HELP コマンドを使えば使用法についての援助をシステムから受けることもできる。ユーザー自身に必要なとされる知識は、たかだか数時間で十分理解出来る SELF の言語機能と表示装置の使い方を知っておればよい。

(2) 移動事項の書換え、新規登録・削除等の更新作業を何時でも行うことができる。表示装置をつうじてこれらの操作を行うので能率が上げられる。とくに特長として上げられるのは、ローマ字の打鍵によるカナ文字への自動変換機能^(a)がとり入れられていることである。

(3) 基本台帳の各々の項目に対して即該問合せが可能であるからシステム運営の当事者はもとより、外部からの問合せに対してもの確かつ速やかに対応することができる。

Ⅱ・1 データ定義

データ定義は、電子計算機の中では物理構造として記録されているデータをユーザーが用いる論理的な名前に対応づけるため、システム環境で示したDB/DC 機能部のデータ・ベース管理機能を用いて行われる。データの画面上での操作は、常に定義した名前が使われる。

データ検索時に参照される特定の項目はキー項目（BANGO, GYOSHU, CHIIKI, KANKO等）として予め定義づけられ、インバーティッド・ファイル

(a) 定道 宏「コンピュータ時代のカナ文字・ローマ字考」, KIIS 11号, 関西情報センター, を参照せよ。

(Inverted ファイル) が作成されている。Inverted ファイルは検索時間を短縮する目的でつくられており、各キーの値とそれが記録されているレコードの物理的アドレスとの対応がとられている。Inverted ファイルの保守は、レコードの追加・削除等キー項目に変更が生じたときに索引を管理するユティリティによって、あるいは更新作業が一定の回数に達した場合自動的に更新される。

データ定義は、一般にデータ・ベース管理部門の協力によって行われる。MAIL SYSTEMの基本ファイル(ファイル名: ADDR)で定義されている項目名を次に示す。カッコ内の数字は項目で確保されている桁数。

(1) 属性レコード

BANGO(4)	レコード番号
CHIIKI(4)	地域区分
GYOSHU(2)	業種区分: 国・公・私立大学, 官公庁, 企業等
TOROKUBI(6)	レコードの登録日
KOSHIN(2)	更新区分: レコードの削除, 項目修正
KOSHINBI(6)	更新日付
NNENPO(1)	発送部数 (経済経営研究年報)
NOBUN(1)	// (欧文記要)
NSOSYO(1)	// (経済経営研究叢書)
NKIKAI(1)	// (経営機械化シリーズ)
NCHUNAN(1)	// (中南米研究シリーズ)
NAJIA(1)	// (アジア経済研究シリーズ)
NKINYU(1)	// (金融研究シリーズ)
NYORANJ(1)	// (研究所要覧(和))
NYORANF(1)	// (// (欧))
NKOKUMIN(1)	// (国民経済雑誌)
NROKKO(1)	// (六甲台論集)
NNANBEI(1)	// (南米研究)

JNENPO(1)	受領確認 (経済経営研究年報)
} JNANBEI(1)	// } (南米研究)
KNENPO(4)	寄贈(ブランク)・交換(交換資料レコード番号)の区分
} KNANBEI(4)	// }

(2) テキスト・レコード

ADDRESS(200)	住所 (改行・段落は特殊コードを挿入)
NAME(50)	宛名 (// //)

II・2 MAIL SYSTEMの処理パターン

刊行物発送管理での処理パターンは通常つぎの三つに分けられる。

(1) 基本データ (台帳データ) 入力

/LOAD 基本データ作成のための初期入力, レコードの追加あるいはレコードの入替作業を行う。

(2) データの更新

/QUERY 更新対象レコードをデータ・ベースあるいはSaveファイルから検索する。

/SORT 更新作業の能率が上げられるように分類する。

/UPDATE 画面上でデータの更新を行う。当該モードの終わりに更新されたデータのリスト・アップを行う。

(3) 実行 (宛名印刷, 内部事務資料)

/QUERY 作表レコードの検索。

/SORT 業種, 地域別などへの分類。

/PRINT 作表

SELFは, /END コマンドを入力することによって終了するから, うえの処理を続けて行うこともできる。

MAIL SYSTEM への指令, オペレータ・ガイダンス等の各種問合せ, データ入力指示, データの更新等のコミュニケーションのすべてが表示装置で行われる。

システムに設定された対話型言語は、オペレータの自由度を高くとりながら同時にオペレータが次にとり得るアクションの表示を画面上に示し、かつ標準機能 (Default Option) をとり入れエンド・ユーザー (ノン・プログラマ) が容易に操作できるよう考慮されている。

画面上の情報レイアウトは、「I・2・3表示画面レイアウト」で示したように三つの部分に分けられる。

(1) 最上部はシステム表示域で、現在の動作モード、動作中のサブコマンド表示、及びオペレータの取得するコマンド、指定可能なパラメータを表示。

(2) 中央部がデータ表示域で検索レコードの表示、オペレータ・ガイダンス機能による表示等。

(3) 最下部がオペレータの入力した情報を表示する指示域である。コンピュータ (MAIL SYSTEM) への指令は、この部分に情報を入力し送信キー等のファンクション・キーを押すことによって行われる。

以後操作例にしたがって説明する。

- ・大文字：オペレータの入力情報とする。
- ・小文字：コンピュータの表示する情報とする。

II・3 操作例

<例1>

ADDR ファイルから丸日電機 (株) のレコードを検索し住所表示の “ROT SUKO” を “ROKKO” に変える。

／Query	① 検索モードの設定
File-name=ADDR	② ファイル名 “ADDR” を指定
Condition NAME	③ 検索条件の設定
=マルヒデンキ／	
Retrieve	④ 検索開始
／Update	⑤ 更新モードに切替
Scan ROTSUKO	⑥ 文字ストリング “ROTSUKO” を走査
End	⑦ 更新モードを終らせる
exec-update	⑧ 更新の処理

- ④：データ定義でつけられた項目名NAME（宛名）の頭7文字が“マルヒデンキ”のレコードを検索し Query ファイルに入れる。/印は、それ以後の文字ストリングを無視する。すなわち、50文字確保されている宛名の頭7文字が比較される。
- ⑤：更新モードに切替えると Query ファイルの最初のレコードが画面上に表示される。
- ⑥：キャラクタ・ストリング“ROTSUKO”が走査されカーソルがセットされる。オペレータはカーソルをTの位置に移動し、ファンクション・キーを用いて TSU を削除しKを挿入する。このようにレコード内の文字ストリングの追加・削除・詰め等の部分修正は特殊機能キーを用いて行うので能率が上げられる。
- そのほか更新モードでは、①指示域へ所要データを呼出すためのサブコマンド、②カーソルの制御、③レコード単位のデータの追加ないし削除、④Query ファイルの指定レコード群の保存、保存しているレコード群をQuery ファイルに読出すサブコマンド等を有しているため多様なデータ操作を能率よく行うことができる。
- ⑧：End の入力直後システム側で出されるメッセージで、オペレータに実行指示を問合せくる。直接送信キーを押すとデータの更新と更新データのプリント（標準機能）が選択され当該モードで操作した更新情報にしたがってデータ・ベースの内容が更新され、ラインプリンタに更新情報がプリントされる。

<例2>

ADDR ファイルから経済経営研究叢書・経営機械化シリーズの発送先レコードを検索し、地域別に分けて郵送用宛名ラベルを指定用紙に印刷する。

/Query	① 検索モードに設定
File-name=ADDR	② ファイル名“ADDR”を指定
Store-mode=Load	③ Query ファイルの最初から入れる

C o n d i t i o n N K I K A I > 0	④ 検索条件の設定
R e t r i e v e	⑤ 検索開始
/ S o r t	⑥ 分類モードに設定
I t e m = C H I I K I (4)	⑦ Query ファイルのレコードを地域別に分類
G o	⑧ 分類の開始
/ P r i n t	⑨ 出力モードに設定
G o A D D R L S T	⑩ 宛名印刷

- ④：項目名“NKIKAI”には機械化シリーズの発送部数が入れている。したがって同条件式によって検索されるのは発送部数1以上の発送先レコードである。
- ⑧：地域コードの上昇順（標準設定）に分類が行われる。複数の分類項目の指定、逆順の分類も可能である。
- ⑩：ADDRLST は、定常的あるいは頻繁に出力されるものために予めシステムに登録されているプログラム名である。
出力対象のレコードはQueryファイルに入れられているデータである。

Ⅱ・4 各モードの操作例

Ⅱ・4・1 データレコード作成(Load)

<例L1>

磁気テープにカード・イメージ様式で入れられているデータを、予めシステムに定義されている ADDR ファイルのデータ仕様にしたがってデータ・ベースへ最初の入力を行う。

/ L o a d	① データ入力モードに設定
F i l e - n a m e = A D D R	② ファイル名“ADDR”を指定。
L o a d - t y p e = I n i t i a l	③ データ・ベースへの最初の入力（初期入力）
G o	④ 入力開始。

- ③：当コマンドを省略すれば Add-load（追加入力）が標準機能として選択される。また ADDR ファイルすべてを入替える場合は、Load-type=Re-load とすればよい。

<例L2>

カードに穿孔されているデータを ADDR ファイルに追加入する。

／Load	① データ入力モード
File-name=ADDR	② ファイル名ADDRを指定
Device=Card	③ カード読取装置を指定
Go	④ 入力開始

<例L3>

操作画面によるデータ入力

／Update	① 更新モードで行う。
Top	② 表示起点をQuery ファイルの第1レコードに設定。
Add-record 2	③ 2レコードの追加を指定、当該コマンドの後システムよりの問合せに応じてファイル名等を指示する。
<div style="border: 1px dashed black; padding: 5px; width: fit-content;"> ADDRファイルのデータ定義に準じてキーボードよりデータを2レコード入力する </div>	④ 画面に表示されたデータは、送信キーを押すことによって逐次Queryファイルに入れられる。
End	⑤ 更新モードの終了。
exec-update	⑥ データ・レコードの追加処理を行う。

- ⑥：Ⅱ・3・1 <例1>—⑧，あるいはⅡ・4・5の説明を参照せよ。

<例L3>

入力用磁気テープ(MT)の記録形式がシステムに未定義の特別仕様データ・テープを読み新しいレコードを追加する。

MTの仕様；ラベルなし，固定長（346バイト），非ブロック。

/ Load	①
File-name=ADDR	②
Input-spec=Specify	③ 入力仕様を以下に指定する。
Mt-label=No	④ ラベルなしテープを指定。
Rec-type=Fixunblock	⑤ 固定長非ブロックを指定 (標準機能)
Block-size=346	⑥ ブロック長=346バイト。
Format=(2A4,2(A2,A6),24A1,12A4,250A1)	⑦ 入力フォーマット。
End	⑧ 入力仕様コマンド(Input-spec)の終り
Go	⑨ 入力開始。

Ⅱ・4・2 検索 (QUERY)

CONDITION サブコマンドで指定した検索条件を満たすレコードがデータベースから抽出され Query ファイルに入れられる。Query ファイルの内容は、UPDATE、PRINT の各モードで画面上に表示することができる。

検索条件式で指定する項目名は、「データ定義」で定義されているものでなければならない。また検索のスピード・アップを計るため予めキー項目として登録しインバーティッド・ファイル (Inverted ファイル) が作成されているものが望ましい。キー項目として登録されていない項目名を条件式に指定すると、逐次検索方式でファイルを探すため検索時間が長くなる。

条件式は一般に、

主語 (項目名) + 比較演算子 + 参照値 (通常は定数)

の形をとる。

以下の例で<例Q1>以外は関連コマンド (/Query, File-name, Retrieve) を省略する。

<例Q1>

経済経営研究叢書・経営機械化シリーズの発送先基本レコードを検索し Query ファイルに追加モード (標準機能) で入れる。

／Query	① 検索モード
File-name=ADDR	② ファイル名“ADDR”を指定
Condition NKIKAI>0	③ 経営機械化シリーズ1冊以上の発送先
Retrieve	④ 検索開始

③：NKIKAI はデータ定義でつけられた項目名で当叢書の発送部数が入れている。

<例Q2>

宛名が“トウキョウダイガク”のレコードをすべて検索する。

```
Condition NAME=トウキョウダイガク/
```

／印は、それに続く文字ストリングを無視する特殊記号である。上例は当経済経営研究所の発行する全刊行物を対象にし、東京大学の全部局・全学部の配布先宛名レコードが検索される。

<例Q3>

経済経営研究年報を送付している国・公立大学の基本レコードを検索する。

```
Condition NNENPO>0 AND GYOSHU=01, 02
```

上の条件式は、関係子(Relational Operator ; > および=) で結びつけられた句を論理演算子(Boolean Operator ; AND) で結合し複合条件を表わしている。

上例の2番目の句は、項目と比較演算子が変わらない場合であり、参照値を複数個並べて書くことができる。

<例Q4>

レコード番号(BANGO) が295から360までの基本レコードをQueryファイルの最初から入れる。

```
Store-mode=Load
Condition BANGO < > 295, 360
```

<例Q5>

Saveファイルに保存されているパラグラフ名“SVKIKAI”のレコード群をQueryファイルに転送する。

```
File-name=SAVE
Paragraph-name=SVKIKAI
```

Saveファイルへのレコードの登録、保守は更新モード (UPDATE) の (KEEP), あるいはPURGE サブコマンドを用いて行う。

II・4・3 分類(SORT)

分類は事務処理のうちで重要な機能の一つである。SELF では分類項目名を指定することによって容易に Query ファイルのレコードの配列を変えることができる。

複数個の項目指定, および分類順を上昇・逆のいずれかに指定することもできる。ただ注意を要するのは, FOLLOWED サブコマンドで指定したファイル・フォーマットを参照して分類されるから, 同ファイルの定義と異なるレコードの分類の配列は保証されない。

以下<例S1>以外は関連コマンド(/Sort, Followed, Go)を省略する。

<例S1>

項目名“CHIIKI”(地域区分)別にQueryファイルの全レコードを上昇順に分類する。CHIIKIはファイル名“ADDR”で定義づけられた地域区分の項目名とする。

```
/Sort
Item=CHIIKI
Followed ADDR
Go
```

<例S2>

項目名“BANGO”(レコード番号)の逆順に分類する。

```
Order=Descending
Item=BANGO
```

<例S3>

項目名“CHIIKI”―“GYOSHU”（業種区分）の二項目を分類項目とし上昇順に分類する。CHIIKI 項目は4桁のうち頭3桁を分類対象とする。

```
Item=CHIIKI(3),GYOSHU
```

I・4・4 報告書作成 (PRINT)

出力する報告書とその仕様、出力様式を定めることは事務処理上最も重要な部面であり、電子計算機のプログラム化の作業も複雑で相当の作業比重を占めるのが普通である。SELFの報告書機能は、標準的な出力様式ならばパラメータを与えることによってこれらを容易に行うことができる。

また、頻繁に出力される特定様式の報告書は予めプログラム化してシステムに登録しておけば、データ・レコードの指定とその様式を呼出すのみで報告書を作成することができる。出力対象のレコードはQueryファイルに入れられているデータである。<例P1>以外は関連コマンド (/Print, Go) を省略する。

<例P1>

“LBLPRINT”の名でシステムに登録されている出力様式にしたがってフォームNo.101の用紙に宛名をプリントする。

```
/Print
Form-change=101
Go LBLPRINT
```

<例P2>

出力様式“LBLDSPLY”にしたがって配布先宛名を画面上に表示する。

```
Device=Display
Go LBLDSPLY
```


<例 P3>

出力様式の仕様をサブコマンドで指定しラインプリンタに出力する。また指定した当該出力様式をシステムに“LBLFORM1”で登録する。

R e p o r t - s p e c = S p e c i f y	①	出力様式を以下に指定
T i t l e = A D D R E S S L I S T	②	タイトルの印刷文字
H e a d i n g = ジュウシヨ, ナマエ	③	見出しの印刷
C o n t r o l - i t e m = C H I I K I (3)	④	CHIIKIで印刷を制御
P r i n t - i t e m = A D D R E S S (8 8) , N A M E (2 8)	⑤	印刷項目の指定
F o r m a t = (A / 5 X , A /)	⑥	印刷フォーマットの指定
E n d	⑦	レベル2サブコマンドの終り
K e e p - f o r m = L B L F O R M 1	⑧	うえの出力様式を登録
G o L B L F O R M 1	⑨	出力の開始

②～⑦は出力仕様を定めるレベル2のサブコマンド。

④： CHIIKI（地域区分）で印刷を制御する。すなわち地域コードが変わる度にカッコ内に指定した行数だけ印刷行をスキップする。

⑤： カッコ内の数字は、項目名 ADDRESS, NAME に確保されているそ

ADDRESS LIST

ジュウシヨ	住		所		
ナマエ		名	前	N ₁	A ₁
}					
	住		所		
		名	前	N _m	A _m

それぞれ200文字、50文字のうち頭から 88文字、28文字の印刷を指定している。

同上例では報告書形式 (Report-type) として標準機能 (Vertically) が選択されて前ページの表のように項目名が表側にとられる。

また、Report-type=Horizontally と指定し、Format=(A,5X,A/)とすると下表のように印刷される。

ADDRESS LIST				
ジュウショ			ナマエ	
住	所	A ₁	名 前	N ₁
}			}	
住	所	A _m	名 前	N _m

<例 P4>

“LBLFORM7” で登録されている出力様式名を削除する。

PURGE=L B L F O R M 7

II・4・5 データ更新 (UPDATE)

表示画面上でデータ・レコードの追加・削除、または項目の修正を当該モードに用意されているコマンドの使用とキーボードのファンクション・キーを用いて効率よく行う。カナ文字の入力はローマ字を打鍵すればカナに自動変換さ

れる。そのほか Query ファイル内の指定する記録群を Save ファイルに保存し、あるいは Save ファイルの内容を Query ファイルに呼出すこともできる。

<例U1>

指示域の表示（以下表示起点と呼ぶ）を Query ファイルの第1記録に設定する。

```
 / Update  
  Top
```

<例U2>

表示起点を Query ファイルの最終記録に設定する。

```
  Bottom
```

<例U3>

表示起点を8記録先にする。

```
  8
```

<例U4>

表示起点を20記録前にする。

```
 - 20
```

<例U5>

表示起点を Query ファイルの50番目の記録にする。

```
  Find 50
```

<例U6>

記録の内容を逐次走査し“ROKKODAI”の文字列と合致する位置にカーソルをセットする。

```
  Scan ROKKODAI
```

<例U7>

カーソルをレコードの頭から数えて 10,20,30,40番目にセットする (定義コマンド)

```
C u r s o r - s e t 1 0 , 2 0 , 3 0 , 4 0
```

同上コマンドに従ってカソルを移動させることを指定する。

```
O n - c u r s o r O n
```

<例U8>

Query ファイルの表示起点から数えて60レコードを“HOZONFL”の名前で Saveファイルに入れる。

```
K e e p H O Z O N F L ( 6 0 )
```

カッコの項を省略するとQuery ファイルの終りまでが入れられる。

同上の名前で登録されたパラグラフ (60レコード分) を Query ファイルに追加モードで入れる。

```
R e a d H O Z O N F L
```

<例U9>

Save ファイルに登録されているパラグラフ名 “HOZONFL” を削除する。

```
P U R G E H O Z O N F L
```

<例U10>

ファイル名 “ADDR” に 3 レコード追加する。

```
A d d - r e c o r d 3
F i l e - n a m e = A D D R
```

```
追加レコード入力
```

<例U11>

ADDR ファイルの表示起点から5レコードを削除する。

```
DELEte-record 5
file-name=ADDR
```

<例U12>

UPDATE モードの終了

```
End
```

同コマンドの入力直後システムより次のコマンドが出される。

```
exec-update
```

同時に次の実行指示をオペレータに問合せる。

- (1) データの更新と更新データのプリント（標準機能）。
- (2) データの更新，プリリントなし。
- (3) // 更新データの磁気テープへの出力。
- (4) 当該モードを無視する。

II・4・6 ファイル情報の表示 (DISPLAY)

データ定義に関する各種の問合せは当モードで行われる。従来は、システム仕様書、マニュアル等を参照しなければならなかったファイルの属性、コード・ブック、データの現時点の様相等をシステムに対して問合せることがでる。

<例D1>

データ・ベースに登録されているファイルの一覧を表示する。

```
/Display
Data-base
```

同上コマンドで表示される情報は、ファイル名・ファイル開設日付・最新更新日付・収録レコード数・レコード長等。

<例D2>

データ定義で定められたファイル名“ADDR”の仕様を表示する。

```
File-spec=ADDR
```

表示される情報は、項目名・項目の型（文字・数値等）・項目の長さ・登録されているキー項目等。

<例D3>

キー項目として登録されている“業種区分”のコード及び意味づけを表示する。

```
Key-code=GYOSHU
```

II・4・7 オペレータ・ガイダンス (HELP)

SELF システムの使用者はノン・プログラマを想定しており、画面操作によるオペレータのとり得るアクションを画面上のシステム表示域に示すようにしている。

HELPモードは各モードの言語仕様を詳細に示し操作上の援助を行う。ファイル仕様を問合せる DISPLAY と共に SELF システムの特徴的な機能である。

<例H1>

QUERYモードの言語仕様を表示する

```
/ Help  
Command Query
```

II・4・8 システムの終了(END)

```
/ End
```

参 考 文 献

- (1) "Generalized Information System," Program Description Manual,
Volume 1 : System Overview and Control, SH20-0622-2.
Volume 2 : Data Description Language, SH20-0623-1.
Volume 3 : Procedural Language, SH20-0736-0.
IBM Corporation.
- (2) "Generalized Information System," Operations Manual, H20-0625-0,
IBM Corporation.
- (3) Storage and Information Retrieval System for DOS/VS (STAIRS/
DOS/VS)General Information Manual GH20-1832-0, IBM Corporation.
- (4) Lancaster, F.W., "Evaluation of the MEDLARS Demand Search
Service," National Library of Medicine, 1968.
- (5) Summit, R.K., et al., "An Online Technical Library Reference
Retrieval System, American Documentation, 17(1), 1966.
- (6) Summit, R.K., "DIALOG An Operational Online Reference Retrieval
System, Proceedings-ACM national meeting, 1967.
- (7) Salton, G. Ed., "The SMART Retrieval System," Prentice-Hall Inc.,
1971.
邦訳：神保健二監訳, "SMART情報検索システム," 企画センター, 1974.

情報処理システムにおける 行列演算の記述法

定 道 宏

まえがき

代表的言語処理システムである PL/I, FORTRAN, ALGOL に行列演算を記述する言語が組込まれていない。科学計算における行列演算の重要性または利用度からこれらのシステムに当然組込まれているべきものであるといえよう。ただ添字の省略化という形式で行列を意識した記述が考慮されている。例えば、PL/I ステートメントでは宣言文での配列の次数と配列演算での添字の範囲とが一致するときに限り、添字を星印で表わすか、または省略することができる。いま、次のような配列の宣言があったとする。

```
DECLARE A(10,5), B(10,5), C(10,5), D(10,5),
        E(5,5)
```

そのとき、 $D=0$ と書けば行列Dのすべての要素の値が0となる。また、 $A=B+C$ と書けば BとC がともに10行5列の行列として行列和が計算され、結果は10行5列の行列Aに代入される。BのI行とCのK行を加えてAのI行に代入したいときには $A(I, *) = B(I, *) + C(K, *)$ と書けばよい。他方、PL/I における配列の乗算は、行列の乗算ではなく、配列の和または行列の和にみられるような配列間の対応する要素の乗算を意味する。 $A=B * C$ と書けば配列BとCの対応する要素の積が配列Aの対応する要素の値となる。また、 $A(I, *) = B(J, *) * E(*, K)$ と書けば、これは配列BのJ行と配列EのK列の対応する要素の積が配列AのI行の対応する要素に代入されることを意味する。

PL/I における配列演算は本質的に行列演算を意味するものではない。行

行列積 $A = B * E$ を計算するには次のような PL/I プログラムを書かねばならない。

```

      MXPROD : DO I = 1 TO 10 ; DO J = 1, 5 ;
                A ( I , J ) = SUM ( B ( I , * ) * E ( * , J ) )
      END MXPROD ;

```

例えば最小2乗推定値の計算式 $B = (X'X)^{-1}X'Y$ のプログラムを書くには相当高度なプログラミング技術を要することになるであろう。行列代数の知識さえあればコンピュータプログラミングに素人の人にも簡単に分析プログラムが書けるように設計された線型代数専用の言語処理システムは未だ開発されていない。しかし回帰分析や因子分析、共分散分析などの統計処理を目的とした応用分析システムの中には行列演算の言語を補足して汎用性を高めているものがある。行列演算の言語表現に2通りの方法があり、1つは行列演算式を、例えば $A = B * C$ のように表わす方法、いま1つはサブルーチン形式で、例えば MXMLT, L, N, M, A, B, C のように書く方法である。ASTRO-FOILは前者、TSPは後者の表現法をとっている。以下では、これら2通りの行列演算表現法を検討し、筆者の作成した経営計測システム SIMPL における行列演算機能を紹介する。

1. ASTRO-FOIL における行列演算

ASTRO-FOIL は和歌山大学の杉浦一平教授によって昭和45年に開発された汎用統計分析システム（「ASTRO FOIL 計画と予測のためのプログラム・システム」日本評論社昭和48年）であり、大学では和歌山大学をはじめ神戸大学、同志社大学で稼働し、民間では東洋情報システムでTSSとして利用されている。ここではこのシステムが備えている行列演算に関する言語のみを論ずることとする。

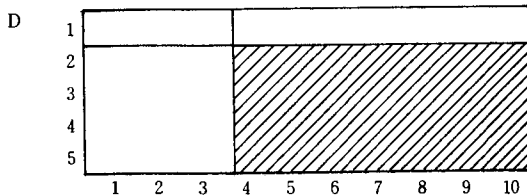
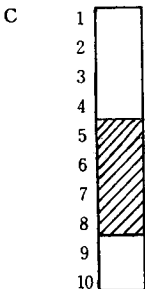
宣言文および範囲文では配列と行列は全く同義語として扱われているが、演算文では行列演算と配列演算とは明確に区別されている。宣言文は配列または行列の名称とその最大次数を定義する。宣言文は次のように表わされる。

DECL ARRAY, C<10>, D<5, 10>

DECL ARRAY を DECL MATRIX と書いてもよい。この命令文は記述部に書かれた変数名が配列または行列であり、変数名の次に大小記号で囲まれた数とその次数であることを示す。上記の例ではCは1次元の配列で次数は10であり、Dは2次元の配列で次数は(5×10)である。行列として解釈するならば、Cは10行1列、Dは5行10列の行列を表わす。次に、範囲文は、宣言文で定義された配列または行列の部分配列または部分行列の次数を指定する。配列演算または行列演算はすべてこの範囲文で指定された部分配列または部分行列に関して行われる。範囲文は次のように表わされる。

DEFINE MATRIX, (5|8) C; (2|5, 4|10) D

DEFINE MATRIX を DEFINE ARRAY と書いてもよい。この命令文は配列または行列を宣言文で定義された次数よりも小さい次数で使う場合に用いられる。Cは先の宣言文で10行1列の行列と宣言されているが、ここでは5行目から8行目までの4要素からなる部分行列として、またDは宣言文で5行10列の行列と宣言されているが、ここでは2行目から5行目、4列目から10列目までの要素で構成される部分行列として定義されている。



以上の宣言文および範囲文では配列と行列は全く同義語として用いられているが、演算文では配列の演算命令は COMPUTE ARRAY, 行列の演算命令は COMPUTE MATRIX と区別して用いられる。配列の演算規則は PL/I のそれと全く同じといえる。

```
DEFINE ARRAY, (1|3, 1|2) E, F, G, H
COMPUTE ARRAY, E=F*G; H=3*G-4
```

においてF, Gの部分配列がそれぞれ次のように表わされるとき,

$$F = \begin{bmatrix} 1 & -4 \\ 2 & 5 \\ 3 & -6 \end{bmatrix}, G = \begin{bmatrix} 1 & 0 \\ -2 & 3 \\ 0 & -4 \end{bmatrix}$$

E, Hはそれぞれ次のように計算される。

$$\begin{aligned} E = F * G &= \begin{bmatrix} 1 & -4 \\ 2 & 5 \\ 3 & -6 \end{bmatrix} * \begin{bmatrix} 1 & 0 \\ -2 & 3 \\ 0 & -4 \end{bmatrix} = \begin{bmatrix} 1 \times 1 & -4 \times 0 \\ 2 \times (-2) & 5 \times 3 \\ 3 \times 0 & -6 \times (-4) \end{bmatrix} \\ &= \begin{bmatrix} 1 & 0 \\ -4 & 15 \\ 0 & 24 \end{bmatrix} \end{aligned}$$

$$\begin{aligned} H = 3 * G - 4 &= 3 * \begin{bmatrix} 1 & 0 \\ -2 & 3 \\ 0 & 4 \end{bmatrix} - 4 = \begin{bmatrix} 3 \times 1 - 4 & 3 \times 0 - 4 \\ 3 \times (-2) - 4 & 3 \times 3 - 4 \\ 3 \times 0 - 4 & 3 \times 4 - 4 \end{bmatrix} \\ &= \begin{bmatrix} -1 & -4 \\ -10 & 5 \\ -4 & 8 \end{bmatrix} \end{aligned}$$

配列演算式の中で用いられるスカラー数は配列のすべての要素に対して演算がなされる。したがって $3 * G$ の3は配列Gのすべての要素に乗ぜられ、 -4 は配列 $3 * G$ のすべての要素に対して加えられる。

次に、行列演算の場合をみてみよう。行列演算は COMPUTE MATRIX 命令によって行われる。

```
DEFINE MATRIX, (1|3, 1|2) E, F; (2|3, 1|2) G
COMPUTE MATRIX, E=F*G
DEFINE MATRIX, (1|2, 1|2) H, E
COMPUTE MATRIX, H=3*G+E
```

行列の演算規則は行列代数における行列積の規則である。L行M列の行列FとM行N列の行列Gとの積 $F * G$ はL行N列の行列Eとなる。Eのi行j列の要素 E_{ij} はFとGの要素の積和として次のように計算される。

$$E_{ij} = \sum_{k=1}^M F_{ik} \times G_{kj} \quad i=1 \cdots L ; j=1 \cdots N$$

スカラー数と行列の積，スカラー数と行列の和，行列と行列の和の演算規則はすべて配列演算における規則と同じである。配列演算の規則と異なるのは行列積の演算である。行列積を計算するときには行列積が定義できるように DEFINE MATRIX 命令で各行列の次数を正しく指定しておかなければならない。上記の例ではFが3行2列，Gが2行2列であるから行列積 $F * G$ は存在し，結果である行列Eは3行2列となる。Eは次のように計算される。

$$\begin{aligned} E = F * G &= \begin{bmatrix} 1 & -4 \\ 2 & 5 \\ 3 & -6 \end{bmatrix} \times \begin{bmatrix} -2 & 3 \\ 0 & -4 \end{bmatrix} \\ &= \begin{bmatrix} 1 \times (-2) - 4 \times 0 & 1 \times 3 - 4 \times (-4) \\ 2 \times (-2) + 5 \times 0 & 2 \times 3 + 5 \times (-4) \\ 3 \times (-2) - 6 \times 0 & 3 \times 3 - 6 \times (-4) \end{bmatrix} = \begin{bmatrix} -2 & 19 \\ -4 & -14 \\ -6 & 33 \end{bmatrix} \end{aligned}$$

また，Hはスカラー積，行列和の演算を含んだ式である。この場合式に現われる行列はすべて同じ次数でなければならない。例ではH，G，Eはそれぞれ2行2列であり，行列演算が成立する。Hは次のように計算される。

$$\begin{aligned} H = 3 * G + E - 4 &= 3 \times \begin{bmatrix} -2 & 3 \\ 0 & -4 \end{bmatrix} + \begin{bmatrix} -2 & 19 \\ -4 & -14 \end{bmatrix} \\ &= \begin{bmatrix} 3 \times (-2) - 2 & 3 \times 3 + 19 \\ 3 \times 0 - 4 & 3 \times (-4) - 14 \end{bmatrix} = \begin{bmatrix} -12 & 24 \\ -8 & -30 \end{bmatrix} \end{aligned}$$

行列の演算式の書き方について若干の制限が付されている。

- (1) A, B, C, Dをそれぞれ行列またはスカラー数とすれば行列の演算式は次の6通りの何れかでなければならない。

$$A = \pm B$$

$$A = \pm B \pm C$$

$$A = \pm B \pm C \pm D$$

$$A = \pm B * C$$

$$A = \pm B * C \pm D$$

$$A = \pm B \pm C * D$$

- (2) 式中の行列には転置操作を自由に施すことができる。転置操作は行列の後に垂直線記号を付して表わす。例えばCの転置行列はC|で表わされる。

$$(例) \quad A = -B | + C | * D$$

- (3) 式中の行列の1つに逆行列操作を施すことができる。逆行列操作は行列の後に斜線記号を付して表わす。例えばCの逆行列はC/で表わされる。

$$(例) \quad A = B / * C$$

- (4) 転置操作と逆行列操作を同時に施すことができる。

$$(例) \quad A = B ! / * C$$

以上が ASTRO-FOIL における行列演算に関する言語の全容である。PL / I の配列演算機能と比較して ASTRO-FOIL の配列演算機能の優れている点は何と云っても配列の次数を自由に変更して部分配列での配列演算ができることである。PL / I では配列演算が宣言文の次数に従って行われているが、ASTRO-FOIL では宣言文とは別に範囲文を導入することによって配列の次数を変えることができ配列演算が固定的な宣言文の次数ではなく変更可能な範囲文の次数に従って行われている。この範囲文が行列演算における行列の次数指定に用いられ、多様性のある行列演算を可能にしている。範囲文はまさに行列演算式の表現に関わる最も困難な点に1つの解決法を示している。杉浦教授が著者の中で述べられているように「行列演算は、試作的なもので、まだ十分な検討を経ていない。」特に行列演算式が先に示された6通りではなく、もう少し制限が緩められる必要がある。演算の優先順序を示す括弧の使用は許さないにしても、加減の項の数は十分に多くし、各項は少なくとも3個の行列積まで許すべきである。例えば、最小2乗推定法で良く用いられる行列に次のようなも

のがある。

$$\mathbf{B} = (\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}'\mathbf{Y}$$

$$\mathbf{B} = (\mathbf{X}'\mathbf{X} + \mathbf{K} \cdot \mathbf{I})^{-1}\mathbf{X}'\mathbf{Y}$$

$$\mathbf{P} = \mathbf{X}(\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}'$$

$$\mathbf{S} \mathbf{S} \mathbf{U} = \mathbf{Y}'\mathbf{Y} - \mathbf{B}'\mathbf{X}'\mathbf{Y}$$

また、行列が特定の形の行列、例えば単位行列、単位ベクトル、対称行列などである場合には、行列演算に際して行列が特殊の形の行列であることが分っているならば、数値計算の簡単化または精度向上に役立つことが多い。行列が特定の形をしていることを行列演算式の中で表現する必要がある。次数の高い行列演算の場合には特に重要である。

2. TSPにおける行列演算

TSPは1966年にマサチューセッツ工科大学で開発され、1968年にはカリフォルニア大学バークレイで改良され、さらにハーバード大学やプリンストン大学でさらに改良が加えられ、1973年には Harvard-MIT 版として B. H. Hall が集大成した時系列分析システム (Bronwyn H. Hall, "TSP Time Series Processor (Version 2.5)" Technical Paper Number 12, Harvard Institute of Economic Research, Harvard University, December 1974) であり、システム開発に関係した上記の大学以外にも多くの大学で利用されている。ここではこのシステムが備えている行列演算に関する言語のみを論ずることにする。

TSP おける行列演算の言語は ASTRO-FOIL のような行列演算式による表現ではなく FORTRAN におけるサブルーチン形式の表現をとっている。いま1つの相違は行列の記憶方式である。ASTRO-FOIL では行列は部分行列にみられるように行列方式で記憶され、1列目の要素と2列目の要素が必しも連続して並んでいない。しかしTSPでは行列はベクトル方式で記憶され、1列目の要素の次に2列目の要素が並び、常に連続して並んでいる。したがってTSPでは部分行列という考え方はない。M行N列の行列はM×N行1列の行列と同じである。まず、行列積を計算する例からみてみよう。AはM行L列、B

はL行N列の行列とする。行列積 $C = AB$ を求めるTSPの行列演算文は次のように書かれる。

GGGMLT, 0, M, N, L, A, B, C

GGGMLTは行列積を計算する行列演算命令である。最初のパラメータである0は行列積の前の行列を転置しないことを示す。もしこのパラメータが1であると行列積の前の行列を転置することを示す。第2のパラメータであるMは行列Aの行次数を、第3のパラメータであるNは行列Bの列次数を、第4のパラメータであるLは行列Aの列次数または行列Bの行次数をそれぞれ示す。またDをM行N列の行列とすると行列積 $C = A'D$ を求める行列演算文は次のようになる。

GGGMLT, 1, L, N, M, A, D, C

最初のパラメータの値が1であるから行列積の前の行列はAの転置行列である。したがって第2のパラメータはこの転置行列の行次数を示すからAの行次数MではなくAの転置行列A'の行次数Lである。

行列積の演算命令では行列の行ベクトルまたは列ベクトルと行列との積を計算することができない。そのため行列の第1行ベクトル（または第1列ベクトル）と行列との積を求める演算命令がある。AはM行L列、BはL行N列の行列とし、Aの第1行ベクトルと行列Bとの積Cを計算する場合を考える。

VGVMILT, N, L, M, A, B, C

VGVMILTは第1行ベクトルと行列の算を計積する演算命令である。第1のパラメータであるNは行列Bの列次数、第2のパラメータであるLは行列Bの行次数、第3のパラメータであるMは行列Aの行次数をそれぞれ示す。いまAがL行M列の行列であるとき、第3のパラメータの値を1として

VGVMILT, N, L, 1, A, B, C

を実行すれば、Aは1行L列の行列とみなされてAの第1行ベクトルと行列Bとの積が計算されるから、これはAの第1列ベクトルと行列Bとの積を計算したことになる。

また、2つの行列の第1行ベクトル間の内積を計算する演算命令もある。A

がL行N列、BがM行N列の行列であるとき、Aの第1行ベクトルとBの第1行ベクトルの内積Pを計算する演算命令文は次のようになる。

I N P R O D , N , L , M , A , B , P

INPROD は2つの行列の第1行ベクトル間の内積を計算する演算命令である。第1のパラメータであるNは行ベクトルの次数、第2および第3のパラメータであるL、Mは2つの行列の行次数をそれぞれ示す。第2または第3のパラメータの値を1にすることにより第1列ベクトルとの内積を求めることができる。

さて、TSPには配列の加減計算をする配列演算命令または行列演算命令といったものはない。TSPでは配列または行列はこの節の初めにも述べたようにベクトル方式で記憶されており、すべて1次元配列の変数として扱われる。この1次元配列の変数の計算処理には数式命令 GENR がある。したがって配列演算および行列の加減演算はこの数式命令文を用いて行われる。例えば、A、B、Cを3行2列の行列とし、 $D = 3A - (B/2 + C)/4 + 1$ を計算する場合、数式命令文は次のようになる。

S M P L 1 , 6

G E N R D = 3 * A - (B / 2 + C) / 4 + 1

SMPLは数式命令で計算処理される1次元配列の要素を指定する命令であり、パラメータ部には最初の要素と最後の要素が示される。この場合、3行2列の行列の加減演算であるから要素数は6、したがって1次元配列では第1要素から第6要素までとなる。GENR は1次元配列の演算を行う数式命令であり、パラメータ部には加減乗除および巾乗、対数、指数、絶対値などの関数を含んだ数式が書かれる。

数式命令を用いて行列の任意の行または列ベクトルを第1行または第1列ベクトルとする部分行列を作ることが出来る。TSPでは1次元配列の添字は要素の相対位置を変える機能に用いられる。これはTSPが時系列分析を主目的としたシステムであるからである。添字の値が負のときは遅行（ラグ）を、正のときは先行を意味する。いま、Aは10行20列の行列とする。例えばAの第11列ベ

クトルを第1列ベクトルとするAの部分行列Bを作るにはAを1次元配列とみなしてその第101要素を第1要素にする1次元配列Bを作ればよい。

```
S M P L 1, 1 0 0
G E N R B = A ( 1 0 0 )
```

Aの添字の値が100であるから100要素先行した要素がBの要素となる。Bの第1要素はAの第101要素が、最後の要素であるBの第100の要素にはAの第200要素が代入される。

行列の積、次いで行列の加減演算をみてきたが、最後に行列の除演算である逆行列について調べよう。N次の正方行列の逆行列 $V = A^{-1}$ を計算する演算命令文は次のように書かれる。

```
M I N V, N, A, V
```

MINV は逆行列を計算する演算命令である。第1のパラメータであるNは逆行列の次数を示す。第2および第3のパラメータであるA, Vはそれぞれ行列および逆行列を示す。

以上がTSPにおける行列演算に関する基本的言語の内容である。TSPにはこれらの一般行列に関する演算命令のほかに時系列分析でよく用いられる対称行列、三角行列に関する行列積、逆行列の演算命令があり、さらに行列の正規直交化、行列の三角行列分解、行列の主成分ベクトルなどの変換命令がある。後者の変換命令は応用分析システムに通常組込まれている機能であり、行列の演算命令とは切り離して考えることにする。前者の対称行列、三角行列に関する演算命令はもちろん無くても一般行列の積や逆行列の演算命令でも計算することができる。しかし、行列の次数が大きい場合には計算能率および精度の上で非常に悪くなる。これら特殊行列の演算命令は次の頁の表に示されるような行列積および逆行列を計算するために用いられる。(TをN次の上位三角行列、SをN次対称行列AをM行N列の一般行列とする。)

さて、TSPにおける行列演算の言語は以上から分るようにFORTRANにおけるサブルーチン形式の表現に似ている。そのために如何なる行列演算をしているのかが判然とせず、使い易いとは言えない。また、行列の加減演算が行列

	特殊行列演算命令	行列積または逆行列
1	G 2 Y M L T, M, N, A, B	$B = A' A$
2	G T G M L T, M, N, A, T, B	$B = A T$
3	T G G M L T, O, M, N, T, A, B	$B = T A$
4	T G G M L T, I, N, M, T, A, B	$B = A T'$
5	T 2 Y M L T, N, T, B	$B = T T'$
6	T I N V, N, T	$T = T^{-1}$
7	Y I N V, N, S, T	$S = S^{-1} = T' T$

演算としてではなく1次元配列演算として処理されていることも行列演算に関する一貫性に欠けている。行列演算はやはりASTRO-FOILにおけるように演算式として表現する方が分り易く、しかも使い易い。他方、TSPは特殊行列の演算に対して特別な注意を払っている。ASTRO-FOILでは行列演算に関する数値計算上の問題には目を閉じた格好になっている。行列演算式の中で特殊行列をどのように表現するかが問題であろう。

TSPでは行列はベクトル方式で記憶されている。そのために拡大行列や部分行列を作るのに不便であり、また、分割行列の演算にも不適である。これに反してASTRO-FOILでは行列は行列方式で記憶されているので拡大行列や部分行列は容易に定義することができる。ベクトル方式での行列は記憶場所がより小さくすみ、行列の演算もより速い。また、行列演算には行列の次数を直接指定することができ、簡単で分り易い。行列方式の行列の場合、行列演算には行列の次数を指定するのではなく部分行列の範囲を指定しなければならないから、新ためて次数計算をして行列演算が成立するか否かを確認する必要がある。ベクトル方式による行列か、それとも行列方式による行列かの選択は、それぞれの方式に長所も短所もあり一概に優劣は決められない。

3. SIMPL における行列演算

SIMPL は筆者が昭和52年に開発した経営計測システム（「経営・経済データベースの制御分析システム」神戸大学経済経営研究所叢書，昭和53年）である。このシステムは，(1) TSP や SPSS のような汎用統計処理システムであるばかりでなく，(2) 社会経済情報の巨大なデータベースを内蔵した簡易データベースシステムでもあり，(3) これらデータベースの情報を時系列として，または横断面系列として抽出し，種々の分析を自由自在に行える時系列分析および横断面分析両用のシステムでもある。ここではこのシステムが備えている行列演算に関する言語のみを論ずることとする。

SIMPL における行列演算の言語は ASTRO-FOIL のように行列演算式による表現をとっている。しかし，行列の記憶方式は TSP のようにベクトル方式に依っている。その理由は行列演算式に対する行列の次数指定が直載簡明となるからである。次数の指定は次数命令 ORDER によって最高 3 つの行列の次数を指定することができる。

ORDER I S I BY L, K BY L, M BY N

1 番目の次数は行列演算式の左辺の行列の次数であり，2 番目の次数は 2 つの行列の積よりなる項における後の行列の次数であり，3 番目の次数は 3 つの行列の積よりなる項における中央の行列の次数である。

行列の演算は行列演算命令 **MAKES UP** によって行われる。

MAKES UP 行列 = 行列演算式

行列の演算式は次の規則にしたがってかなり自由に書くことができる。

- (1) 演算の優先順位を示す括弧を用いてはならない。
- (2) 式中に用いられるスカラー変数はベクトルの要素でなければならない。
- (3) 各項は最高 3 つの行列の積までとする。
- (4) 2 つの行列の積からなる項はすべて後の行列が同じ次数でなければならない。同様に 3 つの行列の積からなる項はすべて中央の行列が同じ次数でなければならない。

- (5) 式中の行列には転置操作または逆行列操作を自由に施すことができる。例えばAの転置行列はA([〃]), Aの逆行列はA(-)で表わす。
- (6) 単位行列, 定数行列(すべての要素が1である行列), 位置行列(K番目の対角要素が1で他の要素はすべて0の行列またはK番目の要素が1で他の要素はすべて0であるベクトル)の特定行列を用いることができる。単位行列はY(@), 定数項行列はY(*), 位置行列はY(K)で表わされる。
- (7) 対称行列と上位三角行列の特殊行列を指示するため行列の語尾に特殊記号#と@をそれぞれ付けることができる。

以上の規則を用いれば色々の行列を簡単に作ることができる。

(1) モーメント行列 $X^{(〃)} * X$	(5) 単位行列 $Y(@)$
(2) 逆行列 $X X \#(-)$	(6) 合計 $Y * Y(*)$
(3) 第K列ベクトル $X * Y(K)$	(7) 2次形式 $X^{(〃)} * A * X$
(4) 第K行ベクトル $Y(K) * X$	(8) 最小2乗推定 $X X(-) * X^{(〃)} * Y$

さて、次数文と行列演算文を用いてSIMPLにおける行列演算がどのように表わされるかを具体例でもって示そう。

(イ) 演算式が行列の積を含まない場合

```
ORDER IS M BY N
MAKES UP C=0.5 * A - B / 2 + 10
```

C, A, BはすべてM行N列の行列である。

(ロ) 演算式が2つの行列の積を含んでいる場合

```
ORDER IS M BY N, K BY N
MAKES UP Z=3 * F * X + G / 4
```

Z, GはともにM行N列, XはK行N列, FはM行K列の行列である。

(ハ) 演算式が3つの行列の積を含んでいる場合

```
ORDER IS K BY 1, K BY 1, K BY N
MAKES UP B=X X(-) * X^{(〃)} * Y
```

BはK行1列, $X^{(n)}$ はK行N列, $XX^{(-)}$ はK行K列, YはN行1列の行列である。次数文の2番目の次数がK BY 1となっているが, 式中に2つの行列の積は含まれていないから, 任意の次数であってもよい。

(二) 演算式が2つの行列積および3つの行列の積を含んでいる場合

ORDER IS 1 BY 1, N BY 1, K BY N
MAKES UP $UU = Y^{(n)} * Y - B^{(n)} * X^{(n)} * Y$

UUは1行1列, YはN行1列, $X^{(n)}$ はK行N列, $B^{(n)}$ は1行K列の行列である。

(三) 演算式がベクトルの要素を含んでいる場合

ORDER IS 1 BY 1, 1 BY 1, N BY N
MAKES UP $S = P * H * Q - T < J >$

Sは1行1列, HはN行N列, Pは1行N列, QはN行1列の行列である。また $T < J >$ はベクトルの第J要素を表わす。

以上がSIMPLにおける行列演算に関する言語の全容である。

むすび

SIMPLにおける行列演算に関する言語は, ASTRO-FOILおよびTSPにおける行列演算に関する機能と言語表現の長所を取り入れて改善し, しかも新しい機能として特殊行列および行列の要素を導入することによってベクトル方式による行列に内在する欠点を補っている。最後にSIMPLにおける行列演算の特徴をまとめておこう。

- (1) 行列演算は行列に関する数式で表わされ, 数式は線型代数の教科書で見られるような式に近い形で書かれる。ただし, 優先順位を示す括弧は用いることができない。また, 行列の積は3重積までである。
- (2) 転置および逆行列操作を自由に施すことができる。
- (3) 単位行列, 定数行列(すべての要素が1である行列), 単位ベクトル, 位置行列(K番目の対角要素が1で他の要素はすべて0の行列)を前以て

定義することなく用いることができる。

- (4) 行列が対称行列または三角行列であることが分っているとき，行列の語尾に特殊記号を付けることによって行列演算における計算の速度と精度を高めることができる。

執筆 者 紹 介

能 勢 信 子……教 授・経営計測部門 経営学博士

下 條 哲 司……助教授・海事経済部門

定 道 宏……助教授・経営計測部門

伊 藤 駒 之……助教授・経営機械化部門

都 藤 希 八 郎……講 師・経営機械化部門

生 島 芳 郎……講 師・附属経営分析分献センター

民 野 庄 造……助 手・経営経理部門

内 田 幸 夫……助 手・経営計測部門

米 花 稔……福山大学教授経済学部 経営学博士
神戸大学名誉教授

杉 浦 一 平……和歌山大学教授経済学部

萩 野 典 宏……甲南大学助教授経営学部

布 上 康 夫……甲南大学講師経営学部

経営機械化叢書(既刊) 目次

第1冊	経営機械化技術論	昭和27年刊
第2冊	会計機械化研究	昭和31年刊
第3冊	経営事務機械化の諸問題	昭和35年刊
第4冊	経営機械化と経営機構	昭和36年刊
第5冊	経営機械化とシステム研究	昭和37年刊
第6冊	EDPSの発展と経営上の課題	昭和38年刊
第7冊	経営機械化研究の新動向	昭和39年刊
第8冊	データ処理と情報検索	昭和40年刊
第9冊	経営機械化と管理情報システム	昭和42年刊
第10冊	経営機械化システムの諸研究	昭和43年刊
第11冊	情報システムの展開	昭和44年刊
第12冊	電子計算機室の構造と管理	昭和47年刊
第13冊	経営機械化の発展とデータ処理	昭和47年刊
第14冊	経営機械化の発展と情報システム	昭和48年刊
第15冊	経営機械化の発展と情報検索	昭和49年刊

第16冊 経営・経済情報分析システムの新展開 昭和50年刊

わが国の経営機械化研究の展開	米	花	稔
経営・経済情報制御分析システム—BEICA—	米	花	野
経営・経済データ・バンクとその制御システム—BEIC—	米	野	庄
経営経済情報制御分析システムのデータ・バンクの統計資料について	能	勢	信
計量経済模型分析のためのプログラム・システム—STEPS—	杉	浦	一
視覚プログラミング・システム—SPEED—	定	道	正
HITAC-10の通信制御モニターの開発	杉	浦	一
HITAC-10(小型機)を端末機または衛星機とする利用方式と交信用	定	道	正
FORTRAN サブルーチンの開発	定	道	正
HITAC-EDOS用簡易 Job Control Language について	定	道	正
EDP 部門平面の試作	都	野	希
<資料>			
HITAC-8350 システムおよび EDOS-JCL の一般解説	民	野	庄
	安	藤	昭

第17冊 現代情報システムの研究 昭和51年刊

オーストラリアにみる情報システムに関する若干問題—視察メモ—	米	花	稔
STEPS FOIL の構造	杉	浦	一
基本統計処理システム—SIMPLE-ATLASS—	民	野	庄
EDP部門の防災について	都	野	希
<資料>			
社会科学とデータバンク—その資料的側面—	生	鳥	芳
汎用簡易作表プログラム CROTAB について	下	條	哲
BEICA システムの初心者用利用マニュアル—SIMPLE—	都	民	野
SPSS 利用者のための手引き	定	道	道
企業財務データのコード表	定	民	野
			庄

昭和53年3月25日発行

(非売品)

神戸市灘区六甲台町2

編集者
発行者

神戸大学経済経営研究所

神戸市生田区中山手通7丁目66

印刷所

有限会社 興文社

電話 078-341-4113・4740

Development of the Management Information Systems

CONTENTS

Supplement on the Development of Management Information System in Japan —1960's—	Minoru Beika
On the Norwegian Microdata File System	Nobuko Nosse
On the Locational Distribution of EDPS in Osaka City	Kihachiro Tsudoh
Design and Construction of the Thesaurus of Accounting Terms	Yoshiro Ikushima
Comment on Bazaraa Algorithm.....	Komayuki Itow
A Manual-free Program — On the Dialogue Cross Tabulation Program —	Tetsuji Shimojo
Strategic Marketing Simulation System	Yasuo Nunokami Tsunehiro Hagino
OPCON : Optimal Control Simulation Program	Yukio Uchida
Syntax Description Language FOIL V-10	Ippeï Sugiura
Sort, Edit, and List Files	Shozo Tamino
How to Incorporate Matrix Operation in Application Programs	Hiroshi Sadamichi

THE RESEARCH INSTITUTE FOR ECONOMICS
AND BUSINESS ADMINISTRATION
KOBE UNIVERSITY

1 9 7 8