

1988年5月発刊

Vol. 8 No. 1

特集：マイクロ・プロダクト

巻頭言

特集「マイクロ・プロダクト」の発刊によせて……………山崎斐雄 1

論 文

マイクロプロセサとワークステーションの技術動向……………野本雄一 3
情報処理システムにおける

マイクロ・プロダクトの位置づけに関する考察 …………… 佐藤 博 22

MAPPER 1100 におけるマンマシン・インタフェース

の改良——直接操作の導入 …………… 伊東春正 34

オフィス・ワークステーションのマルチメディア化動向……………粟科紀夫 48

UNISYS DS 7 におけるコンカレント DOS XM…………… 阿部比呂志 59

MML の実現……………河合昭夫 71

オフコンにおける大学図書館システムの実現……………斎藤ひろみ 80

CAI システム LEARN UP における

オーサリングの特徴と今後の課題 …………… 出石 修 94

UNIX マシン UNISYS 5000/70 上での第 4 世代言語

LINC の実現 ……………佐々木政和 107

統合ネットワーク指向インテリジェント端末の開発

——伊藤ハム(株)における事例…………… 堀籠順之助 125

花王(株)における人事システム専用端末 ……………南瀬末彦 138

UNISYS DS 7 用プリンタでのバーコード印字 ……………北崎 勲 151

図書紹介 ……………162

掲載論文梗概 ……………表 2, 3

最近のマイクロプロセッサ技術の進歩は、32ビットmpuに代表されるようにめざましいものがある。32ビットmpuを搭載したワークステーションの進歩も著しく、今後のコンピュータ・システムに与える影響は大きい。野本雄一のマイクロプロセッサとワークステーションの技術動向は、本特集号のテーマであるマイクロ・プロダクトの概念を規定し、さらにマイクロ・プロダクトの典型であるワークステーションと、その中核になるマイクロプロセッサに焦点をあてて解説をしている。

情報処理システムを設計する際、マイクロ・プロダクトへの機能および負荷分散を検討することの重要性が、高まってきている。佐藤博は、情報処理システムにおけるマイクロ・プロダクトの位置付けに関する考察の中で、その理由を経済的側面（マイクロ・プロダクトの量産性のメリット）と応用的側面（マイクロ・プロダクトの集積・小型性のメリット）から説明している。

マンマシン・インタフェースの動向は、マルチウィンドウ、マウス、アイコン等による直接操作とMML (Micro Mainframe Link)、システム統合化技術等ソフトウェア・インタフェースとの統一の方向にある。伊東春正のMAPPER 1100におけるマンマシン・インタフェースの改良——直接操作の導入は、ソフトウェア面を主たる対象にしてマンマシン・インタフェースの現状を報告し、マンマシン・インタフェースを設計するうえでの考慮すべき事柄を述べている。

オフィスで扱う情報は多様なメディアに記録して利用されている。近年これらのメディアを一つのオフィス・ワークステーションに統合する動きが顕著である。薬科紀夫のオフィス・ワークステーションのマルチメディア化動向は、オフィス・ワークステーションをマルチメディア化という視点から、使用者の立場で考察し、その現状と今後の動向について解説している。

最新の多機能デスク・ステーションDS7は、処理速度向上のため、マイクロプロセッサをIntel社

製の80186から80286に変更するとともに、メモリ容量拡大のため、OSをDigital Research社のコンカレントCP/MからコンカレントDOS XMに変更した。阿部比呂志は、UNISYS DS7におけるコンカレントDOS XMの中で、1メガ・バイトの壁を破るためのハードウェアとファームウェアのアーキテクチャの実現方法を中心に、コンカレントDOS XMのDS7への移植において採用した技術的手法について報告している。

情報処理の方式が、集中処理から分散処理へ移行しつつある。近年、ワークステーションが高機能化するとともに、今までメインフレームで行っていた作業を分担できるようになってきた。この作業分担を円滑に行うための一つの基礎技術がMMLである。河合昭男は、MMLの実現の中で、その一つの事例として、DS7上のイメージ処理プログラムU-IMAGEをとりあげ、解説している。

中小規模の大学図書館では予算的にみてオフコン・クラスのコンピュータによる機械化ニーズが圧倒的に多い。斎藤ひろみは、オフコンにおける大学図書館システムの実現の中で、UNISYSシリーズ8における中小規模大学図書館のトータルシステムの概要、および今後の大学図書館システムの方向について述べている。

文部・通産両省がCAI利用に積極的な姿勢を示した事により、CAIシステムは一段と吟味されることになった。とくに、オーサリング機能の充実が重要視されている。日本ユニシス（当時日本ユニパック）では、昭和62年にCAIシステム「LEARN UP」のオーサリング機能を増強すべく、新LEARN UPシステムの開発を行った。出石修は、CAIシステムとLEARN UPにおけるオーサリングの特徴と今後の課題の中で、オーサリング機能増強のために講じた方法、および今後必要になると思える機能について述べている。

LINCは、現在、MCP (Master Control Program, UNISYS A/B/V シリーズのOS) の下

特集「マイクロ・プロダクト」の発刊によせて

山崎 斐雄

ユーザ・ニーズの多様化という言葉が使われるようになって久しい。また個性の時代という認識もすっかり定着している。情報処理の分野でというより、社会生活のすべての面にわたって少量多品種の時代、標準品や規格品を嫌う時代である。

マイクロ・プロダクトあるいはエンドユーザ・ファシリティが注目を集め、この領域での商品開発に大きな努力が払われ続けているのは、このような社会的な背景からである。比較的少ないコストで、比較的短時間で商品が開発できるようになった。さまざまな要求に対応しやすく、新しいアイデアを商品として手軽に実現できることが現実となって、ベンチャー企業の参入が多い分野でもある。

しかしながら、情報処理システムの開発という立場から考えると、とくにソフトウェア開発にかかるコストの増大、システム・エンジニアの不足は依然解決されない問題として残っており、その開発力が企業の競争力そのものに大きな影響力を与えるといわれる。

絶対的に不足しているシステム・エンジニアの生産性を上げるために、技術の蓄積はもとより、重複した努力を避けるために標準化に対する社会的要請がますます強くなっている。利用者のために、個々の要求を大切にす。一方、全体システムの統合を考える時には、標準化を積極的に推進することがわれわれの置かれている環境であると認識している。

このような環境のもとでマイクロ・プロダクト、エンドユーザ・ファシリティに期待されることは、利用者に対して機能的にも、時間的にも、経済的にもきめ細かな要求に応えられることであり、同時にメインフレームを中心とする全体システムの中で標準的な手順ののっとり調和良く他のサブシステムと連携できることである。

また、最近システム・インテグレーションという言葉がよく使われるようになった。できるだけ多くの要求を取り込んだ巨大な一つの汎用システムを作るよりも、個々のニーズに合った比較的小さな専用のサブ・システムを複数個作り、それらを全体として調和良く、効率良く結合して一つのシステムとして統合するということである。

このような観点からもマイクロ・プロダクト、エンドユーザ・ファシリティの果たす役割は大きい。マイクロ・プロダクト、エンドユーザ・ファシリティを利用して個々の利用者のニーズを十分に吸収し、それをメインフレームを中核として一つのシステムに統合する、これが最近着実に進みつつあるシステム化の方向である。

マイクロ・プロダクト、エンドユーザ・ファシリティに対する重点的な施策、相互接続技術/オープン・アーキテクチャに対する積極的な取り組み、メインフレームを中核とした大規模システム開発の技術の蓄積とその活用、さらにそれらを総合したシステム・インテグレータであ

2 (2)

ることの標榜等々、すべてこのような状況に対する当社の認識の具体的なあらわれである。

当社は本年4月1日に日本ユニパック株式会社とパロース株式会社が合併、日本ユニシス株式会社として新しい歴史に向かって進み始めた。過去にそれぞれ独自の商品群を持ち、独自の技術を蓄積してきた二つのグループが融合することによってお客様の情報処理部門、およびエンドユーザの皆様に対して、従来にも増してより適合性の高い商品群と調和のとれた技術支援を提供できることになった。お客様のニーズに適確に応え、システム・インテグレータ、トータル・サプライヤとしてお客様、ひいては社会にますます貢献できるよう、さらに努力を続けていきたい。

(プロダクト開発本部長)

マイクロプロセサとワークステーションの技術動向

Trend of Microprocessor and Workstation

野 本 雄 一

要 約 最近マイクロプロセサ技術の進歩は 32 ビット mpu (Microprocessor Unit) に代表されるようにめざましいものがある。32 ビット mpu を搭載したワークステーションの進歩も著しい。

そして、それらが今後のコンピュータ・システムに与える影響は大きい。そこで、これらの諸技術の動向を概観し、見解を述べる。

Abstract Recently, the technology of microprocessor has made a remarkable progress in such area as 32 bits mpu chip. In the meanwhile a workstation with 32 bits mpu has improved rapidly.

These technologies have great impact upon the computer systems. Therefore the author reviews a general trend of these technologies and expresses his opinion.

1. ま え が き

マイクロプロセサ, すなわち LSI になった cpu チップはコンピュータをはじめとする情報処理機器に広く使われている。パーソナル・コンピュータ (パソコン), ワークステーション, 高インテリジェントな端末装置や周辺装置などのマイクロプロダクトは, マイクロプロセサなしには存在しえない。

今日, 一般の 32 ビット・マイクロプロセサの cpu 能力をみれば, 1970 年代の汎用コンピュータの最上位モデルに匹敵するほどの性能を持っており, その進歩はめざましい。最近, マイクロプロダクトの一つとしてワークステーションと呼ばれるコンピュータが話題になっている。既存の 16 ビット cpu をもったパソコンとは一線を画した, れっきとしたコンピュータである。

ワークステーションはマイクロプロセサを用いた代表的な製品であり, 今後のコンピュータ・システムに与える影響はたいへん大きい。

そこで, 本稿ではマイクロプロダクトの概念を規定した後で, マイクロプロダクトの典型であるワークステーションと, その中核になるマイクロプロセサに焦点をあてて解説をする。

2. マイクロプロダクト

2.1 マイクロプロダクトの範囲

マイクロプロダクトとは何かを規定することはむずかしい。しかし, おおむね次のようなものと言うことができる。

- 1) 市販 (たとえば, インテル, モトローラ) やカスタムメイドの cpu チップを使用している製品
- 2) パーソナルな利用を第一の設計目標としている製品で, 汎用コンピュータ (メ

インフレーム、ミニコンピュータ、オフィスコンピュータ) でないもの

3) 高機能な端末装置(たとえば、グラフィック・ディスプレイや POS*ターミナル) すなわち、「マイクロプロセサを使用した、パーソナル・システムや端末装置」ということができる。

パーソナル・システムとしては従来からのパソコンが想起されるが、32ビット cpu チップの普及に伴い、マイクロプロダクトは一人前のコンピュータとして、またトータルなコンピュータ・システムの重要な構成要素として「ワークステーション」と呼ばれるようになった。今やマイクロプロダクトとはワークステーションに代表されるといってよい。

2.2 ワークステーション

ワークステーションは、パソコンとミニコンやオフコンの中間に位置するもので、32ビット cpu (マイクロプロセサ) を使用しているものが主流になっている。cpu は 1~10 MIPS という高性能を有しており、スタンドアロンで十分な処理ができる。また、ワークステーション同士を LAN (Local Area Network) で結ぶ水平型接続や、メインフレームと階層的に結ぶ垂直型の接続が可能である。

ワークステーションは作業卓といわれるように、ある個人がコンピュータを独占して使用できるため、身の近くに置かれる装置である(図1)。このためワークステーション自身が高い能力を持つと同時に、自らの能力を超える処理を必要とするとき、他のリソースをあたかも自分のリソースであるように使用できることが必要な条件となる。

ワークステーションを用途から分類すると、大きくエンジニアリング用とオフィス(ビジネス)用とに分けられる。

1) エンジニアリング用を EWS (Engineering Workstation), あるいは TWS

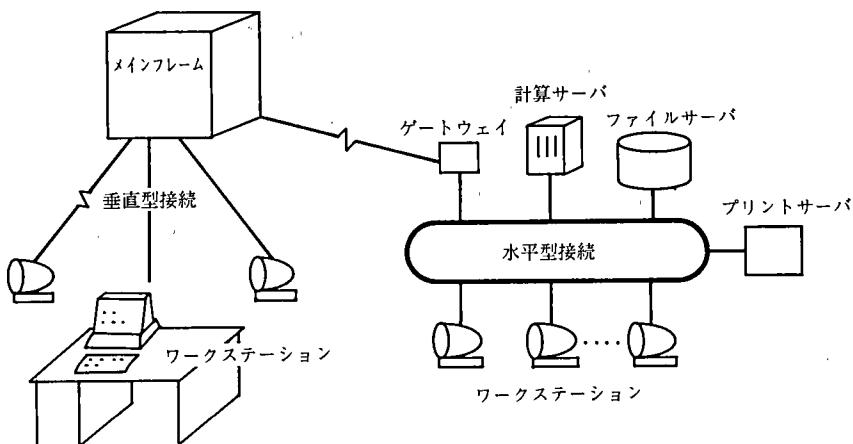


図1 ワークステーションのイメージ

Fig.1 Image of workstation

(Technical Workstation) という。CAD/CAM^{*}、CAE^{**}、プログラム開発といった技術的な分野で利用されている。エンジニアがミニコンピュータをしのぐ処理能力を個人で占有できる環境を提供することが目標であり、高性能化が急である。加えて、

- ① グラフィック処理の高速化を追求した GWS (Graphic Workstation)
- ② AI の研究や開発のための専用 cpu を搭載した AIWS (AI Workstation)
- ③ ソフトウェアの生産性向上のため、共通 OS (ΣOS^{***}) を搭載した ΣWS (ΣWorkstation)

など、特化された EWS がある。

2) オフィス用を OWS (Office Workstation) という。

OWS は OA (Office Automation) を構築するために利用される。OWS には次の二つのタイプがある。

- ① 文書作成や事務処理に加えイメージ (画像) の蓄積・検索・処理ができ、ワークステーション間を高速で通信できる機能をもった狭義の OWS
- ② 従来のインテリジェント端末装置とパソコンが融合したもので、端末装置のパソコン化あるいはパソコンの多機能端末化と考えられる製品である。現在 OA の中核をなしていることからこれを OAWS と呼ぶ。

OAWS は、汎用機とマイクロ・メインフレーム・リンク (MML) などの垂直型分散機能によってサポートされる。

ワークステーションの種類の関係は図 2 に示す通りで、各々の概要を表 1 に示す。

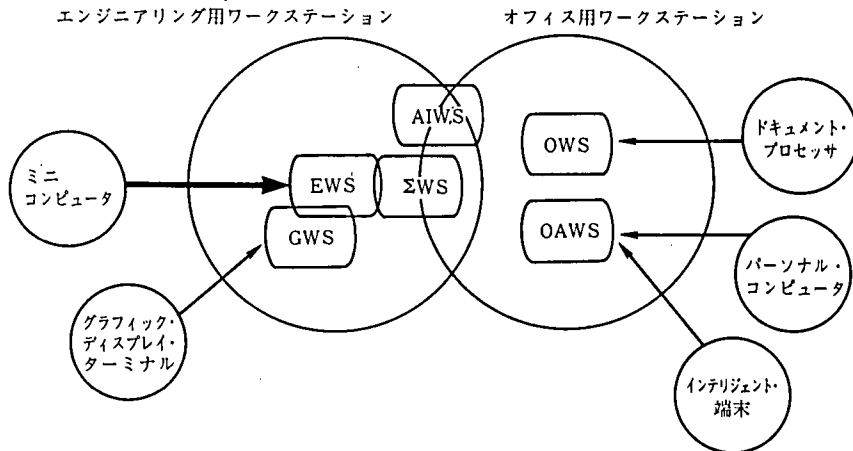


図 2 ワークステーションの種類

Fig. 2 Type of workstation

* CAD/CAM : Computer Aided Design/Computer Aided Manufacturing

** CAE : Computer Aided Engineering

*** ΣOS : SIGMA (Software Industrialized Generator & Maintenance Aids) Operating System

情報処理振興事業協会 (IPA) の Σ (シグマ) プロジェクトで提案されているソフトウェア開発用のワークステーション ΣWS の標準 OS

表1 ワークステーションの分類
Table 1 Taxonomy of workstations

EWSの概要 ¹⁾	
用途	CAD/CAM, CAE, システム開発・運用
特徴	高解像度ディスプレイやマウスを装備しており、ユーザ・インタフェースが優れている。業界標準を多用
処理性能	約1～4 MIPS
CPU	MC 68020 搭載機が多い(FPUを内蔵)
主記憶	4 M～32 MB
補助記憶	HD(70 MB～), FD, CRMT
解像度	1,024×800以上(ビットマップ, マルチウィンドウ)
LAN	Ethernetを標準サポート
OS	UNIX(System V, 4.2 BSD)
価格	約100万～1500万円

GWSの概要 ¹⁾	
用途	CAD/CAM, CAE
特徴	高解像度ディスプレイを装備しており、2次元または3次元のグラフィック処理装置を内蔵している。
処理性能	約1～10 MIPS
CPU	MC 68020 搭載機が多い(FPUを内蔵)。
主記憶	4 M～32 MB
補助記憶	HD(70 MB～), FD, CRMT
解像度	1,280×1,024が標準(ビットマップ, マルチウィンドウ)
LAN	Ethernetを標準サポート
OS	UNIX(System V, 4.2 BSD)
価格	約500万～2000万円

ΣWSの概要 ¹⁾	
用途	システム開発
特徴	共発OS利用によるソフトウェア開発の生産性向上を目指しているハードウェアはEWSとほぼ同じ
処理性能	約2～4 MIPS
CPU	MC 68020 搭載機が多い(FPUを内蔵)
主記憶	4 M～32 MB
補助記憶	HD(70 MB～), FD, CRMT
解像度	1,024×800以上(ビットマップ, マルチウィンドウ)
LAN	Ethernet
OS	ΣOS
価格	約400万円～

AIWSの概要 ¹⁾	
用途	AI研究・開発
特徴	AI用言語を独自プロセッサで高速に実行する。アプリケーションへの応用はこれから
処理性能	Prolog系で約300 KLIPS(リスト処理)～
CPU	独自プロセッサ
主記憶	4 MB～
補助記憶	HD(200 MB～), CRMT
解像度	1,024×800以上
LAN	Ethernet
OS	LISP, Prologに依存
価格	約400万円～

OWSの概要 ¹⁾	
用途	統合オフィス処理, デスクトップ・パブリッシング
特徴	EWSなみの処理性能をもつハードウェアにオフィス処理用ソフトを搭載
処理性能	—
CPU	独自プロセッサ
主記憶	約2 MB～
補助記憶	HD, FD
解像度	1,024×800以上(ビットマップ, マルチウィンドウ)
LAN	Ethernet
OS	独自OS
価格	約300万円～

OAWSの概要 ¹⁾	
用途	事務処理, 文書作成, 表計算など
特徴	高性能パーソナル・コンピュータ(16, 32ビット)にネットワーク機能やホスト接続機能を付加している。
処理性能	約1.5～3.0 MIPS
CPU	80286, 80386
主記憶	1 M～16 MB
補助記憶	HD(20 MB～), FD
解像度	640×400以上
LAN	独自LANが多い
OS	独自OS(マルチタスク), MS-DOS
価格	約50万～300万円

(“ワークステーションの分類”, 日経データプロ ワークステーション, 日経マグローヒル社, WSI-100-051～056より)

3. マイクロプロセッサの歴史

3.1 マイクロプロセッサの誕生

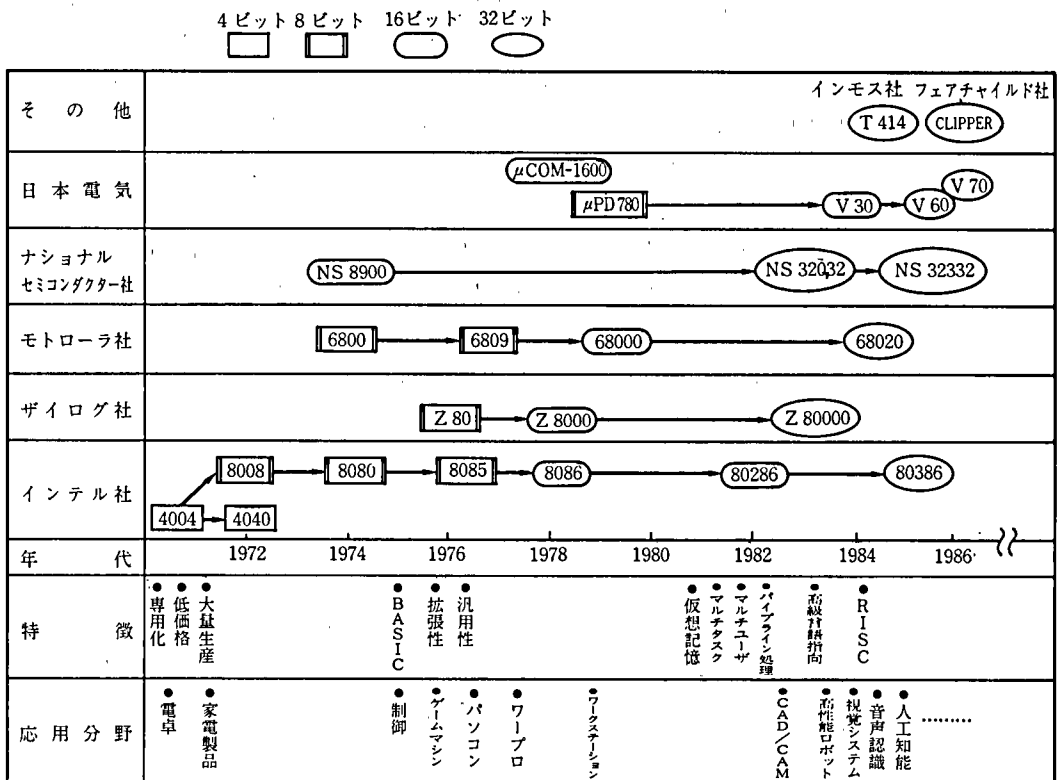
マイクロプロセッサの起源は電卓に始まる。電卓がトランジスタ式から IC(集積回路)式へ、さらに LSI(大規模集積回路)化されるに及んで大幅なコストダウンを実現していった。カシオやシャープが電卓専用の LSI を採用し始めたのが 1970 年頃である。

そのころ米国のサンフランシスコの近くにシリコンバレーと呼ばれる町サニーベールに、インテル (1969 年設立) という半導体メモリを生産する小さな会社があった。また、当時日本に輸出用電卓専業のメーカ、ビジコン社があった。1973 年のドルショックで倒産の憂き目に逢ってしまったが、いろいろな電卓を一つの石 (IC) で作ってしまおうとするユニークな発想をもち、この石の開発をインテルに依頼した。そして、*i*4004 (4 ビット) という世界初のマイクロプロセッサが生まれた。この功績は実に大きかった。これと併行して *i*8008 (8 ビット) が開発され、初めて世に発表されるのが 1971 年の 12 月である。

それから 11 年余り、マイクロプロセッサは急速の進歩をとげ、旧来のコンピュータに大きな革命をもたらすに到った。

3.2 マイクロプロセッサの進化

インテルが *i*8008 を世に出して以来、モトローラやナショナルセミコンダクタ、ザ



(鈴木武夫, "32ビットマイクロセッサの現状と動向",
コンピュータール16, コロナ社, 1986年10月10日より)

図 3 マイクロプロセッサの発展過程
Fig. 3 Progress of microprocessor

イログが続々と参入し各社競って高性能化をはかってきた (図 3)。

マイクロプロセサ (mpu) の発展を追ってみると、

- 1) 4ビットのマイクロプロセサが1971年に出現し、高級電卓に使用される。
- 2) 8ビットのマイクロプロセサが1972年にインテルから発表され、モトローラ、ザイログが続いて参入。
 - ・インテルの8008をスピード・アップした8080が実用化される。8085は制御機器用として使われた。
 - ・ザイログのZ80がヒットし、8ビットの主流となる。
 - ・モトローラは、6800の次に出された6502がパイプライン方式による高速化はかられ、APPLEに使われた。

8ビットmpuは制御用に多用されたのをはじめ、マイコンブームを引き起こし、マイコン・キットが発売され、誰にでも手軽にコンピュータが作れるアマチュア・エンジニアリング時代を開いた。従来、大きな資金と多数のエンジニアを必要としたコンピュータの開発が一転して、マニア達がどこでも手軽に作れる時代となり、ガレージ・カンパニーとか、四畳半メーカといわれる小規模な企業(ベンチャー)が輩出するきっかけになった。

- 3) 16ビットmpuが本格的に出荷されるのが1978年に入ってからである。当時のミニコン程度の性能を持っていたことから、パソコン、インテリジェント端末、グラフィック・ディスプレイなどに大きな変革をもたらした。そしてワークステーションと称されるコンピュータが1980年に出現する。ミニコン並の性能をパーソナルに占有して使用でき、ビットマップ・ディスプレイとアイコン操作による使いやすいマンマシン・インタフェース、LANを介しての水平型結合など斬新な設計思想が大学や研究機関で大きな関心を集めた。

16ビットmpuは多くのメーカから出荷されたが、現在汎用のmpuとしてはインテルの80286とモトローラの68010が、世界のシェアを2分しているといつてよい。

- 4) 32ビットmpuは、1984年にモトローラ68020とインテル80386の発表を機に一斉に32ビットが主流となる。32ビットmpuの高速化は急で、半年ごとに新しいバージョンが出荷される状況になっている。

4. 32ビット・マイクロプロセサの動向

4.1 32ビット・マイクロプロセサの出現

32ビット・マイクロプロセサは、モトローラやインテルを始めとするいくつかの半導体メーカから市販されている (表 2)。32ビット・マイクロプロセサのハードウェア的特徴は、主に処理性能の向上とアドレス空間の拡大にある。

- 1) 処理性能の向上……32ビット・マイクロプロセサになると、次の要素が組み合わさって処理性能が著しく向上した。
 - ① 16ビットに比べ、データ幅が2倍になり、プロセサとメモリ間のプログラム・フェッチ回数が少なくなる。
 - ② データや浮動小数点演算など32ビット単位で処理できる。

表2 主な32ビット・マイクロプロセッサ
Table 2 Typical 32 bits microprocessor

型 名	68020	68030	80386	R 2000	CLIPPER	V 70	TRON チップ
メーカ	モトローラ	モトローラ	インテル	MIPS コンピュータ	フェアチャイルド	日本電気	日立 (三菱, 富士通)
集積素子数 (概数)	23万7千	30万	27万5千	10万	13万2千	40万	70万
クロック周波数 (MHz)	12.5, 16.7 20, 25	16, 20, 25	12, 16	16, 20	33, 40	16	20~25
仮想記憶方式	ページング	ページング	ページ化 セグメント	ページング	ページング	ページング	ページング
内蔵MMU (メモリ管理ユニット)	外付け	内蔵	内蔵	内蔵	外付け	内蔵	内蔵
ページサイズ	256~32K	256~32K	4K	4K	4K	4K	4K
内蔵キャッシュ・ メモリ	命令用256 バイト	命令/データ用 各256バイト	外付け	外付け	外付け	外付け	命令1Kバイト スタック128バイト
内蔵FPU (浮動小数点 演算機構)	外付け	外付け	外付け	外付け	内蔵	内蔵	外付け
汎用レジスタ	16	15	8	32	16×2	32	16
サンプル出荷時期	1985年	1987年第四半期	1985年	1987年	1986年	1987年6月	1988年前半

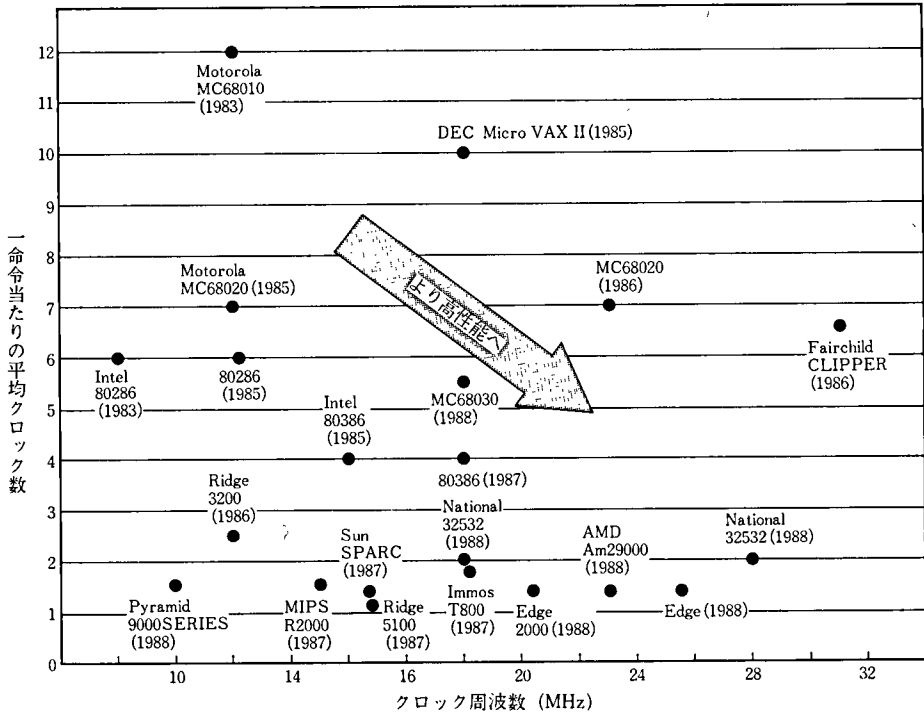
- ③ クロック周波数の向上
- ④ 制御のパイプライン化
- ⑤ MMU (メモリ管理ユニット), キャッシュ・メモリのオン・チップ(内蔵)化
- ⑥ FPU (浮動小数点演算機構) のオン・チップ化などによって, 総合的に向上している。

ワークステーションの中で, とくにEWSは科学技術計算やグラフィックスの処理のために, 常に高性能であることが求められる。OWSにおいてもEWS程でないにしても, ネットワーク制御や高級言語処理さらにイメージ処理のために高速化の要求は強い。

図4に見られるように, 1命令当たりのクロック数を減らし, クロック周波数を上げることによって高性能化が進んでいる。ちなみにモトローラの68020では, クロック周波数と演算能力の関係は表3のようにいわれている。

最近の高性能なマイクロプロセッサを搭載したEWSの中でも, とくにGWSの高性能化がめざましい。一例を表4にあげるが, さらにモトローラ68030の採用や独自のマイクロプロセッサを使うことによって一段と高性能化が予定されている。

- 2) アドレス空間の拡大……アドレス空間は, 物理アドレス空間はすべて4G (ギガ: 10^9) バイトまで可能となり, 論理アドレス空間もほとんど4Gバイトまで拡大された。インテルの80386では論理アドレス空間が64T (テラ: 10^{12}) バイト



(T.Manual 他, “火がついた小型コンピュータ高速化競争”,
日経エレクトロニクス, 1987.10.19, pp.163~178 より)

図 4 高速化されるマイクロプロセッサ

Fig. 4 Evolution of microprocessor's processing speed

表 3 モトローラ68020のロック周波数と演算能力

Table 3 Clock frequency and computing speed of MC 68020

クロック周波数(MHz)	演算能力(MIPS)
16.7	1 ~ 2
20	2 ~ 4
25	3 ~ 5

表 4 最近の高性能 GWS

Table 4 High performance GWS

製 品	CPU (クロック周波数)	FPU (クロック周波数)	演 算 能 力 〔メーカ公称値〕	主 記 憶 (最大サイズ) M バイト
A	MC 68020 (20 MHz)	MC 68881 (25 MHz)	4 MIPS	4 (32)
S	MB 86900 ^① (16.67 MHz)	Weitek 1064 ^④ Weitek 1065	10 MIPS	8 (128)
I	R 2000 ^② (12.5 MHz)	Weitek 1064 ^④ Weitek 1065	10 MIPS	8 (16)
H	独自開発 ^③	独自開発	8 MIPS	8 (48)

①: RISC アーキテクチャ, 富士通マイクロエレクトロニクス社製

②: RISC アーキテクチャ, MIPS コンピュータ・システムズ社製

③: RISC アーキテクチャ

④: Weitek 社のチップ・セット, 性能は単精度で1.6 MFLOPS, 倍精度で1.1 MFLOPS

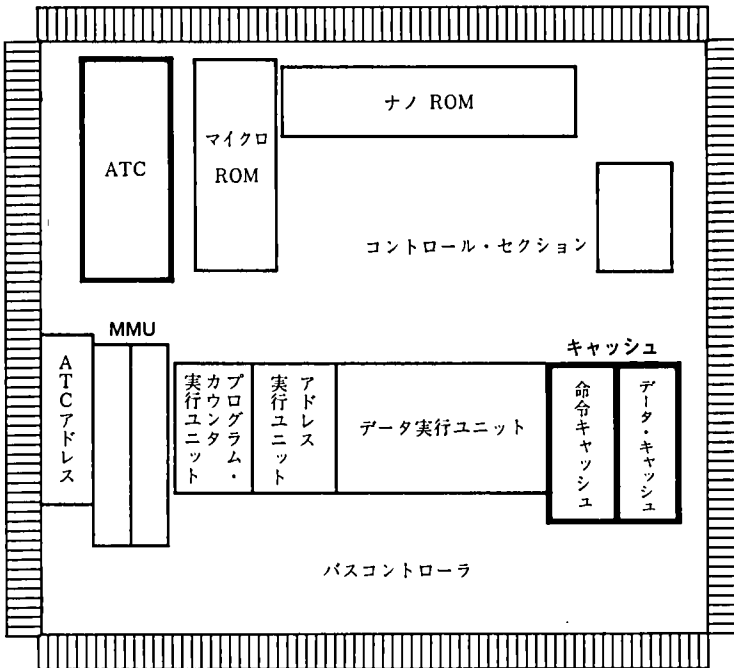
と大きい。これは単に主記憶容量の拡大を可能にするだけでなく、仮想記憶空間の増大と効率的利用を可能とする。また光ディスクなど大容量記憶装置の接続を容易にする。

また、実装メモリ容量が増大し、EWS では表4に示すように基本で4~8 M バイト、最大128 M バイトのものまであり、大型汎用コンピュータ並である。メモリ管理は仮想記憶方式が採用されているが、実行プログラム・サイズが大きくなり、外部メモリとのページング頻度が多くなると急激に効率が低下する傾向があり、実メモリの大きさが効率上重要な要因になる。

4.2 32 ビット・マイクロプロセッサの技術的背景

- 1) 集積度の増大……32 ビット化し処理性能を上げるためには、1チップに集積する素子数を増大させる必要がある。また、チップ集積度を増大させるためには、回路線幅の極細化、チップサイズの拡大、回路やレイアウトの工夫といった多くの要因の改良が必要である。集積度の進歩は2~3年で2倍の進歩である。現在の32 ビット・マイクロプロセッサは、回路線幅1.2~1.5 μm(ミクロン)、チップサイズは9~10 mm 角、CMOS 回路で、20 数万素子以上を集積するのが普通である^[2]。

また1チップ上にメモリ管理ユニットやキャッシュ・メモリ、浮動小数点演算機構も集積し、アドレス変換のクロック・サイクルを削減したり、遅延時間を減少することによって効率向上を計っている。図5にモトローラ 68030 のチップのレイアウトを示す。68030 は68020 に加えて、メモリ管理ユニットとデータ・キャ



ATC: Address Translation Cache(アドレス変換キャッシュ)

図5 68030 チップのレイアウト

Fig. 5 Layout of 68030 chip

ッシュを内蔵している。

- 2) ASICへの対応……ASIC (エイシック: Application Specific IC) の台頭は、マイクロプロセッサに大きなインパクトを与える。

ASICとは、明確な定義はなく、特定用途向けICと訳されることから、カスタムICや各種専用のLSIさらにPLD (Programmable Logic Device) まで含むこともある。しかし、「IC設計CAD」システムを用いて自分専用のLSIを容易に作れるゲートアレイや、スタンダードセル方式によるセミカスタムLSIととらえるのが一般的であろう。

汎用の標準LSIのように、フルカスタム設計によってマイクロプロセッサを開発するのに比べて、チップ面積の効率的利用を犠牲にしなければならないが、設計期間の短縮や設計コストを減少することができる。多品種少量生産や商品寿命の短縮に対応するため、ASICによるマイクロプロセッサの開発が盛んになると予想される。

ASICが普及すると、今の標準的マイクロプロセッサはコア・マイクロプロセッサとして位置づけられ、周辺に独自のプロセッサをASICで付加したチップが出現すると予想される。付加される機能としては、OSI*における各種通信プロトコルや、グラフィックスの標準ソフトであるGKS**など、アプリケーションに応じた基本ソフトのチップへの組み込みが考えられる。

4.3 RISCアーキテクチャの台頭

マイクロプロセッサにRISCアーキテクチャを採用する傾向が強くなってきた。RISCとはReduced Instruction Set Computerの略で、命令セットを簡素化した命令数の少ないコンピュータである。これに対し、従来の命令セットをもったコンピュータをCISC (Complex Instruction Set Computer) という。

- 1) RISCの登場……LSI技術の進歩に伴い、複雑な論理回路のLSI化が可能になった。このため、コンピュータの高速化は、命令を高度にし、複雑にすることによって実現されてきた。しかし複雑化することによって、あまり使わない命令が多くなったり、命令実行時のチェックなどのオーバーヘッドがかかったり、命令のデコード時間が長くなったりして実行時間を長くしてしまうことがある。RISCは、このような欠点を克服するために開発された。

RISCという言葉は、カリフォルニア大学バークレイ校 (UCB, University of California Berkeley) で最初に使われた。そして、同校のDavid A. Patterson教授はプログラムの実行時間を次の式で表し、cpuの性能を決定づけている^[5]。

すなわち、

$$\text{プログラムの実行時間} = I \times C \times T$$

ここで、

I = プログラム実行に必要な命令数

C = 1命令に必要なクロック・サイクルの平均値

* OSI: Open Systems Interconnection, 開放型システム間相互接続
 ** GKS: Graphical Kernel System

$T = \text{基本クロック} \cdot \text{サイクル時間}$

RISCは、このCとTを小さくする方向であり、CISCはIを小さくすることによってプログラムの実行時間、すなわちcpu性能を向上させる。しかしCISCでは、命令の複雑化によってIを小さくすると、CとTが大きくなり、C、T、Iをどう決めるかが設計上重要なポイントとなる。RISCが設計しやすいことは、設計工数がCISCの数分の1ですむということからもうかがえる。たとえば、モトローラ68000では100人/月が、パークレイ校のRISC Iでは15人/月であったことから推測できる。

2) RISCの定義……RISCの厳密な定義はないがColwellらはその条件として次の六つをあげている^[6]。

① 命令とアドレッシング・モードの種類が少ない

基本命令数はRISC Iでは31種、これに対しモトローラ68000では61種、VAX-11/780は248種と多い。

② 固定命令形式

命令語の長さは一定で基本的に32ビットである。

③ 1マシン・サイクルごとの命令実行

1命令を1マシン・サイクルで実行することを基本とする。

以上の性格はパイプライン制御に適しており、より高速処理を容易にする。

④ ロード/ストア方式

メモリへのアクセスはロードとストアの2種類、演算はレジスタ間のみに限っている。CISCでは1命令でメモリからデータをロードし、演算を行って、結

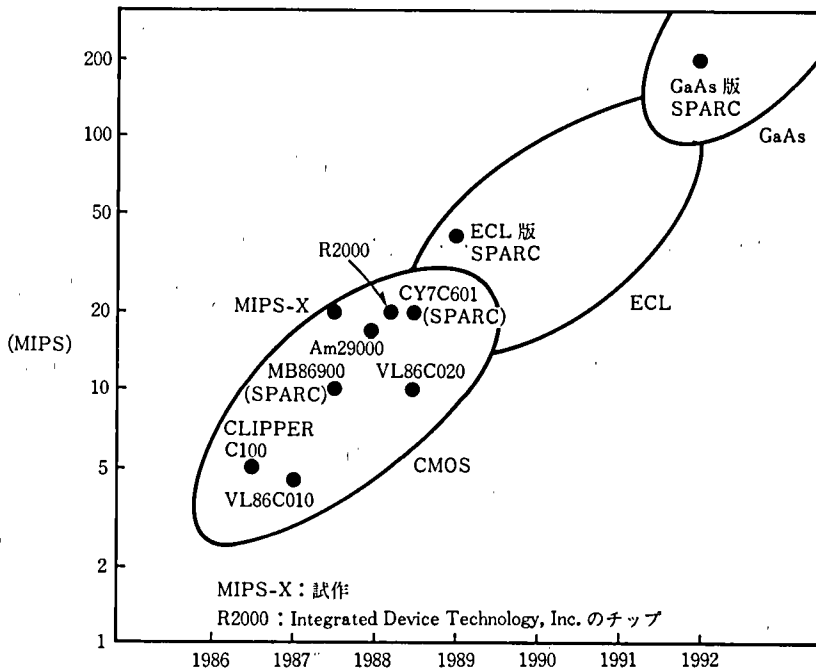


図 6 RISC チップの発展^[7]

Fig. 6 Progress of RISC chips

表5 主な RISC チップの仕様
Table 5 Specification of ^[7] typical RISC chips

型名	MB 86900 ^①	CY7C601 ^①	R 2000	Am 29000	CLIPPER C 100 ^②
メーカー名	Fujitsu Microelectronics, Inc.	Cypress Semiconductor, Corp.	MIPS Computer Systems, Inc.	Advanced Micro Devices Inc.	Intergraph Corp.
トランジスタ数(個)	5万	7万	10万	20万8000	13万2000
チップ寸法	不明	不明	8.5×10 mm ²	113 mm ²	不明
製造プロセス	1.5 μm CMOS ^③	0.8 μm CMOS	2 μm CMOS	1.2 μm CMOS	2 μm CMOS ^④
命令数	89種類	89種類	不明	115種類	168種類
汎用レジスタ (32ビット)	128個	136個	32個	192個	32個
MMU	×	×	○	○	○ ^⑤
FPU	×	×	×	×	○ ^②
キャッシュ容量					
命令	×	×	×	×	4 Kバイト ^⑤
データ	×	×	×	×	4 Kバイト ^⑤

①: SPARC アーキテクチャ採用

③: ゲートアレイ

②: CPU/FPU チップ, 2 個の MMU/キャッシュ・チップ
から成るが, 本表に示すのは CPU/FPU チップ

④: 1.5 μm ルール CMOS の C 300 も開発した

⑤: MMU/キャッシュ・チップ

("RISC 型 32 ビット・マイクロプロセッサ, CISC 型との棲み分け進む",
日経エレクトロニクス, 1988. 2. 8, pp. 145~150より)

果をストアする動作ができる。

⑤ ワイヤド・ロジック

機械語命令をワイヤド・ロジックで実行している。現在 CISC ではマイクロプログラムで解釈実行しており、フェッチやデコードなどのオーバーヘッドがある。ワイヤド・ロジックではこれがなく命令実行時間が速くなる。

⑥ 最適化コンパイラの活用

命令数が少なく、単純であることから最適化がしやすく、最適化によって効率を高められる。

以上のすべてを満たす必要はないが、これらの条件をほぼ満たしていることが RISC の条件であるとされる。

- 3) RISC 型マイクロプロセッサ……現状では 32 ビット・マイクロプロセッサとして、モトローラ 68020/68030、インテル 80386 などの CISC 型が多く使用されている。しかし、最近では RISC 型がこれらの CISC 型よりも高い処理能力を実現する可能性があるとして注目されている。たとえば図 6 に示されるように、1988 年中にも ECL 技術を使った RISC 型マイクロプロセッサが登場しそうである。CISC の MIPS 値と一概に比較できないが、50 MIPS を実現するといわれている。さらに 1992 年には、GaAs を使ったチップで 200 MIPS にも達すると予想されている。

代表的な RISC チップの仕様を表 5 に示す。

4.4 UNIX の標準化

現在、モトローラ 68000 系および RISC 型マイクロプロセッサを採用しているワークステーションの OS は、ほとんどが UNIX である。インテル 80386 系のワークステーションでは、表 6 のように MS-DOS 系と UNIX 系が提供されており、共存化が図られようとしている。

UNIX は 1981 年に AT&T がシステム III をリリースして以来、200 機種以上に採用されているが、ベンダーが自由に手を加えてきたため互換性が薄れてきている。そこで米国、欧州、日本で標準化の動きが活発化している。

- 1) POSIX……UNIX の標準化の最初の動きは、米国の UNIX ユーザ・グループが 1981 年に「アプリケーション・プログラムの移植性を保証するための標準化委員会」を結成したことに始まる。このユーザ・グループが 1984 年に提案した標準が IEEE の P 1003 ワーキング・グループにおいて標準化の基礎とされた。これは POSIX とよばれ、同一アプリケーション・プログラムが POSIX 準拠の UNIX で動くことを保証する規格であった。POSIX は AT&T の SVID (System V Interface Definition) にも大きな影響を及ぼしている。
- 2) X/OPEN……これは、欧州のメーカを中心とした標準化団体で、参加メンバーは次の通りである。

AT & T (米国)	ニクスドルフ (西独)
ブル (フランス)	オリベッティ (イタリア)
DEC (米国)	フィリップス (オランダ)
エリクソン (スウェーデン)	シーメンス (西独)
ヒューレット・パッカード (米国)	サンマイクロシステムズ (米国)
ICL (英国)	ユニシス (米国)
NCR (米国)	

日本からの参加の要請もあるが 1988 年 2 月時点では認められていない。非 IBM 路線に行くメーカを中心に UNIX をベースとしたアプリケーション開発環境の共通化を図ることが目的である。標準仕様として X/OPEN・CAE (Common Application Environment) を定め、その概要^[a]は次の通りである。

OS 基本部	: UNIX System V (AT & T の SVID 準拠)
OS 拡張部 (各国語対応)	: 8/16 ビット処理 (ヒューレット・パッカードの NLS インタフェース準拠)
プログラミング 言語	: C (System V ベース, ANSI 準拠), COBOL (ANSI 準拠, Level II COBOL 機能拡張), FORTRAN 77, ISO-Pascal
データ管理	: C-ISAM インタフェース, SQL (ANSI 準拠)
ウィンドウ管理	: 未定 (X Window またはサンマイクロシステムズの News)
ネットワーク	: 未定 (OSI 準拠のアプリケーション間通信手順およびトランスポート層プロトコル)

(“NC レポート”, 日経コンピュータ, 1988, 1, 18, pp. 46~47 より)

1988 年には各メーカから X/OPEN・CAE の仕様に合った UNIX が商品化され始めるだろう。X/OPEN・CAE はアプリケーション・インタフェースの標準であり、UNIX のカーネル部分にはふれていない。X/OPEN・CAE と POSIX, SVID の関係および X/OPEN に対する国際的支援体制は図 7, 図 8 の通りである。

- 3) Σ プロジェクト……日本では Σ プロジェクト (情報処理振興事業協会内) が共通のアプリケーション開発環境を整備するという目的でスタートし、OS に

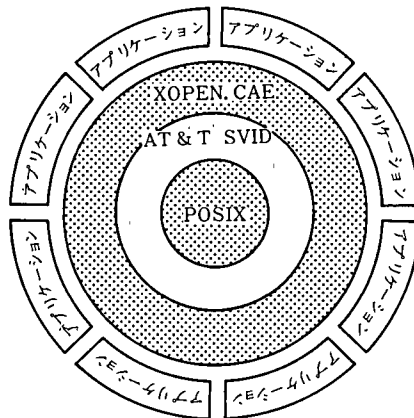
UNIX をベースとした「ΣOS」の仕様を開発している。ところが、ΣOS は X/OPEN の CAE と仕様が一致していないため整合性に問題があり、今後の仕様のすり合わせはむずかしいとみられている。しかし X/OPEN が欧米で事実上の標準になれば、国産メーカーは無視できなくなることから、NTT (日本電信電話) と国産メーカー大手 6 社が日本電子工業振興協会内に「UNIX 連絡協議会」を発足させ対応を検討し始めた。

以上のように標準化の動きは活発であるが、それらの中でも次の二つが大きなインパクトを与える可能性がある。一つは、パソコン用 UNIX として、米マイクロソフトと米 AT&T がインテル 80386 の統一 UNIX を共同開発するという。もう一つは、EWS の主力メーカーの米サンマイクロシステムズと AT&T が UNIX マシンの標準化に向けて提携したことである。

しかし、この標準化が SUN の SPARC というマイクロプロセサを前提にしていることから、特定の企業に拘束されない標準を目指す X/OPEN のメンバーの反発をうけており、AT&T は何らかの対応をせまられていた。これに対し、1988 年 2 月にドラ

表 6 インテル80386用 OS
Table 6 Operating systems for i80386

UNIX 系	VP/IX	インタラクティブ・システムズ フェニックス・テクノロジー
	MERGE 386	ルーカス・コンピュータ・システムズ
	VM 386	ソフトガード・システムズ
	XENIX 386	マイクロ・ソフト
MS-DOS 系	OS/2	マイクロ・ソフト, IBM
	MS-OS/2	マイクロ・ソフト

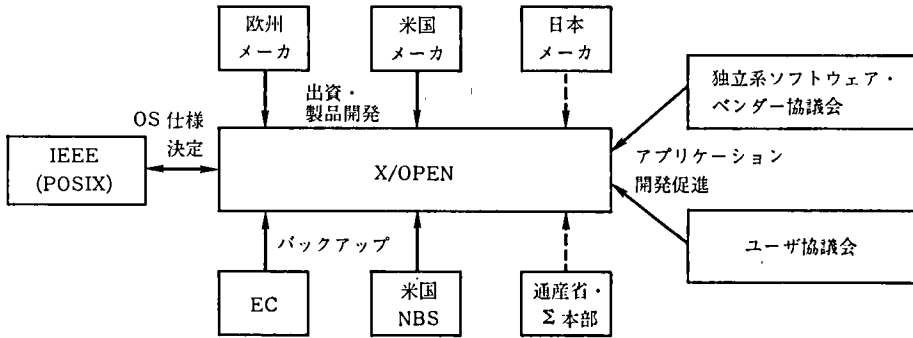


POSIX : Portable Operating System environment-interactive executive
SVID : System V Interface Definition
CAE : Common Applications Environment

(“NC レポート”, 日経コンピュータ, 1988, 1, 18, pp. 46~47 より)

図 7 アプリケーション・プログラムと X/OPEN の CAE, SVID, POSIX の関係^[8]

Fig. 7 Relationship among application programs, CAE, SVID and POSIX



IEEE：米国電気電子技術者協会（POSIXはIEEEが使用を決める標準UNIX）
 EC：欧州共同体 NBS：米国商務省標準局
 Σ本部：情報処理振興事業協会シグマシステム開発本部

“NCレポート、日経コンピュータ、1988.1.18, pp.46～47より”

図 8 X/OPEN に対する国際的な支援体制⁶⁾

Fig. 8 International organization for X/OPEN

スで開催された Uniforum で、AT&T は 1989 年の末に UNIX System V Release 4.0 の詳細を公開し、これにパケレイ 4.2 および 4.3 に加え POSIX, SVID, 現在の X/OPEN の仕様を反映するといっている。また、AT&T は日本語のサポート JAE (Japanese Application Environment) を新しくリリースすることも発表した。

また、UNIX の新しいバージョンが SUN の SPARC やインテル 80386 用にのみ開発されるのではないかと心配されていたが、他のマイクロプロセッサに対するインタフェースも用意するとしてベンダーを安心させた^[12]。

4.5 ワークステーションを特徴づけるソフトウェアの標準化

ワークステーションの中でとくに EWS において、特徴的なソフトウェア・サブシステムが三つある。

- 1) ネットワーク……X/OPEN の CAE では OSI に準拠することを目指しているが、現在最も普及している LAN は TCP/IP を使った IEEE 802.3 (Ethernet) に基づいている。LAN における分散ファイル・システムとして米サンマイクロシステムズが開発した NFS (Network File System) と、米 AT&T が開発した RFS (Remote File System) の二つが代表的であり、両者の一本化は困難と見られていた。しかし、サンマイクロシステムズと AT&T の提携によって一本化が図られるようである。

NFS は図 9 に示すように LAN 下にあるファイルを自分のファイルと同様に自由にアクセスでき、異機種間接続を指向するシステムにとって強力なツールとなっている。UNIX をベースとした LAN のシステムとして、NFS は実質的な業界標準になっているといつてよい。

- 2) ウィンドウ・システム…ワークステーションの重要な要素であるマルチウィンドウ・システムも、多くの種類が出現したが、現在では MIT (マサチューセッツ工科大学) で学内の計算機環境整備計画 (アテナ計画) の一環として開発された “X-WINDOW” が実質的な業界標準となりつつある。ただ日本では漢字

化の問題が残されている。X-WINDOW はパブリック・ドメインで安価に配布されており、移植性に優れていることから、多くのメーカーで採用されてきている。

X-WINDOW のイメージは図 10 のようであるが、サーバ・クライアント型とよばれ、ネットワークを介しても有効で、他のシステムのユーザ（クライアント）

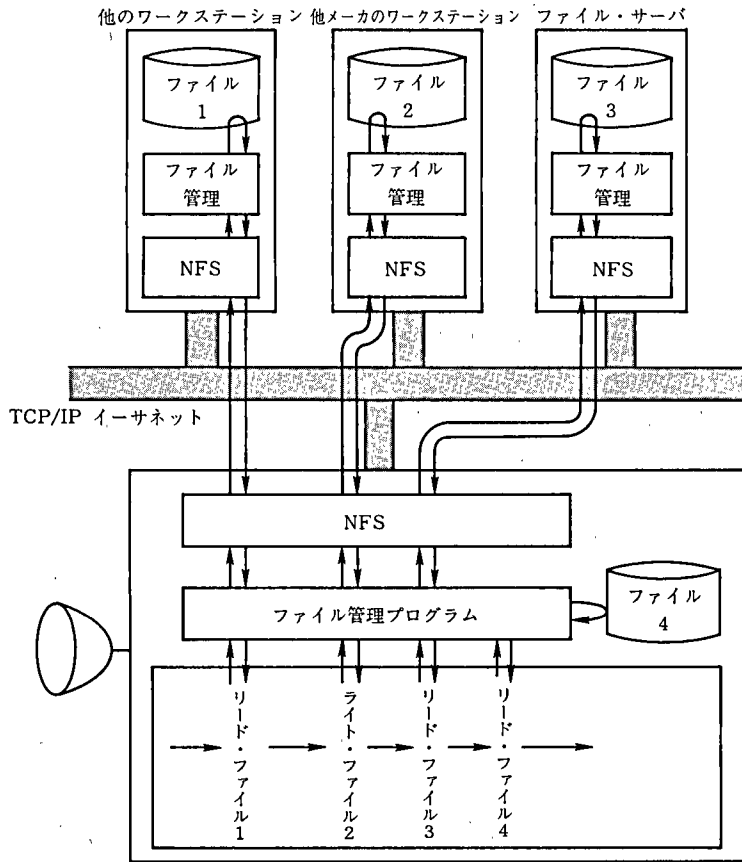


図 9 NFS の概念

Fig.9 Concept of network file system

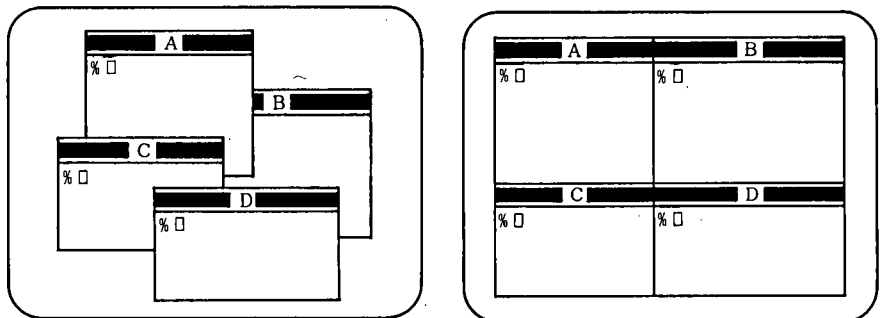


図 10 X-WINDOW のイメージ

Fig.10 Image of X-WINDOW

の動きをモニタすることができる。

この他に、SUN の NeWS や IBM 機上で稼働するカーネギーメロン大学の Andrew などもあるが、EWS の大部分のメーカが X-WINDOW を採用することを宣言している。

- 3) グラフィックス・ライブラリ……EWS がグラフィックスに使われることが多くなっている。このため多くの EWS ではグラフィック・ライブラリが提供されている。グラフィック・ライブラリの標準化はすでに ISO で決められており、アプリケーション・プログラマ・インタフェースとして GKS や PHIGS*があり、仮想デバイス・インタフェースとして CGI/CGM**がある。EWS メーカからこれらのソフトウェアが提供されつつあるが、効率を落とさないための工夫や新しいグラフィック機能のサポートなど課題も多い。

5. 今後の展望

マイクロプロセッサの急激な進歩が続き、それを cpu として搭載した、各種ワークステーションが続々と出現するだろうことは前に述べた通りである。しかも、ワークステーションが従来のコンピュータ・システムに、極めて大きなインパクトを与えることは疑いないことであろう。本章では、ワークステーションに関し、今後の展望と課題について私見を述べてみたい。

- 1) OWS と EWS の二つの進化の方向の相異がますます顕著になる……OWS と EWS が統一化されるだろうとする見解もあるが、むしろ両者はますます分離してゆくのではないかと考えている。それはマイクロプロセッサの適不適や OS の向き不向きによるのではなく、目標となる性能と価格によるところが大と考えるからである。

OWS は、代表的利用例である MML を介しての垂直型分散ネットワークに見られるように、一度に数百から数千台と大量に利用することから、第一に価格が非常に安いことが必要であり、性能はデータ処理が中心なため、それほど高い MIPS 値を必要としない。むしろ第四世代言語のような高級言語や MML のための通信機能が不可欠である。

これに対し、EWS は、科学技術計算やグラフィックス処理が中心になることから、極めて高い性能が要求される。とくに高速な浮動小数点演算機構 (FPU) や、グラフィック・エンジンといった特殊なハードウェアが必要である。またアプリケーションが固定的でなく、常に増殖をつづけることから、メモリや入出力装置に高い拡張性が必要となる。

そして通信機能においては、メインフレームとの垂直型はもとより、LAN 等による異機種間の水平型ネットワークを構築する機能が重要で、しかも高速通信が前提となる。

このように、OWS と EWS とを同一アーキテクチャのもとで作ることには無

* PHIGS: Programmer's Hierarchical Interactive Graphics Standard
 ** CGI/CGM: Computer Graphics Interface/Computer Graphics Metafile

理があるように思えてならない。

- 2) メインフレームとワークステーションの棲み分けが進む……ワークステーションが一人前のコンピュータであるからといって、メインフレームが不要になることはない。あるワークステーション・メーカの戦略では、ワークステーションと計算サーバやファイルサーバ等を LAN で結合することによってメインフレームは不要になるとしているが、筆者はこの考え方に懐疑的である。

確かに、ワークステーションの能力が向上し、LAN によってメインフレームの持つ機能のほとんどを代ることができるかもしれない。しかし、ワークステーション系はあくまでも、マンマシン・ユーザ・インタフェースとして、インタラクティブ（会話的）な処理形態や非定型的な情報処理をするのに適しており、大量なデータ処理や膨大な情報を一元的に蓄積管理（データベース）するのには適していない。

したがって、ワークステーションがローカルな仕事のためにのみ利用されているときは、単独あるいはワークステーション間の LAN を張ることによって十分であり、強力である。しかし、取り扱う情報の一元管理が必要であるときは、メインフレームとの共存が不可欠となる。これはちょうど組織に似ている。組織論では、部門における自主性を重んじる範囲は、各ローカルの運用にまかせるものの、全組織共通の情報や重要な情報は一元的に扱われているのが通常である。そして、コンピュータで処理する情報も同じ考えで扱われるのが無理がないと考えるからである。

すなわち、ワークステーション系にはそれだけでも十分仕事ができる自立性を持たせるが、全情報を一元的に管理する機構としてメインフレームが重要な役割をもつ。

- 3) 標準化は、業界標準から始まる……ワークステーションにおいても標準化の問題は難問中の難問である。UNIX 一つにしても、AT&T と X/OPEN, IEEE さらに日本での ΣOS, 加えてパークレイ版があり、多重に利害が輻湊し合い前途多難を感じさせる。そうかと言って、中立的な立場の ISO などの公的機関がイニシアチブをとって標準化を推進しても後追い標準となってしまう、時代の進歩に遅れかねず実質的な標準となり得ないことが多い。

UNIX に限っていえば、AT&T は同社とサンマイクロ・システムズやマイクロソフトとの提携と UNIX 標準化を切り離して考えるべきであり、UNIX の標準化の所期の目的をおし進めるため、すべてをオープンにし業界標準を根付かせることが社会的責任である。そして UNIX の権利をもつ企業は、しかるべき代価を支払い、標準を厳格に守ることによって“UNIX”を称し、ユーザのソフトウェア財産を保全することが第一義と考えるからである。

また標準化においても、日本は日本独自に考える時代でなく、広く国際的に提言するとともに参画すべきと考える。短期的な利害によるのではなく、長いレンジでの影響力の向上をめざし、国際的標準化活動の場にもっと力を注ぐ必要があり、企業もこれを積極的に支援することが肝要である。

6. おわりに

本稿の記述がEWSに偏った感が強いが、マイクロプロセッサやワークステーションの進化がEWSを筆頭に進んでいることから、そうならざるを得なかった。EWSの技術がOWSへ反映してゆくことは疑いないだろうが、課題はOSを含めたアプリケーション・プログラム・インタフェースの互換性の保持である。

ユニークなコンピュータも魅力的であるが、メーカーはユーザのアプリケーション・プログラムの互換性の保持を確保するとともに、機種も規模に合わせて選択できるように小から大までシリーズ化して提供することが必要である。

また、マイクロプロセッサの進歩が、即ワークステーションの進歩でなく、ワークステーションの進歩が即コンピュータ・システムの進歩ではない。それぞれに多くの要素が組み合わさって成り立っており、総合的にコンピュータ・システムを進歩させるには、ソフトウェアを含めたシステム・インテグレーションの技術が重要となる。さらに、従前と同様メーカーのユーザに対する保証とサービスが一体となって、システムが確立できるものであることには変わりない。

今後とも、マンマシン・インタフェースの優れた高度なワークステーションが出現し、多くのコンピュータ・システムに改善の機会を与えてくれるものと筆者は期待している。

- 参考文献
- [1] 「日経データプロ・ワークステーション」, 日経マグロウヒル社, WSI-100-051~056.
 - [2] 「32ビット・マイクロプロセッサの全容」, 日経マグロウヒル社, 昭和61年12月.
 - [3] 「32ビット・パーソナルコンピュータの世界」, 日経バイトブックス, 昭和62年11月.
 - [4] 「火がついた小型コンピュータ高速化競争」, 日経エレクトロニクス, 1987, 10, 19, pp. 163~178.
 - [5] 「論争を巻き起こしつつ商品化が始まったRISCアーキテクチャ」, 日経エレクトロニクス, 1986, 8, 25, pp. 252~258.
 - [6] Colwell 他, "Computeus, Complexity, and Controversy.", IEEE Computer 1985, 9.
 - [7] 「RISC型32ビット・マイクロプロセッサ, CISC型との棲み分け進む」, 日経エレクトロニクス, 1988, 2, 8, pp. 145~150.
 - [8] 「NCレポート」, 日経コンピュータ, 1988, 1, 18, pp. 46~47.
 - [10] 「エンジニアリング・ワークステーション大量導入時代の幕開け」, 日経コンピュータ, 1987, 11, 9, pp. 46~61.
 - [11] 安田寿明, 「マイコンピュータ入門」, 講談社, 1977, 3.
 - [12] 「Unix Today」, Computer Systems News, Feb. 10, 1988, p. 53.

執筆者紹介 野本 雄一 (Yuichi Nomoto)

昭和41年、九州大学理学部数学科卒業。同年日本ユニシス(株)(当時、日本レミントン・ユニバック(株))に入社。SEとして、製鉄、造船、電力のシステムの開発やサービスに従事。58年、日本ユニシス情報システム(株)(当時、日本ユニバック情報システム(株))に出向し、コンピュータ・グラフィックス事業にたずさわって、グラフィック・ディスプレイ、エンジニアリング・ワークステーション、CGシステム、CADシステムなどの企画・開発・販売に従事し、今日に到る。

情報処理学会、SIGGRAPH会員、日本コンピュータ・グラフィックス協会の運営および企画委員。著書に「プレゼン・グラフのすべて」(日経マグロウヒル社, 1986)。「コンピュータ・グラフィックスの基礎」(CQ出版, 1984)などがある。



情報処理システムにおける マイクロ・プロダクトの位置づけに関する考察

A Study for the Role of Micro Products on Information Processing Systems

佐藤 博

要約 情報処理システムを設計する際の重要な検討事項の一つとして、ワークステーション、パーソナル・コンピュータなどのマイクロ・プロダクトへの機能と負荷の分散をどの程度に図るかが挙げられるが、近年その重要度はますます高まってきている。

その理由を、マイクロ・プロダクト技術のもつ経済的側面と応用的側面から説明する。すなわち、経済的側面では、マイクロ・プロダクトの量産性のメリットとして汎用コンピュータに対する MIPS 単価の優位性の向上を挙げる。

また、応用的側面として、マイクロ・プロダクトの集積・小型性のメリットを活かした「エンド・ユーザの現場での情報処理」を可能にしたソフトウェアの進化を挙げる。これによって、汎用コンピュータの提供する「通信回線を介した情報処理」では実現しえなかったサービスを可能にしている。

しかし、マイクロ・プロダクトと汎用コンピュータは共に MIPS 単価が低下してきており、システム・コストに占めるソフトウェア開発・保守およびシステム運用などのコスト比率の方がハードウェア・コストの比率より大きくなった。すなわち情報処理システムの設計において、マイクロ・プロダクトへの分散化のレベルを決定するポイントは、ソフトウェア開発・保守・システム運用コストとなった。

このような背景から、分散データベース・システム、ネットワーク・マネジメント・システムなどのマイクロ・プロダクトと汎用コンピュータとの統合化機構が重要になってきている。

Abstract In designing the business information processing systems, the key issue arising is how much the functions and the load on the systems should be distributed into micro products, such as intelligent terminals, workstations and personal computers. Today, it is becoming more important to discuss this issue.

In reasoning the above, this paper describes two aspects of technical trends on micro products. One is economical aspect, and the other is application aspect.

The major factor on the former aspect is ascribed to the improvement of the MIPS/cost ratio over the mainframes. The major factor on the latter aspect is ascribed to the evolution of the software technology assisted by the integrity and miniaturization on micro processor units, which provides the end-user with the sophisticated user-friendly operational interfaces using fully 'the on-site processing' capability rather than 'the remote services through the communication line connected to the mainframe'.

However, due to large improvements of the MIPS/cost ratio on both microproducts and mainframes, the cost for system operation and software related one, that is development and maintenance, has been

larger than hardware related cost.

In this background, technical efforts for micro-mainframe integration, such as distributed database system, network management system and others are becoming more important.

1. はじめに

マイクロ・プロダクトとは、一般にマイクロプロセッサを中央演算装置 (CPU) あるいは制御装置として用いている情報機器を言う。本稿では、そのうちユーザ・プログラミング機能を有する事務処理用機器、すなわちインテリジェント端末、ワークステーション (WS) およびパーソナル・コンピュータ (PC) について、それらが汎用コンピュータを中心とする情報処理システムにどのように取り込まれ、どのような役割を担っているか、またこの2～3年の近未来において担うべきであるかを論ずる*。

マイクロプロセッサの処理能力は図1に示すように2年ごとに約2倍の増加を示しており、少なくとも今後10年はこの傾向が続くものと予想されている^{[1][2]}。これは、いわゆる“ムーアの法則”つまり「1チップ当たりの部品数が、毎年2倍となる」ことに基づいている^[1]。また、機能面についても、16ビットと32ビットのCPUの発展に伴いメモリ空間の拡大や、マルチタスク切り換え機能の内蔵など充実してきている。

マイクロ・プロダクトの代表であるPCの、日本における出荷量の伸びが純化している中で、16ビットや32ビットなどの高級機種は年率15%以上の伸びが見込まれている^[3]。

また、1987年の11月30日のビジネス・ウィーク誌によると、米国においては1988年末までに設置が予想される32ビットPCのMIPS値の総和は汎用コンピュータの約100万倍になると報告されている^[4]。

以上のようなマイクロ・プロダクトの高機能化と、その設置数の伸びが行きつく先

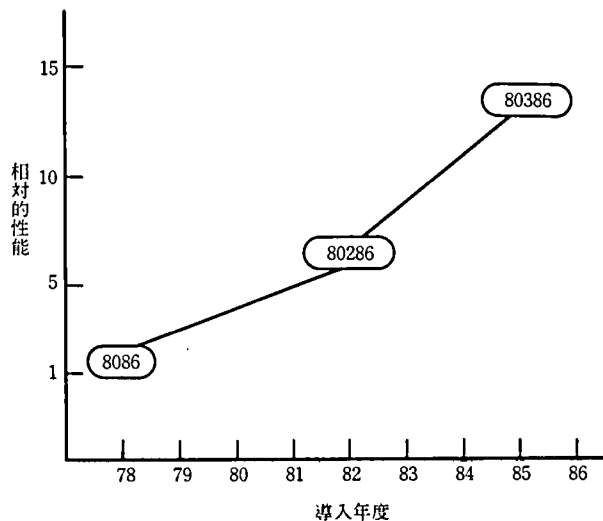


図1 マイクロ・プロセッサ処理能力の伸び
(インテル・コーポレーション社資料より)

Fig. 1 The growth of relative performance of microprocessor

*WSとPCの商品特性による区分は文献^[14]および^[15]を参照。

は汎用コンピュータの追い出しに通じるのであろうか。ほとんどの専門家はそうは見えていない。むしろ、マイクロ・プロダクトの能力と設置数の伸びにつれて、それらを有機的に結合するために、汎用コンピュータはより高性能なものが要求される、としている^{[4][5]}。

すなわち、情報処理システムにおいてマイクロ・プロダクトと汎用コンピュータは、競合関係にあると同時に相互補間の関係にあるとみてよい。ここでは、マイクロ・プロダクトが果たすべき役割を次の両側面から考察する。

- 1) 経済的側面……マイクロ・プロダクトでも可能な仕事は、マイクロでより安く実現したい。
- 2) 応用的側面……マイクロ・プロダクトが新しい利用面を拓く。その結果、システムの統合性、有機的結合性、サービス・レベルなどが高度化する。

上述の経済的側面と応用的側面は共に、マイクロプロセッサの特性である「高集積化による省スペース」と「量産による廉価」にその基盤を置いている(図2)。

すなわち、マイクロ・プロダクトは「省スペース」、「省部品」、「省電力」のゆえに応用的側面から採用されるとともに、「廉価」のゆえに今まで経済的な面から実現不可能であったシステムを可能にする。このように経済的側面と応用的側面は不可分であるが、本稿では、システムの既存の機能やサービスをマイクロ・プロダクトで代替することに焦点を合わせる場合、これをマイクロ化の経済的側面と呼び、新規の機能やサービスがマイクロ・プロダクトで実現可能になったことに焦点を合わせる場合、応用的側面と呼ぶ。

本稿では、この二つの概念に基づき、マイクロ・プロダクトの適用の発展について、とくに汎用機との関係を中心に私見を述べる。

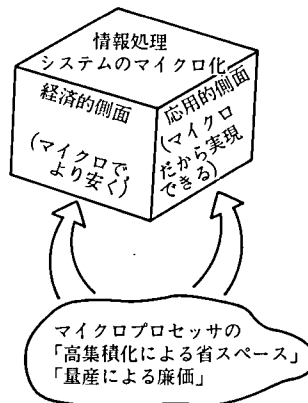


図2 情報処理システムのマイクロ化の二つの側面

Fig. 2 Two aspects of microprocessor usages in information systems

2. 経済的側面におけるマイクロ・プロダクトの特性

2.1 量産効果と規模の効果

マイクロ・プロダクトは、一般に次のハードウェアで構成される。

- ・マイクロプロセッサ
- ・メモリ
- ・外部メモリ (固定ディスク, ディスケットなど)
- ・周辺装置 (CRT などの画面表示装置, プリンタ, 通信制御ボードなど)
- ・バス・ライン, 電源

これらのコスト・パフォーマンスを汎用コンピュータとの比較で考察してみる。比較に際して量産効果と規模の効果という経済概念を、ここで次のように援用する。すなわち、量産効果：同一製品を多量に生産することによる経済効果

規模の効果：類似の仕事を一括して処理することによる経済効果

とする。一般には量産効果は規模の効果に含まれる概念であるが、ここでは規模の効果を「コンピュータ内での処理」の規模、すなわちパーソナルか部門一括かあるいは全社一括かなど、一組みのコンピュータ・システム内での処理の集中の度合いに対応した概念とする。なお、ローリンが「ハードウェアのスケールの経済性」と「生産スケールの経済性」と呼んでいる概念^[6]は、それぞれ本稿の「規模の効果」と「量産効果」に相当している。

図3～図6は、マイクロ・プロダクトと汎用コンピュータのそれぞれの構成要素についての「規模の効果」を表している。

図3は次のことを示している。すなわち、汎用コンピュータにおいては2～3年前まではより性能の高い汎用コンピュータに集中して仕事をさせることが有利であったが、この1～2年では規模の効果はなくなっている。このことは、ローリンも指摘していることである^[6]。また、汎用コンピュータではマス・ファイルにおいても規模の効果が明瞭ではない(図5)。一方、マイクロ・プロダクトを見る(図4と図6)と、グラフ中に「ひざ」(最適規模値)があることを示している。この「ひざ」の存在は量産

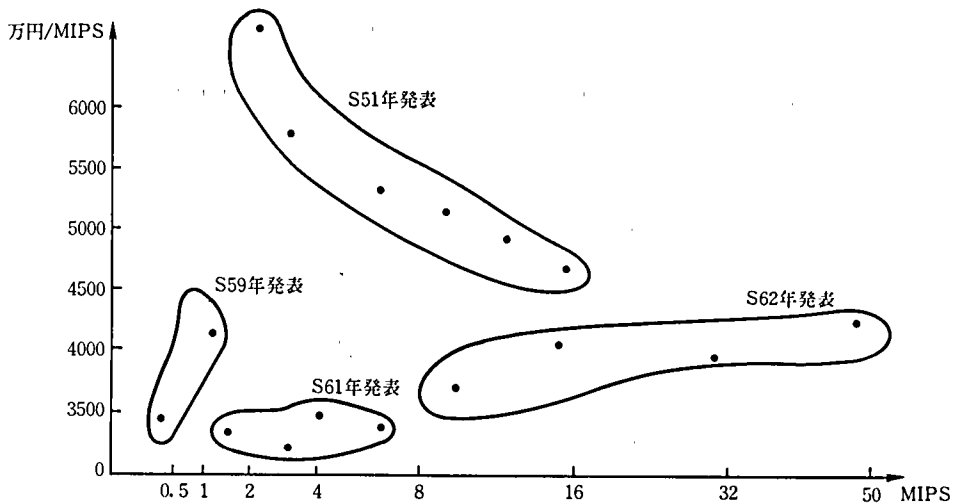


図3 汎用コンピュータにおけるコスト/MIPS 比率の例
 (出典：週刊 COMPUTER WORLD EXTRA 1/26/87 を筆者が図示化、
 この図は4種のシリーズ・マシンの各機種値をプロットしたもの)

Fig. 3 An example of cost/MIPS ratio on mainframes

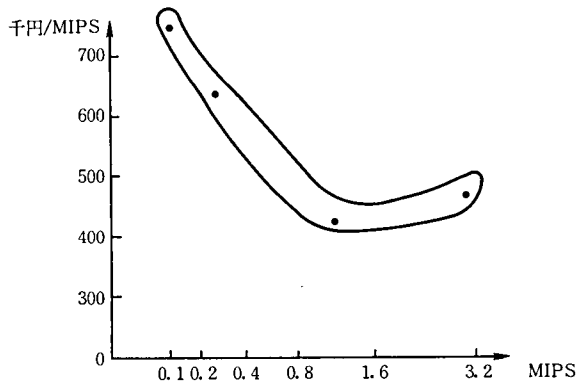


図4 パーソナル・コンピュータ(本体—マスファイル)におけるコスト/MIPS 比率の例
(出典：1987年発表商品を参考)

Fig. 4 An example of cost/MIPS ratio on PC

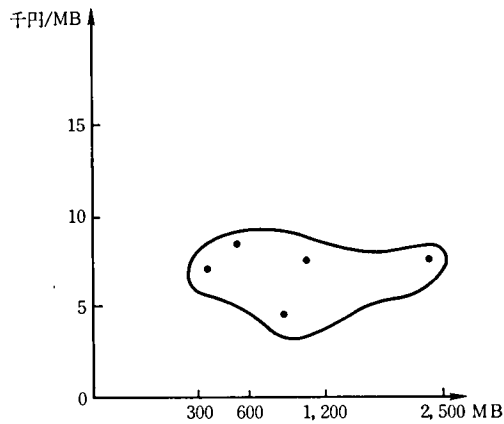


図5 汎用コンピュータにおけるマス・ファイルのコスト比率の例
(出典：1987年発表商品を参考)

Fig. 5 An example of massfile cost/megabyte ratio on mainframe

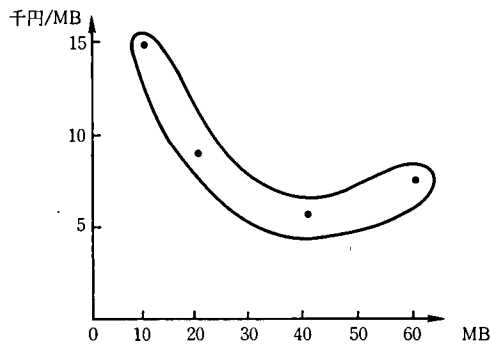


図6 マイクロ・プロダクトにおけるファイルコストの例
(出典：1987年11月実務価格を参考)

Fig. 6 An example of massfile cost/megabyte ratio on micros

効果で説明できる。マイクロ・プロダクトは多くの標準化された部品から成り、それらは非常に多量に、設計され製造される¹⁶⁾。すなわち、部品の複雑さ以上にそれが多量に使用されるかどうかにかかわらずコスト・パフォーマンスの良否がかかっている。したがって、「ひざ」の位置はプロダクトの社会的浸透につれて右へ移動する。

次にマイクロ・プロダクト（量産効果に強みを持つ）と汎用コンピュータ（規模の効果に強みを持つ）とを比べた場合、どちらがコスト・パフォーマンスが良いのであろうか。シュッセルは、「ワークステーションの処理コストはミニコンで同じ命令を実行した時のわずかに10分の1であり、ミニコンの処理コストはメインフレームの3分の1に過ぎない」ことを述べている¹⁹⁾。さて、それではここで実際のデータを見てみよう。CPUとマスファイルのコスト・パフォーマンスを比較したものが表1である。

この表はマイクロ・プロダクトの方がCPUについてはコスト・パフォーマンスが高いことを示している。これは、マイクロプロセッサ・チップの量産効果と見てよいだろう。また、マス・ファイルについては汎用コンピュータの方がメガ・バイト当たりの単価がややマイクロ・プロダクトより有利であることを示している。これは汎用コンピュータのもつ規模の効果がマイクロ・プロダクトの量産効果より（現状では）やや勝っていると解してよいだろう。

このほか、メモリについては、両方とも同じ要素が用いられているので、コスト・パフォーマンス（たとえば、1ビット当たりの単価）は同一と見なせる。

表1 マイクロ・プロダクトと汎用コンピュータのコスト・パフォーマンス

Table 1 A comparison of cost performance between micro-computer and mainframes

	CPU (千円/DHRYSTONE)	ファイル (千円/MB)
マイクロ・プロダクト	0.7	6
汎用コンピュータ	15.0	4

DHRYSTONE: UNIXマシンでの性能評価のための標準的ベンチマーク・テストの実行時間から計算される指標である。

なお、この表における算出方法は次のようにした。

- ① パフォーマンスは、米国のUNIX協会の公表しているDHRYSTONE VERSION 1.0の中から、16ビットPC (IBM-PC/AT) と汎用コンピュータの中で、価格/DHRYSTONE比が最良のものを選んだ。
- ② 両機の価格は公表されているもの(週刊COMPUTER WORLD EXTRA 1987.1.26など)を採用した。
- ③ DHRYSTONE値は次の点で汎用コンピュータにとっては不利である。
 - ・Cコンパイラに優れたものがない。
 - ・UNIXモードのエミュレーション負荷がある。

しかし、マイクロ・プロセッサのMIPS値を汎用のそれとを比較するよりは良いであろう。

表2 システム・コストの推移 (文献¹¹⁾図86より抜粋)

Table 2 A trend of the computer system cost

	1965	1975	1985
コンピュータ・ベンダへの支払い	10	14	15
通信費	0	4	11
人件費	7	21	28

(単位百万ドル)

2.2 システム・コスト

しかし、ここで注意しなければならないのは、ハードウェア・コストとシステム・コストの区分である。システム・コストはハードウェアの他にソフトウェア開発・保守、システム運用に要する人件費や通信回線利用などに関するコストで構成されるが、表2に見られるように人的なファクタに関するコストの比重は年々高まってきている。すなわち、MIPS/コストのみの比較だけでは不十分である。多数のマイクロ・プロダクトに処理系とデータを分散する方式をとると、保守コストと運用コストが増大し、これらのコストの適正化が重要になる。

この問題は分散型 DBMS^{[8][9][16]} やネットワーク管理システムにより解決に向かうと思われるが、その実用化は1990年代になるだろう。したがって、現在および近未来においては保守・運用に対する規模の効果、裏返しに言えば分散化によるシステム・コスト増を十分に考慮しなければならない。

とくにデータは、次の特性をもつことから汎用コンピュータ上に蓄積・保守するのが一般的である。

- 1) 情報は“量”の集積と“質”の向上によってその価値は飛躍的に上がる。
- 2) 情報は入手した時点から急速に価値を失う。

したがって、経済的側面からは、情報処理システムにおけるマイクロ・プロダクトの役割を、次のように設定することができる。

- 1) データ保管機能より処理機能を主とする。
- 2) マイクロ・プロダクトに保管するデータは、活性度の高いものより低いもの(定数的なもの)を主とする。

一方、汎用コンピュータは、より大量で複雑なデータの集中保管機能が要求される。

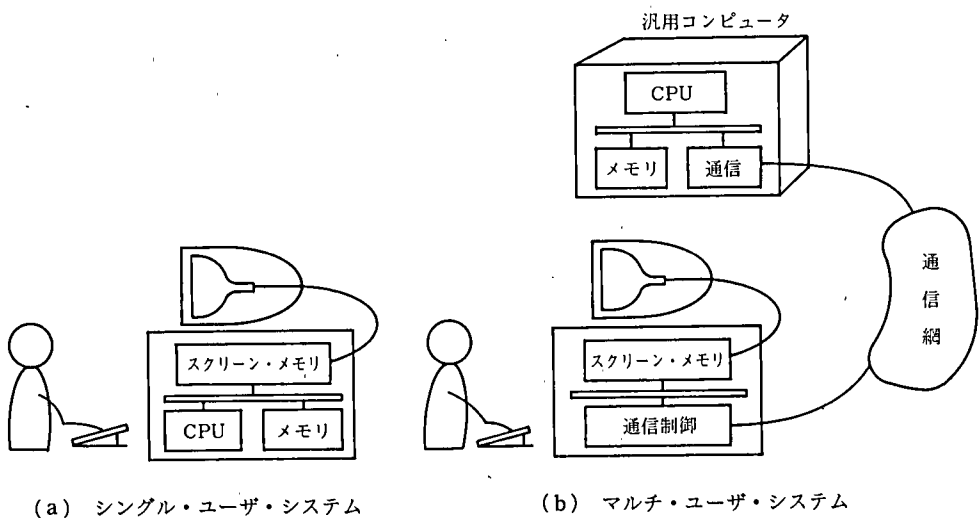


図7 シングル・ユーザ・システムとマルチ・ユーザ・システム

Fig. 7 Single user system vs multi user system

3. 応用的側面におけるマイクロ・プロダクトの特性

3.1 シングル・ユーザ・システムとマルチ・ユーザ・システム

マイクロ・プロダクトの小型・安価・低電力・低発熱等の特性から、その設置場所は、床上（フロア）、机上（デスクトップ）、さらにはひざ（ラップトップ……ただし欧米人のひざ？）になり、使用時には1人が1台を占有する、すなわちシングル・ユーザ・システムである。これに対し汎用機やミニコンなどの使用形態を、マルチ・ユーザ・システムと呼ぶ。これは1台のコンピュータ資源をTSSあるいは、複数のバッチ・ラン等で共有する形態である（図7）。

シングル・ユーザ・システムでは図7(a)のごとくCPU、メモリとCRT、鍵盤、マウスなどの装置が高速バス・ラインで結ばれている。それに対しマルチ・ユーザ・システムでは図7(b)のごとく、端末機側の表示用メモリ（スクリーン・メモリ）と、処理系側のCPUやメモリとは回線あるいはLANなどのネットワークで結ばれている。

さて、表3はUNISYS DS7などの高解像度のCRTの1画面分のデータをメモリからCRTに転送するのに要する時間を転送媒体別に比較したものである。このデータをもとに以下の議論が行われる。

3.2 マンマシン・インタフェース

最近では操作性が高いということから、マンマシン・インタフェースのオブジェクト指向化が盛んに行われている。このためには、次の機能の実現を必要とする。

「表示レポートの10行下を見たい場合に、「表示せよ、現位置から+10」と指示する代わりに、カーソルまたはマウスでレポートの表示域の下を押しつづけ画面を上方にスクロールさせ、カーソル・スキャン・キーやマウス・ボタンから指を離れた時にスクロールを停止させられるようにする。すなわち、いま見たいデータ（すなわちオブジェクト）を「10行下にあるはず」と手探りするのでなく、「ある範囲のデータを順に表示し、見たいデータが出てきた時にそこで停止」させられるようにする。」

表 3 転送媒体別表示時間比較

Table 3 Time comparison by data transfer media to display a data block

媒体種別	UNISYS DS7 1画面 (1920字) 分のデータを転送する時間 (秒)
中速通信回線 (9600 BPS)	3.2
高速通信回線 (48 KBPS)	0.6
低速LAN (1 MBPS)	0.04
高速LAN (20 MBPS)	0.002
バス・ライン (DS 7)	0.002

ここでは、モデム切替時間やLAN制御等のオーバーヘッドを50%見込んでいる。

これは一般に次のように実現される。すなわち、1画面分のデータをまずスクリーン・バッファに転送し（ハードウェアがCRTに表示し）、スキャン・キーの押下を確認してさらに、1行分ずらした画面データをスクリーン・バッファに転送することを繰り返す。この場合に操作者のイライラを招かないためには、1画面当たりの表示所要時間を0.05秒以内（すなわち20行分スクロール・アップするのに1秒以内）にする必要がある。すなわち、1920文字分の（コード）データを処理用のデータ・バッファからスクリーン・バッファへ0.05秒以内に移送する必要があるが、これを満たすのは表3からLANおよびバス・ライン方式でなければならないことがわかる。

また、マン・マシン・インタフェースがWYSIWYG^[10]、すなわち最終レポート形式（たとえば、ワープロ文書中の4倍角文字は画面でも4倍角として表示される）で表示するためには、画面表示はビット・マップである必要がある。しかし、ビット・マップ表示の文書の上下左右のスクロールについては、文字表示のそれに対し約50~300倍のデータ転送速度を要する^(註2)。このため、バス・ラインあるいはLANによる対応のみではむずかしくなり、ビット・マップ表示専用のCPUを必要とする。これを模型的に図8で表す。すなわち、バス・ライン経由の表示方式（図8(a)）ではなく、同図(b)のようなスクリーン・メモリ上のビット、あるいは文字ブロックを直接にCRT表示機構へ送り出す専用CPUを用いる方式を必要とする。

ここで、上述の「より良いマンマシン・インタフェース」と「シングル・ユーザ・システム」との関連を整理すると次のようになる。すなわち、

- 1) 文字データのみを扱うシステムであれば、高速LANを介せばマルチ・ユーザ・システムでも「より良いインタフェース」は可能である。しかし現在のところは高速LANのコストが高く、マルチ・ユーザ・システムでのオブジェクト指向の操作環境の提供は現実的でない（たとえば、汎用コンピュータでワープロ・ソフトを動作させるなど）。
- 2) ビット・マップ方式ではシングル・ユーザ・システム、および高速LANによる

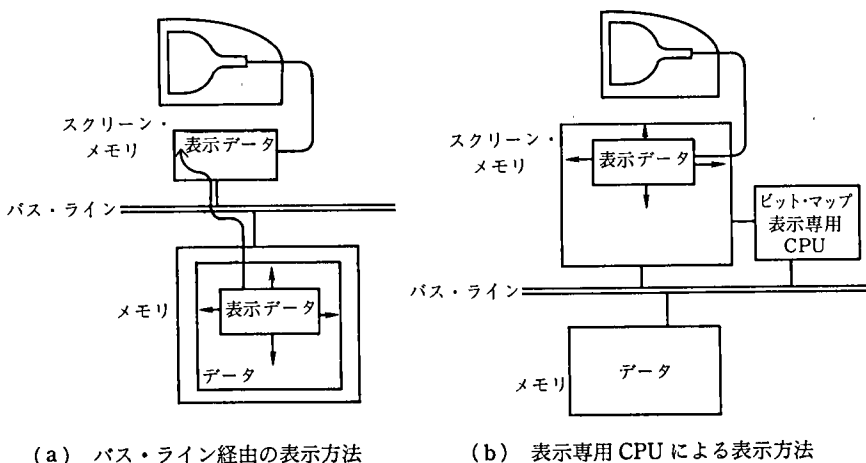


図8 バス・ライン経由の表示方法と表示専用CPUによる表示方法

Fig. 8 Bus-line display method and display CPU method

マルチ・ユーザ・システムのどちらにおいてもマイクロ・プロダクト側に専用 CPU を必要とする。この場合、処理系のデータ・バッファとビット・マップ表示機構側との間のデータ転送の対象は文字コードあるいは表示制御コードになるが、これを汎用プロトコルとして定めればマルチ・ユーザ・システムとシングル・ユーザ・システムを統合した操作環境の提供が実現できる*。

- 3) つまり、歴史的には PC や WS などのシングル・ユーザ・システムは、先進的なオペレータ・インタフェースを提供してきたが、将来的には LAN のコスト低下、ビット・マップ表示 CPU の普及につれて、汎用コンピュータをはじめとするマルチ・ユーザ・システムを統合したインタフェースもマイクロ・プロダクトは提供するようになる。

3.3 サービス機能要素

以上の議論を踏まえて、マイクロ・プロダクトの導入によるサービス機能の向上をまとめると次のようになる。

- 1) 高度なマン・マシン・インタフェースの提供……プル・ダウン・メニュー、マウスによる「オブジェクト指向」、あるいはスムーズ・スクロールなどが実現できる。また、マルチ・メディアの処理が可能になる。汎用コンピュータの処理系への高度な操作インタフェースをマイクロ・プロダクトで提供している例として U-VIEW/MAPPER がある。これは、汎用コンピュータ上の第 4 世代言語 MAPPER とマイクロ・プロダクトの間での機能分散の将来方向の一つを示している。この他に、次のようなサービス機能の向上が期待できる。
- 2) システム応答性の向上……シングル・ユーザ・システムの強味である「操作者による機器の占有」によって、常に安定した処理時間と操作者への応答時間を確保できる。これは、操作者の心理的負担を減ずるのに大きく寄与する。
- 3) システム多様化への対応……メリットとしては、①汎用コンピュータの運転時間に必ずしも「端末」の使用時間を合わせなくてもよい、②一台の「端末」から基幹業務システムだけでなく非定型のオフラインやオンライン業務を統一したメニューで呼び出せる、③操作者のレベルに合わせたコマンドやメニューなどを提供できる、などがありニーズの多様化に対応できる。(たとえば DS 7 の U-VIEW/MAPPER や DS 7 における複数のカナ漢字変換システムの選択など)。

4. おわりに

情報処理システムの設計に際しては、「データ」と「プロセス」の両面からアプローチする必要がある^[15]。基幹データベースは当分の間は、管理の規模の効果を得るために汎用コンピュータ上で集中管理されよう。そして、システムの拡大(「端末」の増加や処理の複合化・高度化など)によってますますデータ量とサービス量の増大がもたらされ、その結果、より大型な汎用コンピュータが必要となる。一方、「プロセス」の実行に関しては、2章・3章に見たようにコスト面やサービス機能面から、マイクロへの分散が進行するであろう。

*UNIX では、文献^[11]等で紹介されている BIT, X-window, あるいは CRT への Post Script の適用(文献^[12])などの試みが先行している。

しかしながら、マイクロ・プロダクトはその誕生の経緯から基本ソフトウェアの構造、ファイル構造、システム・コマンドなど汎用コンピュータのそれと大きく異なっている。これが「プロセス」の分散化を進める上での大きな障害として横たわっている。システム統合化アーキテクチャなどは、この障害を取り払って、システム設計者や開発者に汎用コンピュータと共通のプログラム・インタフェースおよび操作者インタフェースを与えようとする試みである。

これは、10年前のネットワーク・アーキテクチャの提唱とその後の展開とよく比較論議される。すなわち、各コンピュータ・メーカが、それぞれ独自のネットワーク・アーキテクチャを発表し「情報機器の囲い込み」競争に陥ることも一時は心配されたが、現在ではOSIで統一化されつつある。システム統合化アーキテクチャは、それを含みかつ、はるかに遠大で包括的なものである。そうであるがゆえにメーカの社会的責務として、より慎重で活発な論議を進めることが必要であると認識している。

また今後は、シーズとしての問題解決型システムの発展・進化と、ニーズとしての企業活動の分権化の要請が挨まって、情報システムの「分散化 (distributization)」と共に「分権化 (decentralization)」が急速に進展するものと予想される。このような視点から、マイクロ・プロダクト固有の商品特性だけでなく、導入・教育・技術相談といった商品にまつわるサービス特性がより重要になってきている。

なお、第2章のデータは、自社と他社の商品を区別せずに公表されているもののみを使用し、機種は原則としてその商品カテゴリにおいてマーケット・シェアが上位のものを選んだ。

執筆注 (注1) 文献^[4]では、「1987年に(米国内で)出荷されたIntel 80386を用いたコンピュータは約325,000台、さらに1988年には200万台の上乗せと予想されている。これらの計算処理能力の総和は20 TIPS (Trillion Instructions Per Seconds)になる。一方、米国内に設置されているIBMおよびIBM互換の汎用コンピュータの処理能力の総和は145 BIPS (Billion Instructions Per Seconds)と推計されている。(筆者訳)」と報告されている。

しかし、ここで注意しなければならないのは、本稿で述べているように汎用コンピュータのスケール・メリット(規模の効果)が無視されていることである。すなわち、これは日本中の自家用車のエンジンの最大トルクの総和と、バス・鉄道・航空機などの公共輸送車輛のエンジンの最大トルクの総和とを比較するに等しい。ここでは自家用車の稼働率(車庫に眠っていない率)、平均乗車効率(5人乗りに1.5人)を考慮していない。なかんずく、PCの32ビット化によって瞬間風速は上っても、システムのスループットがそれに比例して上るわけではない。操作者の思考・反応待ち、データ入出の待ちがCPU時間の圧倒的割合を占めているからである。

(注2) 1画面分の文字(コード)データの情報は、80欄×24行=1,920文字=15,360ビットである。

一方、ビット・データの情報は、1,120ドット(横方向)×750ドット(縦方向)=840,000ドットである。したがって、DS7クラスの高解像度CRTにおいては、ビット・データの方が文字データより約50倍の情報量になる。

また、スクロールに際して、ビット・マップでは文字行単位に不連続にスクロールするのではなく、巻き紙を緩やかに巻きとるように連続的に(すなわち、目で追えるように)画面に移動させ一層の操作性を良くしようとする。このためには、画面データの表示を(1文字行28ドットでなく)4~10ドット分づつの、スクロールに分割して、スクリーン・バッファに転送する必要がある。

参考文献 [1] K. Wise 他, 有澤訳: マイコンコンピュータの将来—20年後の技術と社会, 近代科学社, 1981, p.14.

- [2] C. Wetzman, 成田ほか訳：マイコン・ミニコン分散型システムの設計, 産業図書, 1982.
- [3] コンピュータ周辺機器市場の展望と戦略, (株)矢野経済研究所, S 62, 11, 30. p.10.
- [4] "Computers : The New Look", Business Week, Nov. 30, 1987, pp. 78~85.
- [5] J. W. Verity, "米国メインフレーム・ユーザ調査-コンピュータ・パワー増強のニーズは依然強い", 日経コンピュータ, 1986, 6. 23.
- [6] H. Lorin, 菅野監訳 石田訳, ユーザのための分散処理システムの最適設計法, ジャテック出版, S 57. 5. 20.
- [7] J. Breslin, Selecting And Installing Software Packages, Quorum Books, 1986, p. 57.
- [8] 増永, "分散型データベースシステム", 情報処理, Vol. 28, No. 4, 1987. 4.
- [9] G. Schussel, "第5世代開発環境による90年代のシステム開発", 日経コンピュータ, 1988, 2. 1.
- [10] 上林憲行, "ワークステーション ユーザインタフェースの構成技術", 情報処理 Vol. 25, No. 2, 1984, 2.
- [11] 下村他, "BIT と WIP と U-Windows", bit 臨時増刊号, UNIX 特集, 1987.
- [12] 石田他, "Post Script 事始め", 日経バイト, 1987年4月~6月.
- [13] 山田他, "データ中心システム設計技法", 日経コンピュータ, 1985, 5, 27.
- [14] 佐藤, "オフィス・ワークステーションの動向", データ通信, 1987, 6.
- [15] '88 オフィスワークステーション(OWS)総市場, (株)矢野経済研究所, pp. 22~34.
- [16] 滝沢, "異種分散型データベースシステム", 情報処理 Vol. 29, No. 1.

執筆者紹介 佐藤 博 (Hiroshi Sato)

昭和42年武蔵工業大学経営工学科卒業。同年日本ユニシス(株)(当時日本レミントン・ユニバック(株))入社。9000シリーズ, S 80 のスプリング・システム, アセンブラー・プロセッサ, 通信制御システム等の基本ソフトウェアの開発を経て, 昭和54年よりワークステーション・システムの開発に従事。現在, ワークステーション・システム部長, 技術士(情報処理部門), 情報処理学会々員。



MAPPER1100 におけるマンマシン・インタフェースの改良 ——直接操作の導入

The Improvement of Man-machine Interface in MAPPER1100 ——Adoption of Direct Manipulation

伊 東 春 正

要 約 マンマシン・インタフェースの動向は、マルチウインドウ、マウス、アイコンによる直接操作と、MML (Micro Mainframe Link)、システム統合技術などの統一化の方向である。

DS 7 のマンマシン・インタフェースとしては、端末エミュレータとしてホスト上のソフトウェアを使用する場合と、スタンドアロンのパソコンとして使用する場合の 2 通りがあるが、これらを統一するためのソフトウェアとして U-VIEW がある。

U-VIEW は、既存ソフトウェアのマンマシン・インタフェース部分を分離したもので、入力オペレーションをエミュレートすることにより、見かけ上の操作を統一する試みである。その最初の応用として MAPPER 1100 に適用し、従来はコマンド入力方式であったものを逐次メニュー方式に改良している。

Abstract The trend of man-machine interface improvement is towards direct manipulation (e.g. multiwindows, mouse and icons) and system integration (e.g. MML (Micro Mainframe Link) and System Integration Architecture). In DS 7 system there are two modes of man-machine interface, one is terminal emulation mode and the other is personal computer mode.

U-VIEW is a software which integrates these mode.

This paper describes my view of man-machine interface design, and also reports the examples of improvement of MAPPER operation by using U-VIEW.

1. はじめに

マンマシン・インタフェースは人間と機械の接点であり、コンピュータ・システムにおいては手による入力と、目で見える画面表示または印刷物として実現される。

優れたマンマシン・インタフェースの目標は、使いやすさを追求することであり、使いやすさとは次の項目を満たすことである^[1]。

- 1) 親しみやすさ (日常言語による表現, 対話がしやすいなど)
- 2) わかりやすさ (覚えやすい, 使い方が容易など)
- 3) 効率のよさ (早い, いらいらしないですむなど)
- 4) 寛容性 (誤った使い方, 不完全な使い方でも許されるなど)
- 5) 統一性 (一貫した使い方, 応用できる使い方など)
- 6) 適応性 (初心者から熟練者まで使えるなど)

最近、ソフトウェア開発部門でもマンマシン・インタフェースの向上が、とくに注目されている。ハードウェア技術の進歩によりアドレス空間が広がり、上記の使いやすさを実現できる環境になったからである。

本稿ではソフトウェア面、メッセージ表示やカーソル移動などからマンマシン・インタフェースの現状を報告し、マンマシン・インタフェースを設計するうえでの考慮すべき事柄を述べ、それに基づいて開発した U-VIEW システムを MAPPER 操作に応用した例を紹介する。

2. マンマシン・インタフェースの現状

2.1 世の中の現状

パソコンが多機能パソコンとなり、通信やビットマップ機能を備えたオフィス・ワークステーションへ発展していく過程で、ソフトウェアの操作環境も大きく変わってきた(図1)。

単機能ソフトウェアによる処理系ごとに異なった操作から、一つのソフトウェアで複数機能をサポートする統合ソフトウェア、あるいは同一コンセプトで開発されたシリーズ・ソフトウェアなどにより、処理が違っても同一操作で利用できる統合操作になってきている。さらに、ビットマップ・ディスプレイ上でマウスとアイコンとマウスにより視覚的な操作で処理を進めていく、オブジェクト指向ソフトウェアによる直接操作へと移りつつある。

一方、汎用機、オフィス・コンピュータ、およびワークステーションなど使用する環境が違っても、統一したマンマシン・インタフェースを提供する目的のシステム統合化アーキテクチャが発表され、ソフトウェアのインタフェースを統一する動きがある。

- 1) 直接操作……直接操作というのは、ユーザの行為と行為の結果をただちに直接にユーザにわからせることであり、次のように定義されている^[2]。
 - ・ 関心の対象を継続的に表示する。
 - ・ 複雑な構文を持つコマンドの代わりに物理的なアクション(マウス、ジョイスティック、タッチ・スクリーンなどを用いての選択)、またはラベル付きボタンの押下により操作する。
 - ・ 逐次的で可逆性があり、迅速な操作ができ、関心の対象に及ぼす影響がただちに表示に反映される。

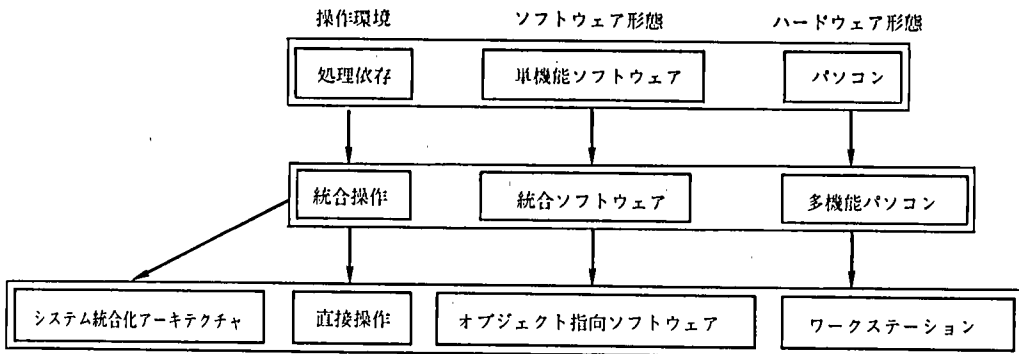


図1 操作環境の推移

Fig.1 Progress in operation environments

- ・ 最小限の知識で使用でき、かつより高度な使い方を徐々に習得できる。そして、そのようなシステムがもつ利点として次のものがあげられる^[2]。
- ・ 初心者には、より経験をつんだユーザが実際に操作するのを見て、短時間で基本的な操作を学習できる。
- ・ 熟練者は、さまざまな作業を短時間で実行できるだけでなく、新しい機能を定義できる。
- ・ システムをあまり使わないユーザでも操作方法を忘れないでいられる。
- ・ エラー・メッセージをほとんど必要としない。
- ・ ユーザは自分の目的を達成できたかどうかすぐわかる。もしうまくいかなかったら、指示を変えればよい。
- ・ システムがわかりやすく、操作のやり直しも簡単なので、ユーザはほとんど不安を感じない。
- ・ 操作はシステムにより制御されシステムの反応を予測できるので、ユーザは自信をもって操作できるだけでなく習熟しやすい。

直接操作が最も成功した例は、初心者でも簡単に覚えられ、熟練者をも長時間熱中させるだけの複雑さをもつテレビゲームであるといわれている。

オフィス・ワークステーションでこの直接操作を採用したのが、1981年ゼロックス社の Star が最初で、その後のアップル社の Lisa が続く。これらのワークステーションでは、オフィス環境をディスプレイ画面上に再現している。すなわち重なりあったウィンドウそれぞれが伝票やメモ用紙などの紙であり、アイコンで表された事務用品を選んで紙(ウィンドウ)の上のデータを加工できるようになっているのでデスクトップ環境という。操作は、アイコンによりプルダウン・メニュー(すだれが下がるように表示されるメニュー)を表示し、その中からマウスで事務用品を選択する。指定は“何をどうする”形式のオブジェクト指向であり、従来方式(ファンクション・キー中心で、“何をする、何に対して”形式)より人間の思考に近く使いやすいといわれている。

図2は直接操作環境で表計算を行っている例であり、左下の絵が時計と電卓のアイコンである。

- 2) 統一化……87年4月IBMがSAA(システム・アプリケーション・アーキテクチャ)を発表した。SAAは、同社の大型汎用機からパソコンまでのアプリケーション・プログラムに対するプログラム・インタフェースとマンマシン・インタフェースを統一するものである。また、国内でもメインフレーム各社がコンピュータ利用環境の統合化を図るアーキテクチャを発表しているが、その中でとくに注目されるのがTRONプロジェクトである。

TRONプロジェクトは、90年代の標準アーキテクチャを目指して、東京大学助教授の坂村健氏を中心として進められているプロジェクトであり、国産主要メーカを含み40社以上が参加している。TRONの特長は過去の互換性にとらわれず、新しいアーキテクチャでマイクロプロセッサ・チップからメインフレーム上のOSまでを開発することであり、またとくにマンマシン・インタフェースについては「TRON作法」として基本的な操作哲学を定め、統一を図ろうとしている。

一方、このようなアーキテクチャによるシステム統合化とは別に、マイクロ・メインフレーム・リンク (MML) 機能によるホストとワークステーションの統合も発展している。MML 機能により、ワークステーション上の操作の延長によりホストに蓄積されたデータを利用できるようになっている。

これら二つの技術により、ユーザはホスト、オフコン、あるいはパソコンなど、どのシステムであれ、同一操作環境で使うことが可能となってきている。

2.2 DS7 の現状

DS7 は日本ユニシス (当時日本ユニバック) が UTS 50 の後継機として開発し、85 年に発表したワークステーションである。DS7 の操作環境は図 3 で示すように、ホスト・システムに接続する端末機能、ワードプロセッサ機能、およびパーソナル・コンピュータ機能の三つの環境がある。

DS7 はマルチプロセスとマルチウインドウの機能を備えているため、ホストとの通信を行いながらオフライン処理を行う並列処理が可能であるが、操作の観点からみる

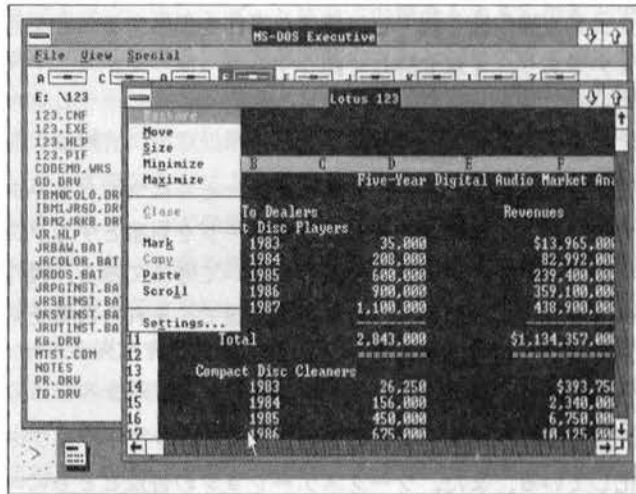


図 2 直接操作環境 (Microsoft Systems Journal Vol. 2 No. 4 より)

Fig. 2 Direct manipulation environment

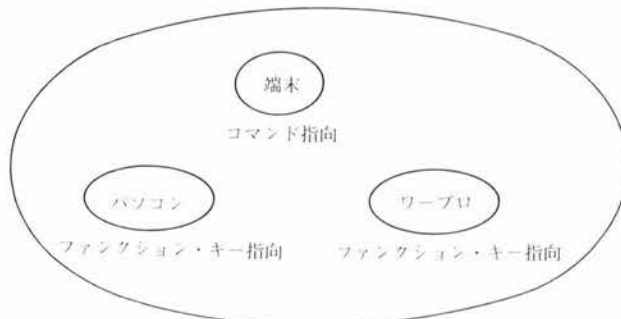


図 3 DS7 の操作環境

Fig. 3 DS7 Operating environment

とこれらは異質の感がある。というのは、ホストと端末の基本的なインタフェースは、文字入力によるコマンド指定型で半二重通信による交互会話であり、DS 7 上ソフトウェアのファクション・キーによるメニュー選択操作とは異なっているためである。これは、歴史的背景による。すなわち、タイムシェア・システム以来の伝統を引き継いでいるホスト処理系の会話システムはコマンド指定型であるのに対して、高度な半導体技術を前提にしたパーソナル・コンピュータの操作は、パソコン出現以来メニュー選択が主流になっていたからである。

これら 2 極化した操作性の違いをシステムとして吸収しているのが AOA (Automated Office Assistants) ソフトウェアである¹⁹⁾。AOA は、MAPPER を核としてホスト・システムとワークステーション・システム全体を統合することにより、ユーザに端末側で処理するかワークステーション側なのかを意識させずにドキュメントの作成や、伝票発行などの処理を可能にしている。

AOA ソフトウェアがホスト側からのアプローチであるのに対して、マンマシン・インタフェースを改善するためのワークステーション側からのアプローチとして U-VIEW がある。U-VIEW とは、すでに存在し稼働しているホスト系およびパソコン系のソフトウェアの操作性を統一し、簡便な操作を提供することを目的として開発したソフトウェアであり、4 章で詳しく紹介する。

3. マンマシン・インタフェースの設計要素

オフィスにおけるコンピュータのユーザは、とくにコンピュータ専門教育を受けているわけではない。ここでは、そのようなユーザを対象とするソフトウェアが備えるべきマンマシン・インタフェースの設計について考えてみる。操作環境のレベルとインタフェースを決定する要素について述べた後で、それらの考えに基づく画面入出力インタフェースについて論じる。

3.1 操作環境のレベル

オフィス業務のコンピュータ化の進展とともに、ソフトウェアの種類および機能が多種多様化している。また、ワークステーションの普及とともにユーザも増大しており、ソフトウェアの一貫性と共通性を保つ統一した操作環境が重要となってきた。しかし、個々のソフトウェアは異なる作業環境で開発されており、統一した操作を実現するには、そのための方法が必要となってくる。

統一の進展には図 4 で示すように、個々のソフトウェア・レベルの統一からユーザを取り巻くすべての操作環境が統一されたレベルまで 3 段階あり、そのどこを目標にするかによりインタフェース設計のアプローチは異なってくる。

- 1) レベル 1 ……ソフトウェアの起動から実行、終了にいたるまでの一貫した操作である。この場合、表計算における MULTIPLAN のように世の中で広く使用されているソフトウェアのインタフェースであれば、何々ライクなソフトウェアと呼ばれるようになる。このレベルは、ソフトウェア設計者だけの努力で実現できる。
- 2) レベル 2 ……ワークステーション・システムとして統一した操作である。表計算とかワープロなどの機能が統合化され、機能間でのマンマシン・インタフェー

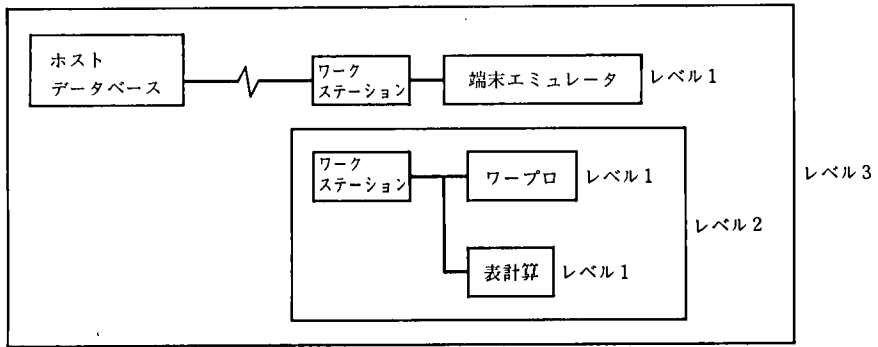


図 4 マンマシン・インタフェースの統一レベル

Fig. 4 Manmachine interface integrate level

スの相違がなく、データやファイルが共用できるレベルである。

統一対象は、①アプリケーション・ソフトウェア、②ユーティリティ・ソフトウェア、③基本ソフトウェア、④ファイル管理システムであり、核となるソフトウェア（オペレーティング・システムの場合もある）のもとで、開発ツールの完備、ライブラリの整備、開発部門間のコンセンサスなどがあれば実現できる。

- 3) レベル 3……コンピュータ・システム間における統一した操作である。オフィス業務の機械化の定型処理から非定型処理への移行、データベースの分散化、電子メールシステムなどによりマンマシン・インタフェースの影響するところは拡大しており、各メーカーが発表したアーキテクチャはこのレベルを目指している。このレベルは、統一対象システムを包含するシステム統合化アーキテクチャを確立しなければ実現できない。

3.2 インタフェースを決定する要素

インタフェースを決定する要素としてユーザ、機能、使用環境があげられる。これらをいろいろな視点から吟味することにより、どのようなマンマシン・インタフェースをとればよいか明確になる。

- 1) ユーザ……ソフトウェアのユーザを想定し、その特徴を考えてみる。
ユーザの習熟段階には、次の 5 レベルがあるという報告がある¹⁴⁾。
- ① 模倣：コマンドの合成ができない。
 - ② 初歩：コマンドの意味は理解しているが、操作は項目ごとである。
 - ③ 中間：階層構造を持つメニューや逐次入力方式を冗長かつ煩雑に感じる、作業全体のことに注意を集中する。
 - ④ 習熟：言語を駆使する。
 - ⑤ 熟練：システムを拡張して新たな機能を作り出す。

これらのレベルと一般的な操作方法とを、次のように対比させることによりインタフェースの入力形式（メニュー、コマンドなど）を決定できる。

模倣・初歩：メニュー、ガイダンス、逐次選択
 中間：メニュー、一括選択
 習熟：コマンド、一括入力

熟練 : マクロ機能による自動実行

対象ユーザを絞りこむことにより、インタフェースの標準モデルを設定できるが、経験を重ねたユーザはより速い操作を望むようになる。また同一レベルのユーザでも好みや使用環境の違いにより、使いやすさの基準は異なっている。

一つのインタフェースですべてのユーザを満足させるのは不可能である。多重階層を持つメニュー中の機能をコントロール+文字キーで呼び出す、良く使う機能を先頭メニューにもってくるなど習熟度に応じた操作や、ユーザ自身の設定による操作ができる柔軟性をそなえるべきである。

- 2) 機能……ソフトウェアがもつ機能がどういう目的をもち、どういう使われ方をするか考える。また、機能を他ソフトウェアのそれと類似している共通機能と、そのソフトウェア固有な機能に分け、共通機能は他処理系と同様なインタフェースとし、極力そのソフトウェア固有のインタフェースを少なくすることも必要である。

ソフトウェアの基本機能で使用頻度が高いならば、その機能に関するユーザの操作はすぐに上達すると考えられるので、ガイダンスは最小限にし習熟レベルの操作を前提とする。

エディタの行編集などのように連続して繰り返し使用する機能は、ワンタッチで起動しパラメタは前回の設定を継続できるようにする。複数回の指示を一度で行える考慮も必要である。

スピードより確実性を要求される機能は、メニュー選択よりも文字入力で確認させたほうがより確実である。

- 3) 使用環境……ソフトウェアが使用される環境をハードウェア面とソフトウェア面とから考えてみる。ハードウェア面においては、標準的な構成を前提として初期設定し、接続装置を自動検知しユーザ指定のパラメタを少くしたり、また環境設定ファイルを設けて事前に設定するといった工夫も必要である。

ソフトウェア面においては、ワークステーション上のソフトウェアは、ホストおよびワークステーション上のそれとの連携を前提とし分散処理として発達しているため、他ソフトウェアとの連動を考慮する。すなわち、統合操作環境をめざしたシステムにより、作業全体の一プロセスとして使われる場合があるので他ソフトウェアと違和感のない操作性にする必要がある。

また最近では MML 機能の発展とともにホストとの連動で、ユーザが設定するパラメタをコマンド・テイル（プログラムを起動するコマンドに次のコマンドを並記する）で与え、ソフトウェアの部分的機能を使用するケースも増えているため機能ごとにモジュール化しておくことも重要になってきた。

3.3 画面入出力インタフェース

現在の画面入出力インタフェースは、Star や Lisa の直接操作環境を手本とし日本語処理を考慮して発展している。ここでは、ユーザとの具体的なコミュニケーション技術である画面入出力についての考えを述べる。

- 1) 直接操作……Star や Lisa に共通している直接操作の基本技術は次の四つである⁵⁾。

① 操作の指示手順が『対象物の指示→コマンド指定→パラメタ指定』であるので、システムのすべての機能を類推しながら試行錯誤的に習得できる。たとえば文章やイメージまたは表計算の一部を他へ移すには、ユーザは、まず範囲を指定して“切出し”を選び移動先をマウスで指定する。ユーザは文章、イメージ、表などをまったく意識しなくてよい。

② WYSIWYG (What You See Is What You Get)

字体、フォント、レイアウトなどの最終結果の仕上りを、たえずディスプレイ上で認識できる。

③ Seeing and Pointing

覚えてタイプするより見て指定することによって、人間の思考過程で重要な役割を果たしている短期記憶の役割を、コンピュータ側に代行させ必要最小限の入力でコミュニケーションする。

④ モードレス

コンピュータ側の都合で設定されていたモードを、マウス選択と専用ファンクション・キーにより取り払っている。

これらのシステムは一見すると初心者向けのようだが、熟練したユーザの方が好んで使っている^[6]。熟練者は、頭の中はすでに出力結果を考えて無意識に操作をしているため、指定やタイプミスの間違いを起こしてもすぐには気づかないことが多い。このような時でも直接操作ならば、視覚的にフィードバックされるためミスに気づいて無意識に戻れるという。

直接操作では結果がすぐに視覚的に確認できるため、ユーザ中心の操作となるユーザは自分自身で好きなようにコントロールでき、やめたければ途中でやめることができる。従来のコンピュータ主導からユーザ主導への転換こそが、直接操作がマンマシン・インタフェースを改善した大きな効果である。

2) マルチウィンドウ……人間の情報処理は逐次処理であるが、意識は絶えずいろいろなところに移って連想は次々に展開していく^[7]。マルチウィンドウ機能によって実行中の作業を中断し、別のウィンドウの上で思いついた事を行った後、元の作業を再開する操作は人間の思考にあってはいるが、コンピュータ操作はむしろかしい。

一般ユーザが積極的に多数のウィンドウを開いて、複数処理を進めていくとは考えられない。むしろ、ソフトウェア自身が必要に応じてマルチウィンドウを利用することにより使いやすい環境となる。たとえば MAPPER 操作で複数のレポートを参照するとき、ユーザが新たなウィンドウを開くのではなく、他レポートの参照が指定された時点で、ソフトウェアがウィンドウを開いてレポートを表示すればよい。その方がユーザの負担はずっと軽くなる。

3) アイコン……短い時間に最大限の情報を表現できるということで、アイコン(象徴図形)によるメニュー選択が普及してきたが、次のようなアイコンの問題点の指摘もある^[2]。

① 何を図形表現するかを選択が利用効果を左右する。

② ユーザは図形表現された構成要素がもつ意味を言語と同様に学ばなければな

らない。

- ③ 図形表現が誤解を招く可能性がある。
- ④ 図形が占める画面スペースが大きい。

漢字は表意文字であり、少ない字数で抽象的な概念を表現できる。ユーザに図形文字の学習を要求するよりも、漢字表現をうまく活用することを考えた方がよい場合も多い。

- 4) マウス……カーソルを高速を移動できるため、ポインティング装置としてマウスが主流であるが、指示時間と誤り率との比較でタッチスクリーンの方が優れているという報告がある⁶⁾。日本のオフィス事情(マウスを移動するために平面が必要)や、より直接的な操作を考えると、ディスプレイ画面を指先で触れカーソル移動するタッチスクリーンへと移行すると思われる。
- 5) ヘルプ機能……ユーザはまずマニュアルを読んで操作するのではなく、使ってみてから解らないところをマニュアルで読むという傾向になってきている。マニュアル参照による作業の中断をなくす意味で、ヘルプ機能は操作性において重要な位置を占めている。ユーザがどのような状態でヘルプを求めているかをソフトウェアで判断し、適切な情報(文字だけでなく図形も)を提供することによりマニュアル以上の手助けとなる。
- 6) 日本語入力……データを入力していて最も煩わしく思うのが、通常モードと仮名漢モードの切り換えである。全角文字でのコマンド指定や、全角数字/半角数字の自動判別などをソフトウェアが考慮すればモードはなくせるのである。今後は、日本語処理を基本としてソフトウェアを設計しなければならない。

4. U-VIEW によるマンマシン・インタフェースの改善

上記の動向や問題点を踏まえて、DS7のマンマシン・インタフェースを改善するためにU-VIEWを開発した。U-VIEWは、個々のソフトウェアごとに異なる操作を統一し、メニュー方式による簡便な操作を可能にする統合化オペレータ・インタフェースである。すなわち、ホストとワークステーションとを統一する環境(3.1章のレベル3)をめざして、オフィスにおける主ユーザである模倣と初歩レベルの習熟段階のユーザを対象として、使いやすい操作環境を実現する。

U-VIEWの統一した簡便な操作とは以下のとおりである。

- ・各ソフトウェアに含まれる類似機能の名称の統一
- ・各ソフトウェアに含まれる類似機能のキー割り当ての統一
- ・プログラム・ファンクション・キー(PFキー)とスキャン・キーによるワン・タッチ入力
- ・日本語メニューでの対話方式によるコマンド選択
- ・ヘルプ機能による操作マニュアルを必要としない操作

U-VIEWが対象とするソフトウェアは、オフィス・オートメーションの代表的ソフトウェアであるワード・プロセッサ、表計算、データベースおよびグラフ・イメージである。その第一段階として、MAPPER操作を行うU-VIEW/MAPPERを開発しリリースしている。

4.1 U-VIEW の概要

個々のソフトウェアは、独自に整ったコマンド体系やメニューを持つだけでなく、バージョン・アップを重ねているソフトウェアでは互換性確保のために古いインタフェースを引き継いでいる。また、開発済みのソフトウェアを操作性向上のために修正することはかなりの工数となる。これらに対して、U-VIEW は、各ソフトウェアとユーザとの間にマンマシン・インタフェースを代行するソフトウェアを介在させ、見かけ上の操作を統一しようというアプローチである。

U-VIEW はユーザ・インタフェース管理システム (UIMS) の一種である^[9]。UIMS は、ソフトウェアからユーザとの対話部分であるユーザ・インタフェースを分離し独立させ、ユーザ自身でユーザ・インタフェースを構築しようとする試みである。

ソフトウェアにおける処理の流れは、図5で示すようにプロンプト表示に対してコマンドを入力する。入力されたコマンドに従って処理を実行し、その結果を出力したあと再びプロンプトを表示してコマンド入力を持つ。UIMS ではコマンド入力部と結果出力部を画面とし、コマンド処理を実処理として分割し、一つの作業は画面と実処理を対応させている。これらの中から必要機能を組み合わせ、ユーザの使い方にあった応用ソフトウェアを作成する。

U-VIEW では、既存のソフトウェアを対象とすること、ホストソフトウェアも含まれること、複数ソフトウェアをサポートすることなどを考え、以下の基本方針としている。

- ① U-VIEW は対象ソフトウェアのフロント・プロセッサとして動作し、ソフトウェアのキー入力と回線入力を常に監視しコントロールする。
- ② メニューやコマンドの内容は、あらかじめメニュー定義ファイルに登録し、起動時にこのファイルを読み込み処理を組み立てる。

これによりソフトウェアごとに異なる操作を統一し、ユーザの習熟レベルに合った操作環境が作れる。すなわち、

- ① 初心者ユーザは標準的に提供される操作手順にそって操作する。

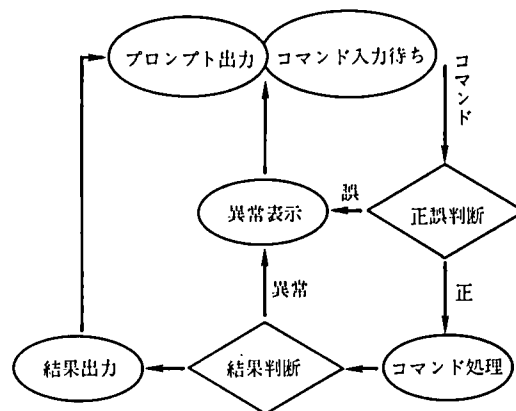


図5 応用ソフトウェアの状態遷移^[9]

Fig. 5 Status transition of application software

- ② 習熟したユーザは自分の必要な機能を好みに合わせて再構成できる。また、不足している機能があれば新たに定義し組み込める。

U-VIEW は図 6 のような構成であり、メニュー定義ファイルを入れ変えることにより種々のソフトウェアおよびユーザ・レベルに対応できる構造となっている。

4.2 MAPPER のマンマシン・インタフェース改善例

U-VIEW システムを構成するソフトウェアの一つに U-VIEW/MAPPER がある(以降 U-VIEW/MAPPER を VIEW と略称する)。

MAPPER は日本ユニシスの第 4 世代言語であり、ユーザが対話形式で処理を実行する機能(会話型処理機能)とあらかじめ一定の手順を登録しておき、それを使用し実行する機能(ラン)に大別される。

VIEW により、MAPPER の会話型処理機能をコマンド入力方式でなく、逐次メニュー方式によって行うことが可能になった。この章では VIEW によって、どのような操作に変わったかを紹介する。

図 7 と図 8 は、ソート機能を従来のコマンド入力操作で行った場合と、VIEW で行った場合の例である。

従来のコマンド入力方式では、MAPPER を操作するには、データベースの概念、コマンドの機能、実際の操作方法、さらに端末とホストとの関係を理解しなければならない。

コマンド入力操作でのソート実行の例をとると

- 1) コマンドは 1 行目の SOE 文字(▶)の後に入力。図 7 (b)では、SORT コマンドが用いられている
- 2) ソート・パラメタは数字と英字の組み合わせでマスクの下に指定(図 7 (c)では、1D つまり、営業所コードを第 1 位のキーとして降順にソートすることを示している)
- 3) リターン・キー入力という MAPPER 特有な動作
- 4) 送信・キー入力という端末動作

を知っていなければならない。

一方、VIEW では MAPPER 固有の前提知識や操作は極力排除し、一般的なソフトウェアの機能である表示・更新・保存・印書・自動実行(MAPPER コマンドもこれに

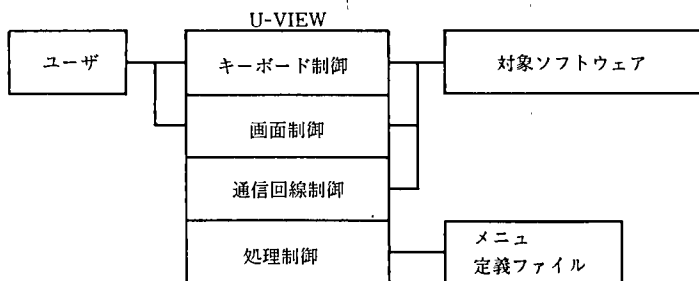


図 6 U-VIEW システムの構成

Fig. 6 U-VIEW system components

```

LINE# 1 FMT# RL# SHFT# HLD CHR# HLD LN# fcs
.DATE 17:11:44 RID 3 21 DEC 87 HANA
<< 給与台帳 3 >>
*販売員 営業所 基本給 その他 時間給 時間外
*コード 販売員氏名 コード 基本給 手当 時間給 手当 総支給額
-----
025680 秋田浩一郎 110200 316000 45000 0 23
026754 上田周造 110200 280000 34000 1300 17
068524 安藤裕二 110300 312000 32000 0 16
134525 青田功助 110200 183000 21000 840 30
145258 安本和行 110300 231000 21000 1000 10
158457 上野幸久 110300 164000 19000 740 17
201524 村木明俊 110200 359000 42000 0 16
215268 守田平次 110300 215000 32000 950 28
225400 塩原義一 110300 249000 27000 1100 25
334250 田中洋次 110300 193000 24000 870 23
335265 金子真造 110200 167000 23000 750 24
377654 平野史朗 110300 294000 33000 1300 14
388524 江口正 110200 235000 24000 1000 13
425230 石田謙二 110300 358000 42000 0 5
445205 土田八郎 110200 334000 31000 0 30
458770 中村信行 110200 308000 36000 1400 8
477582 西川清次 110300 206000 29000 930 35
523625 川合純一 110300 263000 29000 1200 10
Console#0 UTS1 ROW#01 COL#07
    
```

(a) レポート表示状態

```

LINE# 1 FMT# RL# SHFT# HLD CHR# HLD LN# fcs
.DATE 17:11:44 RID 3 21 DEC 87 HANA
<< 給与台帳 3 >>
*販売員 営業所 基本給 その他 時間給 時間外
*コード 販売員氏名 コード 基本給 手当 時間給 手当 総支給額
-----
025680 秋田浩一郎 110200 316000 45000 0 23
026754 上田周造 110200 280000 34000 1300 17
068524 安藤裕二 110300 312000 32000 0 16
134525 青田功助 110200 183000 21000 840 30
145258 安本和行 110300 231000 21000 1000 10
158457 上野幸久 110300 164000 19000 740 17
201524 村木明俊 110200 359000 42000 0 16
215268 守田平次 110300 215000 32000 950 28
225400 塩原義一 110300 249000 27000 1100 25
334250 田中洋次 110300 193000 24000 870 23
335265 金子真造 110200 167000 23000 750 24
377654 平野史朗 110300 294000 33000 1300 14
388524 江口正 110200 235000 24000 1000 13
425230 石田謙二 110300 358000 42000 0 5
445205 土田八郎 110200 334000 31000 0 30
458770 中村信行 110200 308000 36000 1400 8
477582 西川清次 110300 206000 29000 930 35
523625 川合純一 110300 263000 29000 1200 10
Console#0 UTS1 ROW#01 COL#07
    
```

(a) レポート表示状態

```

LINE# SORT FMT# RL# SHFT# HLD CHR# HLD LN# fcs
.DATE 17:11:44 RID 3 21 DEC 87 HANA
<< 給与台帳 3 >>
*販売員 営業所 基本給 その他 時間給 時間外
*コード 販売員氏名 コード 基本給 手当 時間給 手当 総支給額
-----
025680 秋田浩一郎 110200 316000 45000 0 23
026754 上田周造 110200 280000 34000 1300 17
068524 安藤裕二 110300 312000 32000 0 16
134525 青田功助 110200 183000 21000 840 30
145258 安本和行 110300 231000 21000 1000 10
158457 上野幸久 110300 164000 19000 740 17
201524 村木明俊 110200 359000 42000 0 16
215268 守田平次 110300 215000 32000 950 28
225400 塩原義一 110300 249000 27000 1100 25
334250 田中洋次 110300 193000 24000 870 23
335265 金子真造 110200 167000 23000 750 24
377654 平野史朗 110300 294000 33000 1300 14
388524 江口正 110200 235000 24000 1000 13
425230 石田謙二 110300 358000 42000 0 5
445205 土田八郎 110200 334000 31000 0 30
458770 中村信行 110200 308000 36000 1400 8
477582 西川清次 110300 206000 29000 930 35
523625 川合純一 110300 263000 29000 1200 10
Console#0 UTS1 ROW#01 COL#11
    
```

(b) コマンド入力

```

項目を選択し [リターン] を押してください
.DATE 17:11:44 RID 3 21 DEC 87 HANA
<< 給与台帳 3 >>
*販売員 営業所 基本給 その他 時間給 時間外
*コード 販売員氏名 コード 基本給 手当 時間給 手当 総支給額
-----
025680 秋田浩一郎 110200 316000 45000 0 23
026754 上田周造 110200 280000 34000 1300 17
068524 安藤裕二 110300 312000 32000 0 16
134525 青田功助 110200 183000 21000 840 30
145258 安本和行 110300 231000 21000 1000 10
158457 上野幸久 110300 164000 19000 740 17
201524 村木明俊 110200 359000 42000 0 16
215268 守田平次 110300 215000 32000 950 28
225400 塩原義一 110300 249000 27000 1100 25
334250 田中洋次 110300 193000 24000 870 23
335265 金子真造 110200 167000 23000 750 24
377654 平野史朗 110300 294000 33000 1300 14
388524 江口正 110200 235000 24000 1000 13
425230 石田謙二 110300 358000 42000 0 5
445205 土田八郎 110200 334000 31000 0 30
458770 中村信行 110200 308000 36000 1400 8
477582 西川清次 110300 206000 29000 930 35
523625 川合純一 110300 263000 29000 1200 10
Console#0 UTS1 ROW#01 COL#11
    
```

(b) コマンド選択

```

SORT
<< 給与台帳 3 >>
*販売員 営業所 基本給 その他 時間給 時間外
*コード 販売員氏名 コード 基本給 手当 時間給 手当 総支給額
-----
10
Console#0 UTS1 ROW#00 COL#26
    
```

(c) パラメタの指定と送信

```

SORT
<< 給与台帳 3 >>
*販売員 営業所 基本給 その他 時間給 時間外
*コード 販売員氏名 コード 基本給 手当 時間給 手当 総支給額
-----
134525 青田功助 110200 183000 21000 840 30
145258 安本和行 110300 231000 21000 1000 10
158457 上野幸久 110300 164000 19000 740 17
201524 村木明俊 110200 359000 42000 0 16
215268 守田平次 110300 215000 32000 950 28
225400 塩原義一 110300 249000 27000 1100 25
334250 田中洋次 110300 193000 24000 870 23
335265 金子真造 110200 167000 23000 750 24
377654 平野史朗 110300 294000 33000 1300 14
388524 江口正 110200 235000 24000 1000 13
425230 石田謙二 110300 358000 42000 0 5
第1ソート・キーを選択し [リターン] を押してください
#ソート 昇順降順 トルボ 取消
    
```

(c) パラメタ選択(キー・フィールドの指定)

図 7 コマンド入力 (上3図)
Fig. 7 Operation using command input

```

SORT
<< 給与台帳 3 >>
*販売員 営業所 基本給 その他 時間給 時間外
*コード 販売員氏名 コード 基本給 手当 時間給 手当 総支給額
-----
134525 青田功助 110200 183000 21000 840 30
145258 安本和行 110300 231000 21000 1000 10
158457 上野幸久 110300 164000 19000 740 17
201524 村木明俊 110200 359000 42000 0 16
215268 守田平次 110300 215000 32000 950 28
225400 塩原義一 110300 249000 27000 1100 25
334250 田中洋次 110300 193000 24000 870 23
335265 金子真造 110200 167000 23000 750 24
377654 平野史朗 110300 294000 33000 1300 14
388524 江口正 110200 235000 24000 1000 13
425230 石田謙二 110300 358000 42000 0 5
ソート順 パラメタメニュー
フィールド属性 昇順 降順
マイナスを含む数値
パラメタを選択し [リターン] を押してください
トルボ 取消
    
```

(d) パラメタ選択(昇順・降順の指定)

図 8 VIEW (右4図)
Fig. 8 Operation using U-VIEW/MAPPER

分類され、更新をさらに編集・探索・計算に細分している。自動実行はランに相当する)の概念的な知識があれば使用できる。

1次メニューは、25行目に表示されるPFキーで選択するバー・メニュー(図8(a))である。2次、3次メニューはウインドウ・メニューである(図8(b)~(d))。

すべての操作がメニュー選択であるため、ユーザはコマンド名、操作手順、レポート番号、ラン名を記憶する必要はない。現在実行している機能と次の動作の説明をガイダンス・メッセージとして表示し、よりわかりやすくしている。

このほか、VIEWではマニュアル参照によるオペレーションの中断をなくすためヘルプ機能を備えている。MAPPER自身もヘルプ機能をもつが、前者の方がよりリアルタイムな表示である。ヘルプの内容もマニュアル相当であり、機能・操作・操作例を表示している。機能説明では1次メニューの概要と、2次メニュー以降のパラメタの二つのレベルを用意している。

なお、VIEWはビジネス・ショウとデータ・ショウに合計3回出展しているが、ショウでのオペレータは、MAPPERの知識を持っていないことはもちろん端末操作も初めてという人も多い。事前に1時間程度デモ内容を練習するだけなので、1日目は操作ミスが多いが、慣れるに従い自分の操作が何をしているかを理解(画面をみて学習している)できるようになり、最終日にはデモ内容以外の機能を実行できるまでに至った。

以上紹介したことは、ワークステーションのインテリジェント機能だけで実現しているため、ホスト側の負荷はコマンド入力で操作した場合と変わらない。マンマシン・インタフェース機能を独立させワークステーションに分散したシステムとして、VIEWは一つの方向を示したソフトウェアといえる。

5. お わ り に

U-VIEWによる回線、およびキー入力のエミュレーションだけでインタフェースの改善が可能であることをMAPPERの例で示したが、よりきめの細かい直接的な操作を実現するには、いまのアプローチでは困難である。

このような例に、MAPPERでカーソルが表示画面の枠を越える場合、表示レポートを縦または横に自動的にスクロールする機能、画面上のデータを変更すると自動的にホスト・データベースも更新する機能、入力時点でのデータ妥当性の検査などがある。これらは、U-VIEWが画面以外の付加情報を保持しておかなければ実現できない機能である。

このような情報を効率よく伝送するには、U-VIEWとMAPPERが互いに相手を意識していなければならない。すなわちMAPPERのマンマシン・インタフェースをワークステーションに分散し独立させ、MAPPERとワークステーションとはアプリケーション・レベルのインタフェースで通信することにより実現可能である。

しかし、このレベルとなるとU-VIEWというよりMAPPER画面エミュレータの位置づけであり、統合操作環境におけるMAPPERのマンマシン・インタフェースをどうすべきかの議論をしなければならない。

U-VIEWは、のもう一つの目的である統一化を実現するためには、適用するソフト

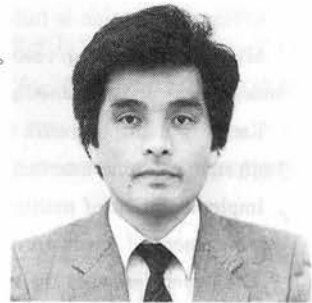
ウェアを MAPPER 以外にも拡張しなければならない。この場合、ソフトウェア固有のインタフェースを U-VIEW でどこまで吸収し、先行開発した U-VIEW/MAPPER と融合するかが課題となる。

U-VIEW 対象とするソフトウェアのフロントエンド・プロセッサのため限界はあるものの、既存インタフェースの検討・改善により、システム統合化技術や、ソフトウェアの実処理部とマンマシン・インタフェース部の最適な構成方法などの方向を示せるものと確信している。

- 参考文献 [1] 堀川勇壮 “マンマシン・インタフェースにおける一考察”, 情報処理, Vol. 24. No. 6.
 [2] B. Schneiderman, “マン・マシン・インタフェースの新時代を切り開く直接操作技法”, 日経コンピュータ, 11, 28, 1983.
 [3] “IOS システム概説書”, 481203405-0, 日本ユニバック(株), 10, 1987.
 [4] M. L. Schneider, “ユーザ・インタフェースの人間工学的設計とは何か”, 日経コンピュータ, 10, 31, 1983.
 [5] 上林憲行 “ワークステーション・ユーザ・インタフェースの構成技術”, 情報処理, Vol. 25, No.2.
 [6] R. W. Woolfield, “パソコンの新しいインタフェース技術”, 日経コンピュータ, 3, 5, 1984.
 [7] 森本正昭, 情報処理心理学, 誠信書房, p.158.
 [8] 中山剛, “ヒューマンファクタの課題”, 情報処理, Vol.27. No.10.
 [9] 岡崎哲夫・佐藤宏人, “ユーザ適応形応用ソフト構成法”, 文書処理とヒューマンインタフェース, 11, 5, 1987.

執筆者紹介 伊東 春正 (Harumasa Itou)

昭和 48 年田川工業高校機械科卒業。同年日本ユニシス(株) (当時日本ユニバック(株))入社, 90 シリーズの保守を経て, 55 年以降, 現在までは UTS 端末ソフトウェアと DS7 アプリケーション・ソフトウェアの開発に従事。



オフィス・ワークステーションのマルチメディア化動向

Trend of Multi-Media Facilities on Office Workstation

藁 科 紀 夫

要 約 最近、情報が企業の経営活動における重要な資源として認識されてきた。

オフィス・ワークステーションは、オフィスにおいて情報を創造するためのツールという役割を果たすものとしておおいに期待されている。

一方、オフィスで扱う情報は多様なメディアに存在し、それぞれ有効に利用されている。

近年、これらのメディアを直接的あるいは間接的にオフィス・ワークステーションに統合する動き、いわゆるオフィス・ワークステーションのマルチメディア化の傾向が顕著である。オフィス・ワークステーションの基本性能の向上ともあいまって、さらにこの傾向が加速されると予測される。本稿では、このオフィス・ワークステーションのマルチメディア化の動向について解説する。

Abstract It is well recognized today that information is an important resource for corporate managements.

Office workstation is fully expected to fill the role of the gears to create information in office.

Multiple media, e. g. voice and image, are utilized in office even today. However use of them is independent and not intelligent.

Recently multiple media facilities are built in or coupled to office workstation in order to improve both man-machine interface and intelligence of media.

Implementation of multiple media capabilities into workstation will be accelerated more rapidly by advancement of hardware technology, e. g. 32 bit MPU etc.

The paper describes the trend of multiple media facilities on workstation.

1. はじめに

本稿は、オフィス・ワークステーション（以下 OWS と記す）をマルチメディア化という視点から、使用者の立場で考察し、その現状と今後の動向を記述する。

ここで OWS とは、一般にデスクステーション、ワークステーション、パーソナル・システム、パーソナル・コンピュータなどと呼称されているビジネス用情報機器であり、マイクロプロセッサをベースとするものである。したがってエンジニアリング・ワークステーションと称され、用途が CAD, CAE などに特定されているものは除く。一般に、“メディア”という言葉はその用途によりいろいろと使い分けられているが、ここでは“メディア”とは情報を取り扱う機器で、かつ通信系（有線および無線）を有するものとして定義する。メディアが取り扱う情報の種類は、一般に表 1 のように分類されている。

オフィスに存在する現状の各種メディアを、機器名とそれらが使用する通信系とを対比して表 2 に示す。

表1 情報の種類

Table 1 Types of information

情報の種類		例
符合化情報		文字, 数字, 記号
非符号化情報	画像情報	絵, 写真
	音声情報	人の声, 動物の鳴き声

表2 現状の各種メディア (機器名と通信系)

Table 2 Media currently used in office
(equipments and communication systems)

機器名	通信系
コンピュータ端末	専用網, 電話網
ファクシミリ	電話網, ファクシミリ通信網
ビデオテックス端末 (キャプテン, NAPLPS)	ビデオテックス通信網, 電話網
パソコン, ファミコン	NTT PCネットワーク*, 電話網
ワープロ	電話網
電話	電話網
テレビ	無線, ケーブル
電子ファイリング装置	専用網, 電話網

* NTT PC ネットワークは, NTT-PC コミュニケーションズの
パソコン通信網である。

OWS は, 一般にはそれ自体がもともとコンピュータ端末としての機能をもっている。本稿では OWS が本来の端末機能に加えて, 他のメディアと直接または間接に連動することをマルチメディア化という。また OWS 自体がファクシミリ, ビデオテックス端末などのメディア機能をもつものを直接型, OWS が外部のメディアと密接に連動するものを間接型と定義する。

オフィスには多様な情報があふれているが, 情報の種類ごとに別々のメディアが必要な状況では, 使い勝手も悪く, オフィス空間は機械に占拠されてしまう。マルチメディア化は, 快適なオフィス空間で知的創造的な仕事を行いたいという要求に応えるために始まったもので, OWS に重要な進化をもたらすものといえる。

2. OWS への期待とマルチメディア化の必要性

2.1 OWS への期待

オフィスの仕事は, 定型なもの, 日々, 状況によって変化する非定型なものがある。非定型の仕事では, 一般に知的, 創造的な要素を強く求められ, 知恵を働かせ思考をめぐらせ, 応用動作を伴うといったように人間の最大の特性が発揮されることになる。OWS は定型な仕事はもとより, 非定型な仕事に対しても, 人間を支援してくれることが望ましい。

情報の記憶, 計算, 作表, 作図, 清書などのように人の作業を代行する機能や, データベースの検索, シミュレーションなどのように人の作業を支援する機能など, OWS に求められる機能は多様であり, 使用する人の仕事に応じてどのようにも使える道具であることが望まれる。

経営者層の仕事では、現在の電話程度の操作で、企業内のコンピュータ・システムの基幹情報が得られること、テレビ放送データベース・サービスをはじめとする外部情報が得られること、実務担当者へのメッセージや指示が送れること、そして音声・画像などの情報媒体で外部の人達と相互に通信ができることなどが要求されよう。しかも、これらが1台のOWSで可能になれば、忙しい時間を有効に活用し、多くの情報をもとに的確な判断と指示を行うことがより容易となる。

実務担当者層の人達の仕事では、情報の入力・処理・出力をいかに効率的に、いかに使用者の意のままにできるかが重要であり、多様な情報媒体を容易に使えることがさらに要求される。

これらの場合に、使用者がオフィス内を歩き廻って使用するのではなく、必要な時に各個人のデスク上からいつでも使用できることも重要であり、そのための方法や技術が強く求められている。

2.2 マルチメディア化の必要性

OWSが経営者層から実務担当者層に至るまで、それぞれの仕事に応じて真に使いやすくなるためにはいくつかの課題があるが、その主要テーマの一つがマルチメディア化である。

前述の表2の各メディアはそれぞれ特長を持ち操作も簡単なため、オフィスで利用されている。これらのメディアは、単一用途に対しては十分に役立つものである。

しかしながら発想を転換して、メディアを使用する人間を中心に考えれば、いくつかの難点があり、このことが前述のOWSへの期待を満足させ得ない原因となっている。

OWSと各種メディアが連動せず、それぞれが単独に存在する状況で経験する不満の例を列举すると以下の通りである。

- 1) メディアごとに機器が必要となり、オフィススペースが常に不足する。
- 2) メディアごとにそれぞれの通信系が必要となり、配置変更の都度、いろいろな工事が必要となる。
- 3) すべてのメディアを1人に1台与えるというのは現実的ではなく、そのために人間がメディア間を歩きまわることになる。
- 4) 他人が使用しているとすぐには使えないことがあり待たされる。
- 5) OWSで処理や加工した情報をそのままではメディアへ送出不可能。
- 6) メディアから得た情報をOWSへ取り込んで利用するために、再入力などの手間がかかる。
- 7) OWSは、コンピュータ端末、パソコン、ワープロといったメディア機能を持ち、それなりの使い勝手があるが、メーカーが異なるOWS間の文書交換すらできない状況にある。
- 8) 情報を伝える相手のメディアが異なると、それぞれのメディアが受けられる方法で情報を送らなければならない。

以上のような不満や障害を解消し、既存メディアを活用するためには、OWSがそのメディアと連動する手段が必要であり、さらにはOWSそのものが各種メディア機能を直接もつことが必要となる。

3. OWSのマルチメディア化の現状

OWSのマルチメディア化が、現実にとどの程度まで進んでいるのかを事例により紹介する。

3.1 OWSとビデオテックス

メディアとしてのビデオテックスの最大の優位性は、情報をビジュアルに、色彩豊かに表現する機能と、特別な練習をしなくても情報検索などを可能にする操作の簡便性にある。センタ側に汎用コンピュータ・システムを使用することにより、既存の情報処理システムとビデオテックス・システムを連動させて、コンピュータ・システムの利用者の枠を大幅に拡大させることが可能となる。

日本ではビデオテックスとして、CAPTAINとNAPLPSの両方式が利用されており、それぞれ対象となる利用者、使用するネットワーク、サービス内容などにより、特徴を生かした使い分けがされている。

一般には、専用のビデオテックス端末を使用してビデオテックス・システムの利用者となるが、UNISYS DS7はNAPLPS方式のビデオテックス端末としても使用可能である(図1)。

DS7のOWSとしての機能はそのままに、NAPLPSデコーダ機能をソフトウェアで実現している。デコーダの機能は「Videotex/Teletext Presentation Protocol Syntax North American PLPS」に準拠している。

DS7上のビデオテックス端末機能は、アプリケーション・ソフトウェアの一つとして位置付けられるため、業務メニューにビデオテックス・システムを含めることにより、利用者は新しく操作を覚えなくても、必要とするビデオテックスの検索やデータ入力が可能である。

ホスト・コンピュータで処理されている情報のうち、ビデオテックスの情報表現が相応しいものだけをコンピュータ内で媒体変換し、ビデオテックス情報にすればよい。

また、DS7のモニタを標準品から変更し、TVチューナ付きのモニタとして、通常のテレビ放送を受像できるようにしたものもあり、この方式ではOWSとビデオテックスとテレビとが一体となり、直接型のマルチメディア化をさらに一步前進させている。

ビデオテックス端末としての利用頻度が高い場合は、標準キーボードの代わりにワ

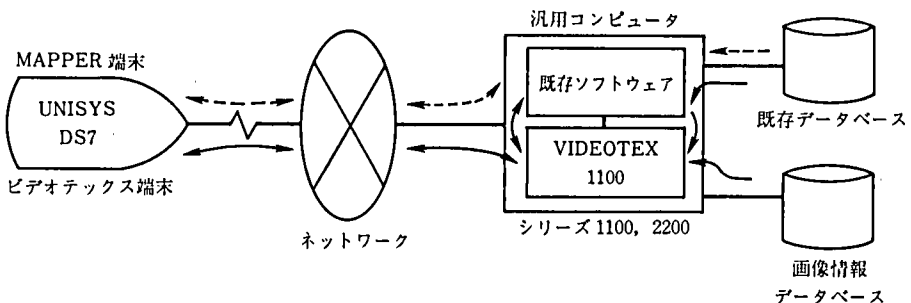


図1 UNISYS DS7とビデオテックス

Fig. 1 UNISYS DS 7 and VIDEOTEX system

タッチ型のスクリーン・キーボード（標準キーボード機能もあり）も利用できる。この場合は、キーボードにメニュー機能をもつことになり、操作の簡便性が一層向上する。

さらに他のメディアである電話との連動も可能である。コード化したダイヤル番号を受けて、相手電話を呼び出すオートコール機能をもつ電話機と組み合わせることにより、相手を指示するだけで通話が可能となる。

3.2 OWS とパソコン通信

パソコン通信はNTT-PCコミュニケーションズ(株)が高度な通信機能をもつJUST-PC (Japanese Unified Standard for Telecommunication-PC: 郵政省告示によるパーソナル・コンピュータ通信装置推奨通信方式)方式を採用したのが契機となり、ビジネス分野での重要メディアとなってきている。

OWSがコンピュータ端末として汎用コンピュータに接続されることに加えて、パソコン通信網に接続できることは非常に大きな意味をもつことになる。これは、現在、電話がそうであるように、OWSが機種を問わずに相互に通信できる機器に変身するからであり、企業内のみならず企業間、さらには一般家庭へとネットワークが拡大され、情報の送受信の範囲が既存メディア(パソコン)をそのまま使い飛躍的に広がる。OWSとパソコン通信網の接続は、既存メディアがそのまま使えるため、費用対効果が非常に大きく普及しやすい。

今後のパソコン通信は、コード情報やイメージ情報だけではなく、音声情報も切り替えて伝送する方式をとるものが主流となり、扱える情報媒体が多様になる。さらに、通信の上位層(アプリケーション層)としてJUST-MHS(電子メール通信システム)をサポートをすることによって、高度なメールサービスも可能となる。

たとえばUNISYS DS7では二種類のソフトウェアがあり、使用目的により使い分けたり、併用することができる(図2)。

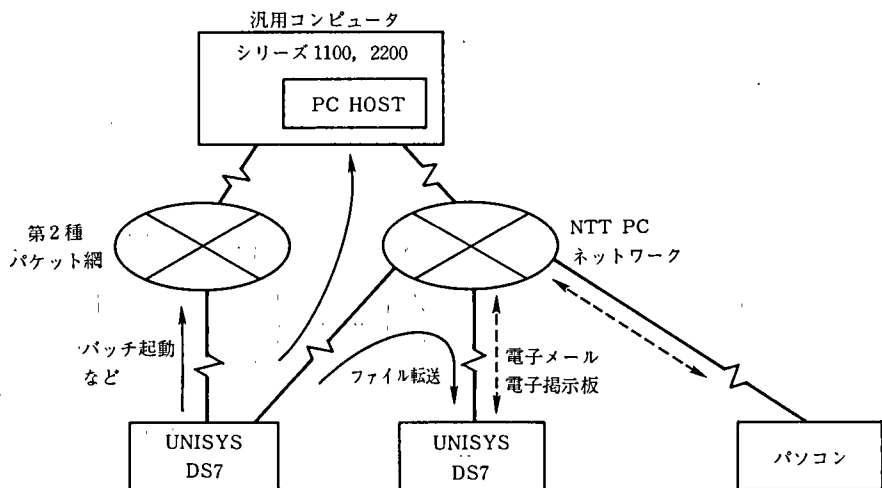


図2 UNISYS DS7 とパソコン通信

Fig. 2 UNISYS DS7 and personal computer communication system

一つは「JUSTPIA*」であり、NTT-PC コミュニケーションズ(株)の NTT PC ネットワークの電子メール、電子掲示板サービスを行うためのパソコン通信網対応ソフトウェアであり、電話網から NTT PC ネットワークへアクセスする。「JUSTPIA」は JUST-PC、CCITT および ISO 推奨の標準通信方式に準拠している。

もう一種類は「PCHOST/DS7」であり、情報センタとしての UNISYS シリーズ 1100、2200 と接続し、ファイル転送、トランザクション転送、バッチ起動などを行う。電話網から第2種パケット、または NTT PC ネットワークを通してセンタと接続する。「PCHOST/DS7」も JUST-PC、CCITT および ISO 推奨の標準通信方式に準拠している。

3.3 OWS と光ディスク文書ファイリング・システム

光ディスク文書ファイリング・システムは DRAW (Direct Read After Write Once) 型と呼ばれ、書き込みが一度だけできる光ディスクが情報記録媒体として用いられており、大量の文書をコンパクトに記録し、高速で検索できる。光ディスクは直径 5 インチで A4 版文書約 1.7 万枚、直径 12 インチで約 8 万枚(いずれも CCITT 標準原稿を 200 ドット/インチで入力した場合)分を 1 枚に収容する。密度 400 ドット/インチ、64 階調ハーフトーンでスキャナから入力できるため、文字や図形はもとより写真も相当な高品質で入力できる。プリンタへの出力も同様である。入力した文書を画面上で加工する機能も備えている。書き込みが一度だけとはいえ、大量に記録できるので不要な文書をインデックス上で消去すれば、消去書き込みが可能な記録媒体とほぼ同様に考えることもできる。

OWS と光ディスク文書ファイリング・システムとの連動は、オフィスの隅にある書類キャビネットを電子化し、OWS を仲介させることにより、OWS 上で汎用コンピュータ上の情報を電子キャビネット上の情報と組み合わせ、加工することを可能にする。イメージ・データ (圧縮しても A4 文書 1 枚で 50 K バイト程度) の処理は、OWS や

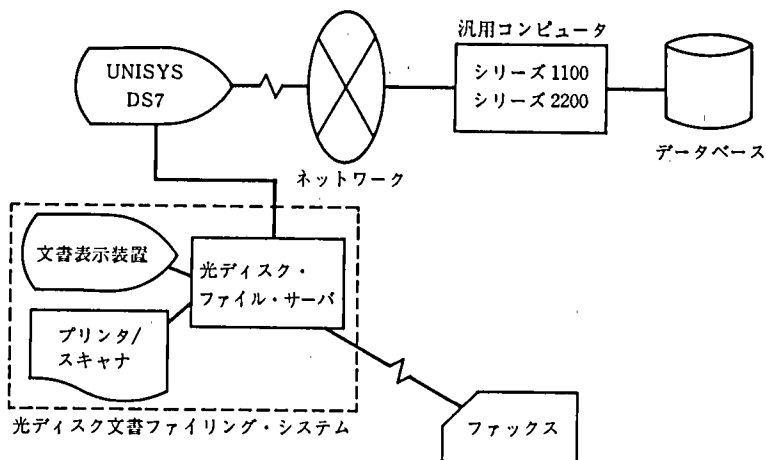


図3 UNISYS DS7 と光ディスク文書ファイリング・システム
 Fig. 3 UNISYS DS7 and optical disk document filing system

* JUSTPIA : NTT-PC コミュニケーションズ(株)の登録商標である。

汎用コンピュータにとって大きな負荷となるが、光ディスク文書ファイリング・システムにその処理を任せれば、OWS や汎用コンピュータの負担が軽減され、実用的なシステムが実現できる。たとえば、UNISYS DS 7 から汎用コンピュータの検索ソフトウェアを利用して、高度かつ複雑な検索を行い、その結果に基づき文書ファイリング・システムへ指示し、同システムによって必要な文書をディスプレイ装置やプリンタへ出力したり、遠隔地に設置されたファクシミリへ送信することができる (図 3)。

3.4 OWS とファクシミリ

OWS にアダプタを付加することによって、通信回線経由で G III ファクシミリと接続できる。

アダプタと組み合わせた OWS は、機能的にはインテリジェントなファクシミリではできなかった以下のような機能と性能がもてるようになる。

- 1) OWS 画面上のコード形式の情報をイメージ形式に変換して、直接、相手先のファクシミリへ送信できる。利点としては、スキャナ入力が必要なので、送信データの品質が向上することがあげられる。
- 2) 相手先ファクシミリからの受信データは圧縮されたイメージ・データとして OWS で扱えるため、画面上に表示し、削除・追加・修正後、送信できる。
- 3) OWS で作成したワープロ文書を電子メールとして他の OWS に送信したり、ファクシミリへ送信したりできる。
- 4) OWS がファクシミリ・コントローラとなるため、送信相手の選択、時間指定、ビジーの場合の再送信など自由に設定できる。

ファクシミリは日本国内においては設置台数も多く、さらに普及する傾向にあり、OWS との連動は実用性から見て極めて価値が高いといえる。

3.5 OWS と音声処理

現状では OWS に音声処理を取り入れている事例はまだ少ないといえるが、ある新聞社では記事の校正用に利用している。

同システムでは、OWS を入力端末として記事原稿を作成し(作成された原稿は日本語テキスト・データ)、それをミニコンによって言語処理する。この言語処理とは、音声合成処理を行う音声変換装置へ渡すまでの処理のことで、以下の内容をもっている。

- 1) 漢字仮名混じり文を辞書を利用して自立語と付属語に分解する
- 2) 読み仮名表記への変換と単語アクセントの付加
- 3) 文節アクセントの付加
- 4) 文のイントネーションの付加
- 5) CV・VC 合成パラメータの生成 (C: 子音, V: 母音)

言語処理によって生成された合成パラメータをもとに、変声変換装置が音声出力を行う。これで日本語テキストが音声として出力されることになる。校正をする人は、スピーカから流れる音声を聞きながら作業する。この場合、OWS の画面上には音声と同時に日本語文が表示されているので、誤りや修正があれば訂正する。通常、読み合わせを行う場合には読む人と見る人が必要であるが、このシステムでは 1 人でできる。

この事例では、43 万語システム辞書と無制限のユーザ辞書を用意しているほか、ミニコンを使用するなど非常に大規模なものであり、現時点では OWS のマルチメディア

ア化とは言いがたいが、将来は OWS の性能向上により可能となる領域といえよう。

3.6 マルチメディア複合機器

本稿で述べる OWS のマルチメディア化とは別に、個別メディアを複合化した製品が発表されている。電話、ファクシミリ、ワープロ、およびカード型データベース、表計算などが使える。いわゆるパソコン機能と、他のメディア機能を1台の機器に複合したマルチメディア機器である。

これらの機器の第一の特徴は、マンマシン・インタフェースにあり、画面に重ねた透明のタッチパネルに触れて指示することが操作の基本となっている。データの入力、電話やファクシミリへのダイヤリング、電卓など、それぞれの用途に応じた画面が表示され、画面上の数字などをタッチして入力または指示を行う。キーパッド、ジョイスティック、マウスも使用可能であるが、タッチパネルを通しての簡単な操作によって、使い勝手をよくしている。

第二の特徴は、オフィス内の諸作業に使用する機器類を、とにかく1台の機器にまとめたことである。とくに電話に重点が置かれて、多機能電話を中心にファクシミリがあり、これらの通信機能とイメージも扱えるワープロ、およびパソコン機能が連動している。そして、電卓、カレンダー、時計、伝言用のメモ用紙の機能がある。

電話、ファクシミリ、ワープロなどオフィスで頻繁に利用される機能に着目した製品ではあるが、OWS そのものに求められる機能、とくに情報処理に弱点がある。現状のままでは複数のメディアが一つにまとめられただけとなっているが、複合機器をベースにより高度な情報処理機能を取り込む方向へ進むものと考えられる。

4. OWS のマルチメディア化の将来動向

OWS のマルチメディア化の将来動向を考察するにあたっては、その構成要素としての OWS そのもの、その外側に存在する外部メディアおよびこの両者を結ぶ通信系とを考える必要がある。さらに、これら三者のマルチメディア化において重要となるのは、情報すなわちコード情報、画像（イメージ）情報、音声情報のうち、後二者に関する技術動向であり OWS を人間にとって使いやすくするための鍵でもある。

4.1 OWS の高性能化と低価格化

OWS のエンジンともいべきマイクロプロセッサ・ユニット（MPU）の進歩が、OWS の基本処理能力を向上させてきた。そして、今や高速処理が可能な 32 ビット MPU が登場している。また OWS の内部構造も進化し、コード・データの高速度処理だけでなく、画像情報に代表される大容量データを高速にあつかうための専用バスや画像処理専用の LSI など、情報特性を考慮したアーキテクチャが採用されている。メモリやハード・ディスクのマイクロ化と高速化も着実に進歩している。これらのハードウェア技術とオペレーティング・システムや各種ソフトウェア技術の進歩があいまって OWS の高性能化が急速に進展するため、直接型のマルチメディア化傾向がさらに強くなると予測される。

同時に、OWS の提供側における生産性向上や製造技術の進歩、利用者側における導入増加の相乗効果等によって、価格の低下や費用対効果の向上が継続し、これが OWS の直接型のマルチメディア化を普及させる大きな要因となる。

4.2 外部メディアの改良および複合化

既存の外部メディア、すなわち、電話、ファクシミリ、パソコン通信などが急激に衰えたり、他のメディアに統合されてしまうことはないであろう。電話における伝言板サービスや音声メールなどのようなサービス機能の向上、機器の操作性の改良、低価格化などによって、これらのメディアは、さらに重要性を増し普及する可能性もある。また統合される場合でも、短期間に置き換えられるわけではなく、相当の長期にわたって共存するものと思われる。

したがって OWS からみた場合、外部メディアと連動する形式である間接型のマルチメディア化も必要とされ、間接型のマルチメディア化が外部メディアとの連動ニーズを吸収する形で進行すると考えられる。ただし外部メディアの改良、通信系のマルチメディア化の進展により、将来は容易に連動が可能になるものと予測される。

4.3 通信系のマルチメディア化

通信系は日本をはじめ世界各国で高速化、大容量化、デジタル化、標準化の実現に向けて、大きな変革がはじまったところである。

ISDN (Integrated Service Digital Network : 統合デジタル・サービス網) のサービスは 64 K ビット/秒から開始され、次いで 384 K ビット/秒、1536 K ビット/秒まで拡張される。そして、狭帯域 ISDN から高速の広帯域 ISDN サービスへと拡大されてゆくことが確実視されている。

一方、企業内ではすでに高速デジタル専用回線の利用が進んでおり、LAN (Local Area Network) もようやく普及しはじめ、同時に、光ファイバ・ケーブルを使用する 100 M ビット/秒の FDDI (Fiber Distributed Data Interface) に代表される高速 LAN へと向かっている。

このようにオフィスのフロア内から広域網に至るまで、通信系の高速化は標準準拠の形で進展してゆこう。

また高速化は見方を変えれば大容量化ともいえるものであり、高速化によって画像データなどの大容量データの伝送が可能になる。

通信速度そのものの高速化とともにパケット交換の高速化も進められている。コード・データ専用の低速のパケット交換は、多数の通信網加入者が効率よく回線を利用できる利点を追求したものであった。これに対し高速のパケット交換は、時々発生する大量のデータ伝送も可能である点に特徴があり、あらかじめ広帯域幅を確保しておく必要がない。もちろん多数の通信網加入者による回線の有効利用はそのままであり、既存の低速パケット交換網と比較して非常に自由度が高い。

ISDN や高速デジタル専用回線などデジタル型の通信網では、すでに情報処理システムで培われてきた信号処理技術の適用が可能であり、「誤り」のない高度なサービスの実現と、同時に通信系に接続する OWS や汎用コンピュータの負荷を軽減できる。

このほか、間接的に通信系の価値を高める要素として画像・映像・音声の符号化の圧縮技術の著しい進展がある。その結果、帯域幅が十分ではない通信網でも画像データの通信が実用的に行えるようになるう。

このように、通信系は今後急速に発展すると思われる高速網のみならず、高速網とゲートウェイする既存の低速網、LAN まで含めてマルチメディア化へ向かって進ん

でゆくものと予測される。

4.4 音声処理技術

音声は情報処理の対象として研究されて久しいが、OWS への適用はそれほど進んではいない。音声はマンマシン・インタフェースとして、手軽な割には優れた特性を有している。たとえば、手作業と並行的に使える、特別の教育が不要である、特別の注意を払っていない相手にも伝達できるなどの特徴をあげることができる。音声は、人間が直接、使用する OWS にとって重要な要素技術である^[1]。

音声処理技術は音声認識と音声合成に大別される。音声認識は音声波を解析し、その意味内容に関する情報の抽出と決定を行う技術であり、話者を認識する技術もそれに含まれる。音声の自動認識ができ、話者認識まで可能になれば、その有効性と適用範囲の拡がりは計りしれない。音声認識に関する研究成果や製品も発表されているが、特定話者では連続発声 50 語程度、離散発声 200 語程度（いずれも認識率 95～99%）、不特定話者では離散発声で 10 数語程度（認識率 95%）のものが多い。音声そのものの性質（個体差、個人の状態による差、環境条件などによる変動）から単純な技術評価はむずかしいが、着実に進歩している。

今後は、それらの技術と人工知能技術を組み合わせて、どのように適用してゆくかが重要な課題である。音声合成^[2]は、人間が音声を発声するメカニズムを電気回路などで構成し、人工的に音声を生成する技術である。音声合成は、音声符号化技術を基本とする 3 種類の方式がある。それらは、それぞれ語彙数、音の明瞭性、自然さ、装置の複雑さなどにより、一長一短があり、任意の文章を自然な音声で出力するまでには至っていない。3.5 節で紹介したが、日本語テキスト合成装置も製品化されており、その固有技術の進展と OWS そのものの高性能化と歩調を合わせて、徐々に OWS に適用されてゆくことになろう。

4.5 画像処理技術

画像に関する処理は三つの形態に大別される。第一は操作の容易性を実現するために、ディスプレイ上に操作命令をアイコン（シンボル画像）で表示し、マウスで指示する具象処理、第二はビット列の集合としての画像情報をそのまま編集・蓄積・検索し、他のコード情報と結合させる編集処理、そして第三に画像情報をコード情報に変換し、データ処理の対象とする解析処理である。

オフィスの画像処理の主流は編集処理であり、アイコンやマウスによる具象処理が操作性の向上を支援してゆこう。現在でもアイコンやマウスは OWS へ広く適用されはじめている。編集処理は、当初は画像の専用機である光ディスク文書ファイリング・システムで行われていたが、現在では OWS でも可能なものがある。OWS そのものの高性能化と、入出力装置としてのスキャナ、プリンタ、ディスプレイ装置の高性能化・低価格化によって、OWS がこれらを直接取り込む方向をたどるのであろう。

また、記憶装置（ただし再生または読出し専用）としての CD-ROM* (Compact Disc-Read Only Memory) や CD-I** (Compact Disk-Interactive) などは、大量のデー

* CD-ROM: ROM として使用される CD (コンパクト・ディスク) で、560 MB の容量をもつ。

** CD-I: CD の発展形として音声のみならず、画像、文字データを記録する媒体であると同時に CPU、OS を定めたシステムである。

データベースをローカルに持てる点で、OWS にとっても有効に利用できる可能性があり、注目に値する媒体である。CD-ROM は、媒体の物理的フォーマットおよびディスクとディスク・ドライブ間の物理的特性のみが決められていたが、ハイシエラ・グループ (High Sierra Group) による推奨ファイル構造を原案とするものが、ISO の国際規格となることが決定している。CD-I も CD-ROM 規格を基本にファイル・フォーマット、さらにデータ処理のための CPU とオペレーティング・システムまで規定されており、標準的な装置となる可能性もある。

5. お わ り に

人間が最も得意とする能力は創造力・直観力・判断力・予見力などであろう。一方、OWS を含めて、コンピュータは膨大な量の記憶力・高速計算能力・持続力などに能力を発揮する。近年の人工知能やニューロ・コンピュータなどの研究の進展により、機械が人間の能力の一部をもつこともあり得よう。

しかし、機械はあくまで機械であり、OWS は視覚と聴覚を有効に活かした、やさしいマンマシン・インタフェースをもつ機械であり、人間が知的活動に専念できるように人間の作業を代行し、支援する道具として進歩し続けるであろう。

このための必須の要件ともいえる OWS のマルチメディア化は、今、端緒についたところであり、より使いやすい OWS を実現するために、さらなる進歩が期待されている。

- 参考文献 [1] R. A. Bolt, 鶴岡雄二訳, マンマシン・インターフェース進化論, パーソナルメディア(株), 1986, pp. 53~74.
[2] 古井貞照, 「音声合成技術とその応用」, テレビジョン学会誌, 1987, No. 8, pp. 707~715.

執筆者紹介 薬科紀夫 (Norio Warashina)

昭和 18 年生。42 年早稲田大学理工学部電気工学科卒業。同年日本ユニシス(株) (当時日本ユニパック(株)) 入社。主として商品企画業務に従事。現在, NTT 統括部 NTT 企画部に所属。



UNISYS DS7におけるコンカレント DOS XM

Concurrent DOS XM for UNISYS DS7

阿部比呂志

要約 1987年5月に出荷開始された新型の多機能デスク・ステーション DS7は、従来の DS7 と比較し、処理速度の向上およびメモリ容量の拡大が図られたプロダクトである。

この新型の DS7 では、処理速度の向上のため、マイクロプロセッサを Intel 社製の 80186 から 80286 に変更するとともに、メモリ容量の拡大のため、OS を Digital Research 社のコンカレント CP/M* からコンカレント DOS XM* に変更した。

本稿では、コンカレント DOS XM の DS7 への移植において採用したメモリ容量拡大のための技術的手法について報告する。

Abstract New DS7 is designed to obtain higher processing speed and larger memory capacity compared with its old model. In order to realize these improvements, we changed its micro engine from Intel 80186 to Intel 80286, and the operating software from Concurrent CP/M to Concurrent DOS XM released by Digital Research Inc.

This paper describes Concurrent DOS XM, centering around some technical considerations on the expanded memory capability.

1. はじめに

多機能デスク・ステーション DS7 の旧モデルは、オペレーティング・システム (OS) にマルチタスク機能をもつ Digital Research 社のコンカレント CP/M を採用したシステムで、1985年7月の出荷以来、さまざまな用途に使用されている。この中にあって、DS7 が扱える最大メモリ容量が1メガ・バイトのため、アプリケーション・プログラムによっては、メモリ不足で動作しなくなるといった事態が生じてきた。

この問題の解決のために、新たに開発したのが最大4メガ・バイトのメモリが実装可能な新型の DS7 であり、移植したのがコンカレント CP/M に拡張メモリ機能などを付加したコンカレント DOS XM である。本稿では、1メガ・バイトの壁を破るためのハードウェアとファームウェアのアーキテクチャの実現方法を中心に、新型 DS7 のコンカレント DOS XM について報告する。

2. コンカレント DOS XM の概要

本章では、コンカレント DOS XM (XM は拡張メモリ Extended Memory の略) の必要性和、内部構造について述べる。

2.1 コンカレント DOS XM の必要性

新型の DS7 (以下 DS7E と称する) が開発された目的は、旧型の DS7 (以下 DS7 と

* コンカレント CP/M とコンカレント DOS XM は、Digital Research 社の登録商標である。

称する)と比較し、以下の四つの目標を達成することである。

- 1) 処理速度の向上
- 2) メモリ容量の拡大
- 3) ソフトウェア互換性の維持
- 4) MS-DOS プログラムの実行の可能化

上記の 1)に関しては、CPU が 80186, OS がコンカレント CP/M の DS7 と比較し、CPU が 80286, OS がコンカレント DOS XM (以下 CDOS XM と称する) の DS7E では、約 2 倍のパフォーマンスが得られている。これは、80286 自身の処理速度向上によるものである。

2)の面からみると、80286 が持つ二つのアドレス・モードのうちの保護アドレス・モード (プロテクト・モード) を使用すれば、CPU 内部のメモリ管理機能により、16 メガ・バイトのメモリ空間を制御できるが、内部セグメント・レジスタの使用法の違いから、3)の目標であるソフトウェアの互換性が失われる。この互換性を維持するためには、もう一つのモードである実アドレス・モード (リアル・アドレス・モード) を使用しなければならないが、このときのメモリ空間は、1 メガ・バイトに制御されてしまい 8086 や 80186 同様 “1 メガ・バイトの壁” が立ちふさがる。

このように、2)と 3)の目標が 80286 を採用すると、相反するものになってしまう。ここで、実アドレス・モードでも “1 メガ・バイトの壁” を打ち破ろうと、独自に考案したのが CPU の外付け回路によるメモリ・ページング機構である。これにより、最大 4 メガ・バイトのメモリ空間を制御可能にし、2)と 3)の目標を同時に達成しようというものであった。

一方、DS7E が採用する OS の面では、今までのコンカレント CP/M のプログラムがそのまま実行可能で、かつ 1 メガ・バイト以上の拡張メモリをサポートする CDOS XM を採用することにより、2)と 3)の目標を達成している。さらに、この OS が MS-DOS* 用のプログラムのエミュレーション機能をもっている (そのためコンカレント CP/M XM ではなく、コンカレント DOS XM という名称になっている) ことから、4)の目標も達成される。

このように、DS7E で 2)~4)の目標を達成するためには、CDOS XM が必須のものとなった。

2.2 コンカレント DOS XM の内部構造

CDOS XM の内部構造は、図 1 に示すように、いくつかのモジュールに分かれている。基本的には、ハードウェア依存部の XIOS と、物理的なハードウェア構成からは独立した論理部のカーネルの二つからなる。

ここでは、CDOS XM の特徴である拡張メモリ機能と、DOS エミュレーション機能に関与するモジュールについてだけ簡単に触れることにする。

2.2.1 拡張メモリ制御モジュール

- 1) メモリ管理部 (MEM) ……ユーザ・メモリを一定サイズのパーティション (区画) に分割し管理する。パーティション・サイズは、システム生成時に決定され、

* MS-DOS は Microsoft 社の登録商標である。

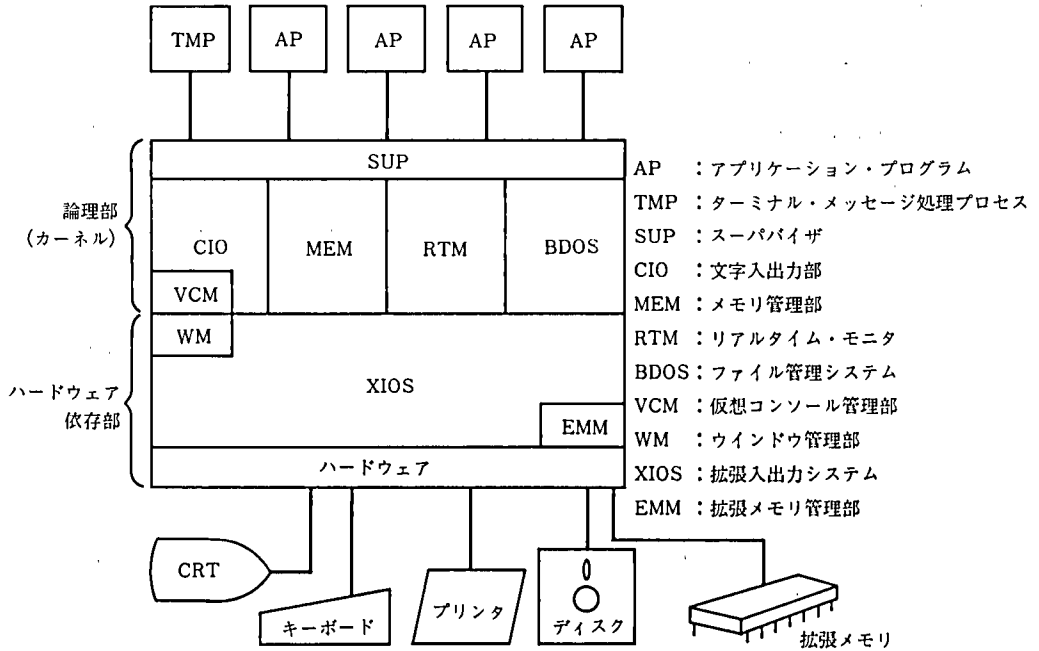


図 1 CDOS XM の内部構造
 Fig.1 Block diagram of CDOS XM

通常は 8 キロ・バイト～32 キロ・バイトである。DS7E では、16 キロ・バイトとなっている。

メモリの割り当ておよび解放などのシステム・コールは、このモジュールで処理される。

- 2) リアル・タイム・モニタ (RTM) ……リアルタイムのマルチタスク処理の核である。タスクのスケジューリングとディスクパッチング時の CPU とメモリのコンテキスト・スイッチングや、タスク間の同期と通信など、タスクに関するあらゆる管理機能を統括する。
- 3) 拡張メモリ管理部 (EMM) ……拡張入出力システム (XIOS) 内に存在し、上記の MEM と RTM からの拡張メモリ・コントロール・ファンクションを処理する。このモジュールは、その処理時間の短縮に重点を置き、当社で独自に設計・開発した。

2.2.2 DOS エミュレーション制御モジュール

- 1) ターミナル・メッセージ処理プロセス (TMP) ……コンソールから入力されたコマンドを解釈してプログラムを起動する。プログラムが MS-DOS プログラムの場合には、MS-DOS のシステム・コールを CP/M のシステム・コールに変換して実行 (エミュレート) するフィルタとして動作する。
- 2) ベーシック・ディスク・オペレーティング・システム (BDOS) ……ディスク・ファイルにアクセスするシステム・コールを処理する。CP/M フォーマットのディスク (CP/M メディア) 上のファイルのほかに、MS-DOS フォーマットのディスク (DOS メディア) 上のファイルのアクセスもサポートする。

3. 拡張メモリ機能

本章では、DS7E の CDOS XM の最大の特徴である 4 メガ・バイトのメモリ空間のサポートを実現した拡張メモリ機能について述べる。

3.1 メモリ・ページング・ハードウェア

80286 のような CPU がアクセス可能なメモリ（これを論理メモリと称する）の空間の中で、最大論理メモリより大きいメモリ（これを物理メモリと称する）の空間をアクセスするため、CPU の外側に別回路を設け、CPU がアクセスする論理メモリ空間を切り替える技術は、メモリ・ページング機構と呼ばれている。

このメモリ・ページング機構の実現方法は、図 2 に示すようにバンク・スイッチ制御により論理メモリ上に、バンク A あるいはバンク B のどちらかを割り当てるバンク・スイッチ方式が代表的なもので、DS7 がユーザ・メモリとスクリーン・メモリとの切り替えに、また古くは、UTS 20/UTS 40 が表画面と裏画面のスクリーン・メモリの切り替えに使用していた方式である。

この方法とは別に、DS7E が実現した方式は図 3 に示すように、細切れにしたページを物理メモリに割り当て、そのうちの任意のページを集めて論理メモリ上に展開する

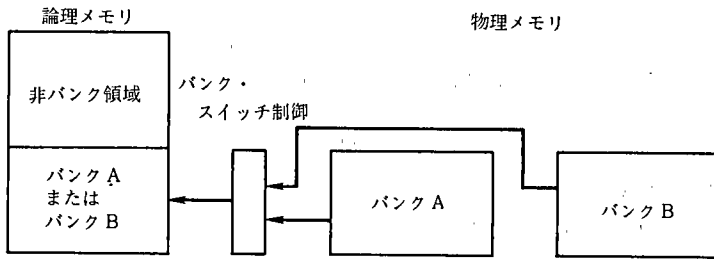


図 2 バンク・スイッチ方式
Fig.2 Bank switching method

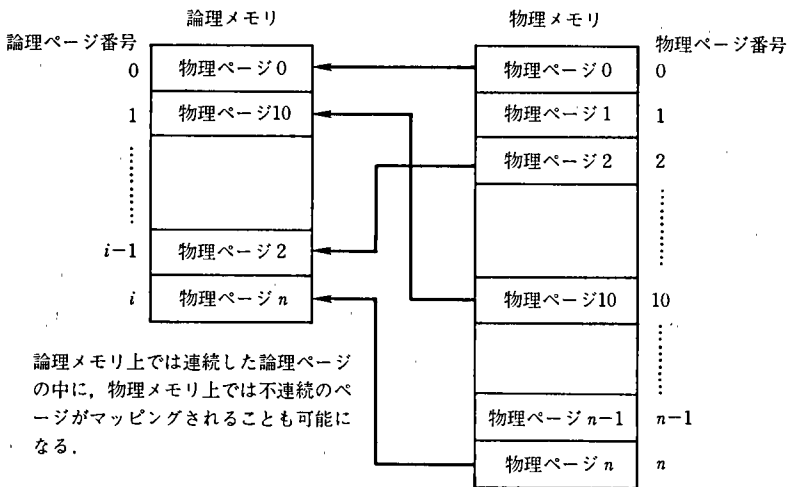


図 3 マッピング方式
Fig.3 Mapping method

マッピング方式である。このマッピング方式では、バンク・スイッチ方式と比較して、制御は少し複雑になるが、物理メモリの使用に関する自由度が高くなり、ムダも少なくできるメリットがある。DS7Eでのページ・サイズは、CDOS XMのMEMモジュールが管理するメモリ・パーティション・サイズと同じ16キロ・バイトとし、XIOS中のEMMの処理の負荷を軽減してある。

DS7Eのメモリ・ページング機構をもう少し詳しく説明したのが図4である。80286は実アドレス・モードでは、20ビットのアドレス線(A0~A19)を持ち、アクセス可能な論理メモリ空間は1メガ・バイトである。DS7Eが実装できる4メガ・バイトの物理メモリ空間をアクセスするには、もう2ビット多い22ビットのアドレス線(A0~A21)が必要となる。このため、論理アドレス20ビットのうち、上位6ビット(A14~A19)をページング・メモリのインデックスとして用い、そのページング・メモリからの出力8ビット(A14~A13)を、下位14ビット(A0~A13)に付加して、22ビットのアドレス線を生成する。

以上のようなメモリ・ページング機構を採用する上で、パフォーマンスの低下をもたらす二つの問題が生じてくる。

一つは、80286が直接に物理メモリをアクセスする場合と比較し、ページング・メモリなどの余分なロジックを経由してアクセスするので、メモリ・アクセス時間に遅延が生じる点である。この遅延が生じる場合には、80286の処理サイクルを一時停止させるウェイト・サイクルを必要な数だけ挿入する方法がとられる。しかし、1個のウェイト・サイクルが入ると、80286のパフォーマンスが20~25%低下するので、ウェイト・サイクルを必要としないロジックが必要となってくる。

もう一つは、ページング・メモリを管理するソフトウェア(DS7EではXIOS中のEMM)が、ページング・メモリをアクセスする時間が、マルチタスク動作時のタスク・

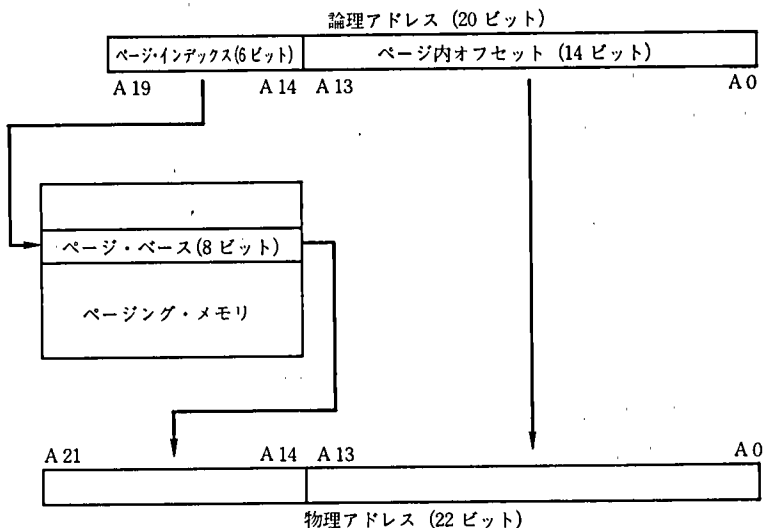


図4 DS7Eのアドレス生成

Fig. 4 Address generation in DS7E

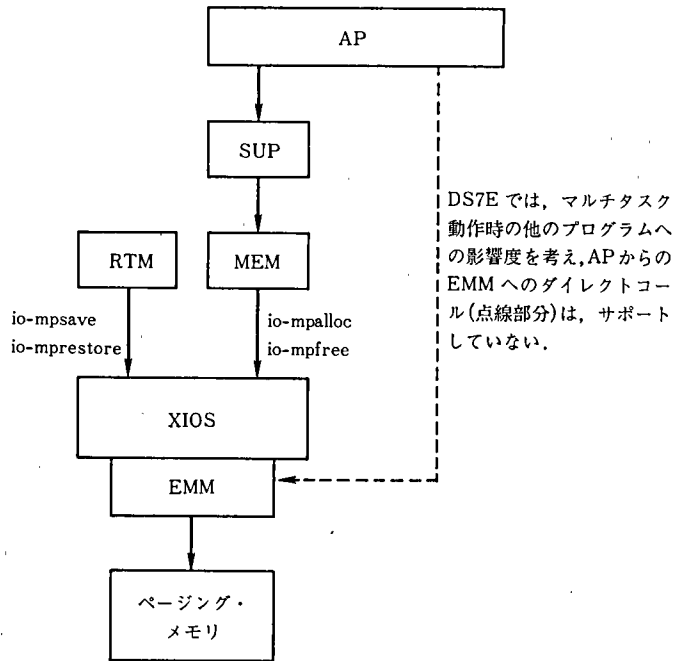


図 5 DS7E の EMS 環境

Fig. 5 EMS environment in DS7E

スイッチング効率に多くの影響を及ぼす点である。このページング・メモリのアクセス時間の最小化には、ページング・メモリ自身を 80286 のメモリ空間に置き、I/O 空間のアクセスの数分の 1 の時間で普通メモリ・アクセスと同じようにリード/ライトできるようにするロジックが必要となってくる。

DS7E では、この二つのロジックをハードウェア設計時に実現し、CDOS XM 全体のパフォーマンス低下を未然に防いでいる。

3.2 メモリ・コントロール・ソフトウェア

3.2.1 EMS の導入

1986年4月にLotus, Intel, Microsoftの3社は、IBM PC上にマッピング方式のメモリ・ページング機構をもつオプション・ボードを載せ、“1メガ・バイトの壁”を破るハードウェアとソフトウェアの規格を、EMS (Extended Memory Specification) の名で発表した^[1]。これを機に、DS7とDS7Eがこれまで採用してきたコンカレントCP/Mの供給元であるDigital Research社は、コンカレントCP/Mの上で、EMSを使用して拡張メモリ機能を実現するCDOS XMをリリースした。

DS7EのCDOS XMにおける拡張メモリのサポートは、図5に示す環境の下でメモリ管理部(MEM)、リアルタイム・モニタ(RTM)、およびXIOS内の拡張メモリ管理部(EMM)に設けた以下の4個のメモリ・コントロール・ファンクションにより実現する。

- ① io-mpalloc (Allocate memory page)
- ② io-mpfree (Free memory page)

- ③ io-mpsave (Save memory page)
- ④ io-mprestore (Restore memory page)

実際には、MEM は1メガ・バイトの論理メモリ空間内で論理メモリ・ページの割り付けおよび解放を行い、それに成功した場合に EMM に対して物理メモリの割り付け(①)および解放(②)の指示を発行する。また、RTM はタスク・スイッチング時に、EMM に対してページング・メモリの内容のセーブ(③)およびリストア(④)の要求を発行する。

3.2.2 物理メモリのマッピング

DS7E の CDOS XM では、物理メモリを16キロ・バイト単位のページに区切りメモリ管理が行われるが、その物理メモリ全体のマッピングを図6に示す。

物理メモリ・ページ全体のうち、OS、ホスト通信バッファ、ビットマップ、スクリーンおよびROM用の固定ページを除いた部分が、ユーザ・メモリとして使用できる。その大きさは、2メガ・バイト(2048キロ・バイト)システムで1488キロ・バイト(93ページ)、4メガ・バイト(4096キロ・バイト)システムで3280キロ・バイト(205ページ)となり、コンカレント CP/M でのユーザ・メモリの大きさ670キロ・バイトと比較して、2~5倍のメモリ容量をサポート可能となっている。ただし、メモリ・ディスク機能*を使用した場合には、そのメモリ・ディスクの容量の分だけ物理メモリ上のユーザ・メモリ領域は減少する。

3.2.3 論理メモリのマッピング

DS7E の CDOS XM では、コンソール0からコンソール4**まで、5画面あるコン

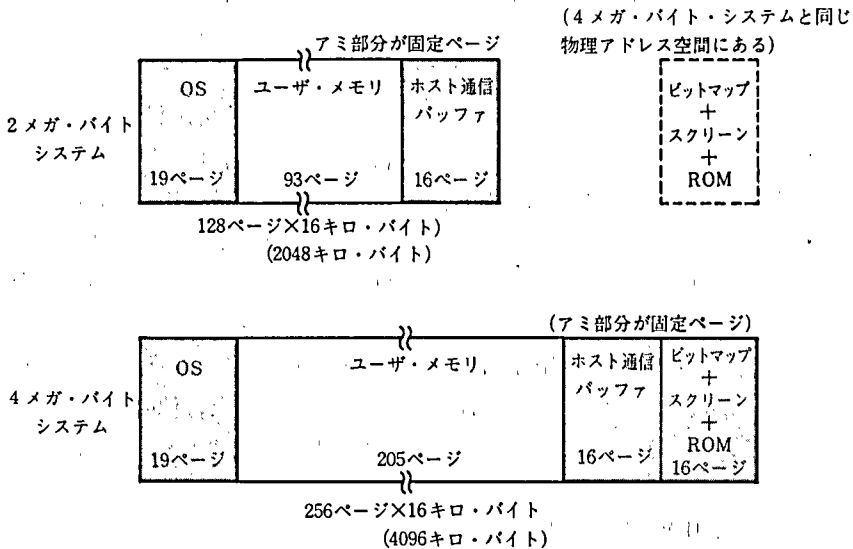


図6 物理メモリのマッピング

Fig. 6 Physical memory mapping

* メモリを補助記憶装置として使用する機能

** コンソール4は、カナ漢字変換システム(FSX)用の特殊コンソールで、通常のユーザ・プログラムでは使用できない。

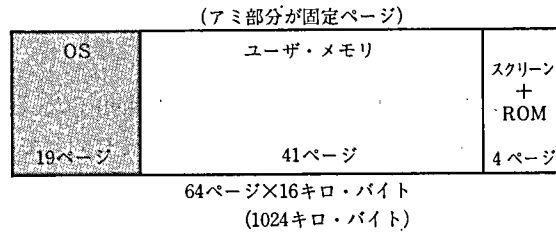


図 7 論理メモリのマッピング

Fig. 7 Logical memory mapping

ソールごとに1メガ・バイト(1024キロ・バイト)の論理空間をもたせ、他のコンソールのメモリ使用状態に関係なく、コンソールごとに独立したメモリ管理を行っている。論理メモリのマッピングを図7に示す。

論理メモリ・ページ全体のうち、OS、スクリーンおよびROM用の固定ページを除いた部分が、一つのコンソール上でユーザ・メモリとして使用できる。その大きさは、656キロ・バイト(41ページ)となり、コンカレントCP/Mの時よりやや小さくなっている。

論理メモリのマッピングの中の固定ページに、物理メモリにはあったホスト通信バッファとビットマップ用の固定ページがないのは、ホスト通信バッファとビットマップの処理の時には、ROM用の固定ページを一時的に割り付け直しているからで、論理メモリ上のユーザ・メモリ用のページ数の減少を阻止するためのものである。

このようにして、複数のコンソールのプログラムが、最大41ページまで同時に、ユーザ・メモリを使用することができるようになってきている。ただし全コンソール上で使用されるユーザ・メモリの総ページ数が、物理メモリ上のページ数を越えてはならないという制限がある。

たとえば、4メガ・バイト・システムの全コンソール上で、最大の41ページを使用した場合には、総ページ数が205ページ(41×5)となり、物理メモリ上のページ数と等しくなるので問題はない(ただし、メモリ・ディスクを使用しない場合)が、2メガ・バイト・システムでは、二つのコンソールが41ページずつ使用すると、残りの物理メモリが、93-(41×2)の11ページとなり、他の三つのコンソールでは、合計176キロ・バイト(11ページ分)のプログラムしか走らなくなる。

しかし、コンカレントCP/Mにおいて、一つのコンソールで650キロ・バイト程度の大きいプログラム(たとえば、BASIC)を動作させると、他のコンソールでは、何のプログラムも走らせることができなかつたことと比較すると、格段の進歩と言えよう。

3.2.4 論理メモリと物理メモリの結合

ここでは、EMMにより論理メモリと物理メモリとが、どのようにソフトウェア的に結合されるかを述べる。

具体例として、コンソール0とコンソール2で32キロ・バイト(2ページ)のプログラムAP0とAP2がすでに動作している時に、コンソール1で80キロ・バイト(5ページ)のプログラムAP1を起動しようとする状況を想定する。

図8に、論理メモリから物理メモリへのページの展開の状態を示す。この例では、まずMEMによってコンソール1の論理メモリ上に、AP1用として連続した5ページの論理ページが割り付けられる。そして、EMMがこの論理ページを物理メモリの未使用エリアに割り付けるわけであるが、図のように物理メモリ上での最初の未使用エリアが、5ページよりも少ない3ページしかない場合、EMMはAP1用の5ページをAP2用の2ページの後ろに割り付けるのではなく、隙間の3ページと、AP2の後ろの2ページとに分断して割り付け、物理メモリの無駄（メモリ・フラグメンテーション）が発生しないようにしている。

こうすると、物理メモリ上でのアドレスの連続性はなくなってしまうが、AP1が動作するのは、コンソール1の論理メモリ空間上であり、そこでのアドレスの連続性が保たれていれば、物理メモリ上の並びは如何であってもよいのである。

一方、論理メモリ上でのメモリ・フラグメンテーションの問題については、コンカレントCP/Mのように単一のメモリ空間へ、複数のコンソールのメモリ・パーティションを割り付ける場合には、かなり発生するが、CDOS XMではコンソールごとにメモリ空間を持つため、ほとんど発生することはない。

3.2.5 EMMの詳細

3.2.1項で触れた4個のメモリ・コントロール・ファンクションが、EMMでどのよ

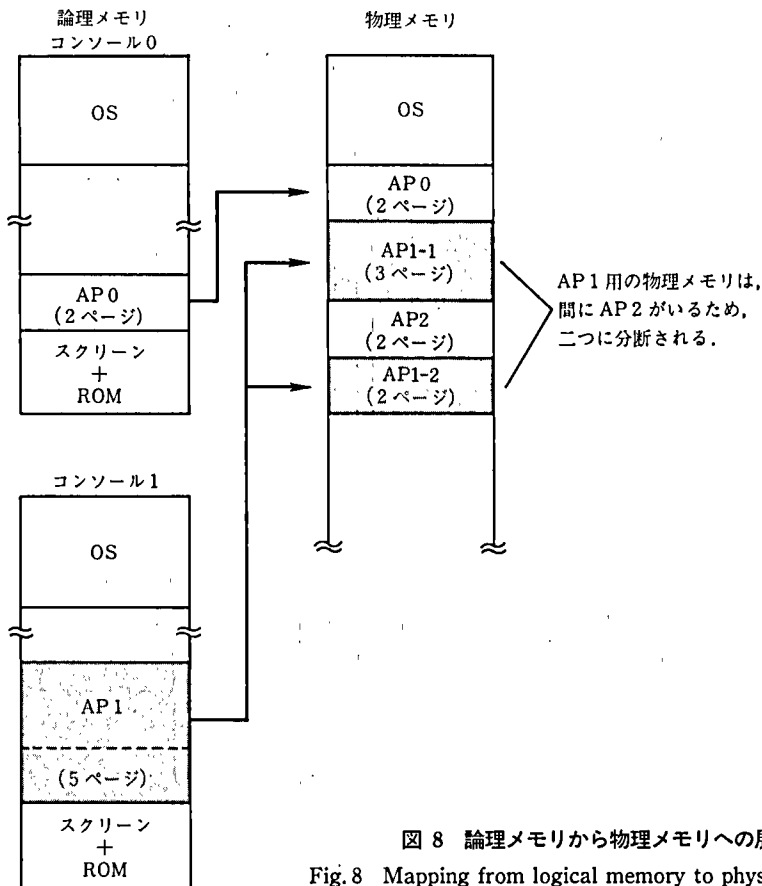


図8 論理メモリから物理メモリへの展開

Fig.8 Mapping from logical memory to physical memory

うに処理されるかを、図9を用いて説明する。

EMMは2種類のテーブルを管理する。一つは物理ページ・テーブルで、最大物理ページ数(256個)と同数のエントリを持つ。このテーブルのインデックス(00H~FFHまでの8ビットで表現される)が物理ページ番号を示し、エントリ中の情報はその物理ページが、固定ページ(使用できないページ)、使用中ページ、あるいは未使用ページのどれかを表す。また、未使用ページが何ページ残っているかを示す未使用ページ・カウンタを設け、テーブルの空き状態をチェックするときに、いちいちテーブル全体をサーチしなくてもすむようにしてある。

もう一つのテーブルは、論理ページ・テーブルで最大論理ページ数(64個)と同数のエントリを持つ。エントリの中には、その論理ページが割り付けられた物理ページの番号(つまり物理ページ・テーブルの8ビット・インデックスで、この8ビットがそのままページング・メモリの内容となる)を格納する。同じテーブルをコンソールの数(5個)だけ用意し、フォーマットも実際のページング・メモリと同一にすることによって、タスク・スイッチング時の論理メモリ空間の切り替えを、論理ページ・

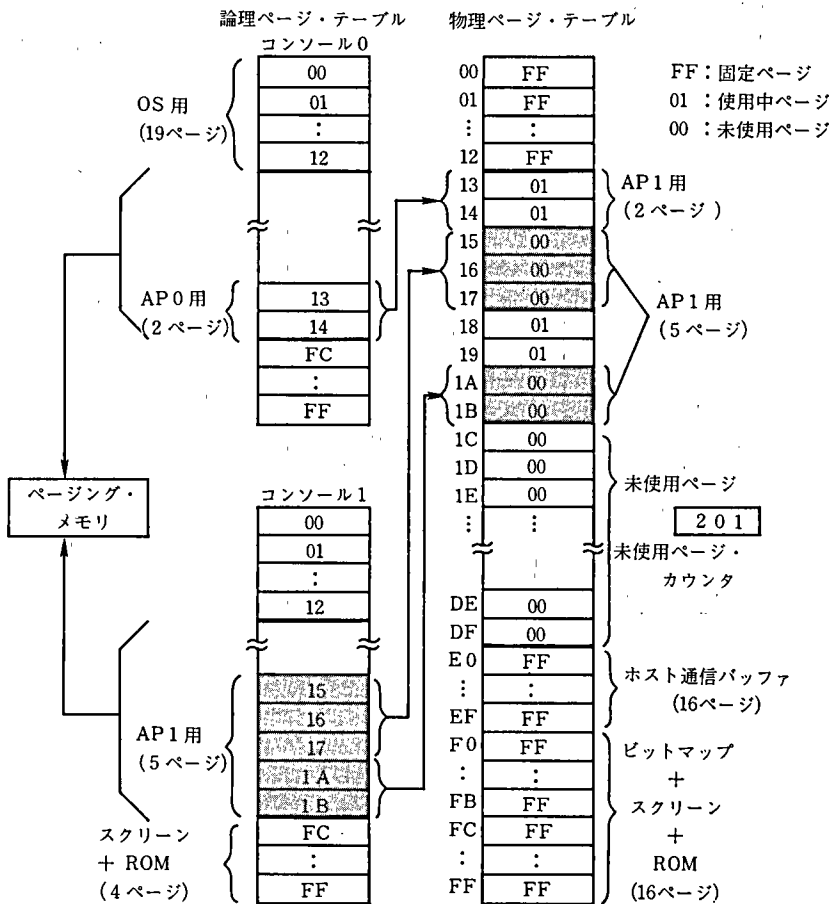


図9 EMMのテーブル制御

Fig.9 EMM table control

テーブルからページング・メモリへの単なるメモリ・コピーですむようにしてある。

EMM がメモリ・コントロール・ファンクションを受け取った際には、引数パラメータとして、コンソール番号、最初の論理ページ番号、および処理するページ数が算出可能な MPAD (Memory Page Allocation Descriptor) というストラクチャを得る。

EMM は、この情報を元に、各ファンクションの処理を実行する。

以下にファンクションごとの処理の詳細について説明する。

1) io-mpalloc の処理……3.2.4 項で想定した状況での AP1 の起動時の物理メモリの割り付け方法について述べる。

- ① 処理ページ数と物理メモリの未使用ページ・カウンタとを比較し、未使用ページ数が少ない場合は処理を中止しエラーを返す。
- ② コンソール番号から、該当する論理ページ・テーブルをポイントする。
- ③ 物理ページ・テーブルから未使用ページを見つけ、使用中ページに変えるとともに、そのページ番号を論理ページ・テーブルに格納する。このとき、未使用ページ・カウンタを減少 (-1) させる。
- ④ 処理ページ数だけ③の処理を繰り返す。
- ⑤ 論理ページ・テーブルの内容をページング・メモリにコピーする。

2) io-mpfree の処理……AP1 の終了時の物理メモリの解放の方法について述べる。

- ① コンソール番号から、該当する論理ページ・テーブルをポイントする。
- ② 処理する論理ページのエントリの内容を取り出し、それをインデックスとした物理ページ・テーブルのエントリを使用中から未使用の状態に変える。このとき、未使用ページ・カウンタを増加 (+1) させる。
- ③ 処理ページ数だけ②の処理を繰り返す。

3) io-mpsavae の処理……RTM がタスクをスイッチ・アウトする時に使用するファンクションであるが、EMM では常にページング・メモリと現在の論理ページ・テーブルとを同じ内容に保っているため、このファンクションでは何の処理も行わない (No-Operation)。

4) io-mprestore の処理……RTM がタスクをスイッチ・インする時に使用するファンクションで、EMM は該当するコンソール用の論理ページ・テーブルの内容を、ページング・メモリにコピーするだけの処理を行う。

以上のように、EMM はできるだけ短時間でメモリ・コントロール・ファンクションを実行できるような設計にした。

とくに、io-mpsavae と io-mprestore の二つのファンクションは、マルチ・タスク動作時に生じる 16.7 ミリ秒ごと (1 秒間に 60 回) のタスク・スイッチングの際に必ず実行されるので、その処理時間を極小に抑え、メモリ・コントロールのためのオーバーヘッドの増加によるパフォーマンスの低下を防ぐよう細心の注意を払った。

3.3 パフォーマンス評価

DS7E に CDOS XM を採用した場合、拡張メモリ機能のサポートのためのオーバーヘッドが、全体のパフォーマンスを著しく低下させるのではないかと懸念が早くからあった。

表1 メモリ・コントロール・オーバーヘッドの測定
Table 1 Measurement of memory control overhead

[テスト環境]

CPU使用率100%のアセンブラ・プログラムを二つの
コンソールで同時に動作させる。

コンソール	プログラム概要
0	6個のレジスタのPUSHとPOPを 270万回繰り返した場合の時間を測定
1	LOOPインストラクションによる無限 ループ

[テスト結果]

機種	拡張メモリ 非使用時 ①	拡張メモリ 使用時 ②	メモリ・コントロール・ オーバーヘッド (②-①)/①×100
DS7E	41.05秒	41.30秒	0.61%
IBM PC/AT	72.73秒	74.12秒	1.91%

そこでIBM PC/AT上のEMS環境で動作するCDOS XMで測定されたオーバーヘッド1.91%と比較して、DS7EのCDOS XMでのオーバーヘッドがどうなるかを、同一テスト環境で測定したのが表1である。

この結果、DS7Eでのオーバーヘッドは、PC/ATの約3分の1(0.61%)というかなり低い結果が得られた。これは1時間かかるジョブだと22秒の遅延となるが、メモリ容量の増大のメリットと天秤をかけた場合、無視できるものであり、このことからDS7Eの拡張メモリ制御用のハードウェアとソフトウェアの設計が正しいものであったと考えられる。

4. おわりに

日本でのCDOS XMの移植は、DS7Eが初めてであることから、各種の創意工夫を必要とした。ことに、拡張メモリ機能による“1メガ・バイトの壁”への挑戦は、この機能を使用することによって生じるシステム・オーバーヘッドを、米国IBMのPC/ATよりもかなり小さく抑えることができ、満足できる結果が得られた。

16メガ・バイトのメモリ空間をCPU自身でサポートする80286本来の保護モードでは、アプリケーション・プログラムの互換性が保たれないため、保護モードを使用したOSは市場に浸透していない。“1メガ・バイトの壁”を持つ実アドレス・モードを使用するOSが主流となっている。この中において、DS7Eが提供する4メガ・バイトのメモリ空間は、ユーザにとってメリットが大きいと思われる。

参考文献 [1] R. Duncan, “8086マシンのメモリを8 Mバイト拡張するEMS”, 日経バイト, Mar. 1987, pp. 137~147.

執筆者紹介 阿部 比呂志 (Hiroshi Abe)

昭和50年国立宮城工業高専電気工学科卒業。同年日本ユニシス(株)(当時日本ユニバック(株))入社。テクニカル・サポート担当エンジニアを経て、54年よりマイクロ・プロダクトの開発に従事。現在ワークステーション開発部開発一課に所属。



MMLの実現

Micro Mainframe Link Construction

河合 昭 男

要 約 情報処理の分散化が進行している。水平分散として、メインフレーム間は WAN (Wide Area Network), ワークステーション間は LAN (Local Area Network), 垂直分散としては MML (Micro Mainframe Link) の諸技術が実現されている。

近年、ワークステーションが高機能化するとともに、今までメインフレームで行っていた作業を分担できるようになってきた。そして、その割合は今後増加する方向にある。この作業分担を円滑に行うための一つの基礎技術が MML である。

われわれは、MML を MAPPER 1100 と DS 7 間にマイクロ・マップパー・リンクとして実現した。本稿では、その一つの事例として、DS 7 上のイメージ処理プログラム U-IMAGE をとりあげ解説する。また、このシステムでは MAPPER 1100 を核としたため、ファクシミリ網や電話網に接属されたファクシミリ端末と U-IMAGE は、マップパー・レポートとしてイメージ・データを共用できる。

MML の目指すものは、ホストとワークステーションが一体となった統合操作環境であり、これが今後の課題と言える。

Abstract Information processing trends towards distributed processing. WAN for mainframes and LAN for workstations are used as horizontal distribution technique. MML is used as vertical distribution technique. Recently, capacity of workstations is increasing so that some mainframe tasks can be executed by them.

We implement a MML between MAPPER 1100 and DS 7 named micro-MAPPER-link. This paper describes an image data micro-MAPPER-link. The image data edited by DS 7 U-IMAGE is able to be transmitted to facsimile through MAPPER 1100, and vice versa.

The target of the MML is to construct a unified operation environment of a mainframe and a workstation.

1. はじめに

情報処理の方式は、集中処理から分散処理へ移行しつつある。1970年代頃は、大型汎用機によるバッチ処理や端末を使った TSS 等の集中処理が中心であった。70年代後半になると、コミュニケーションの技術に関心が集まり、ネットワーク・アーキテクチャが各メインフレームから発表された。LAN の代表とも言える Ethernet が開発されたのはこの頃である。同一アーキテクチャ間の通信には見通しがついたが、ユーザの立場から見ると、異なるメーカー製機種種の接続が強く要請されるのは当然であった。ISO で OSI の標準化作業が開始されたのはこの頃である。

1980年代に入ると、LSI, VLSI およびマイクロプロセッサの技術が目ざましい進歩を遂げ、パソコンやワークステーションが普及してきた。ワークステーションの高機能化による、水平分散や垂直分散が唱えられ始めたのはこの頃からである。水平分散

は、メインフレームでは WAN, ワークステーションでは LAN として実現される。垂直分散を実現するのが、80 年代後半に入って注目されてきたマイクロ・メインフレーム・リンク (MML) である。WAN や LAN は、異機種間接続できるように、OSI 標準化が進められているが、MML は、独自の方法で各メインフレームが異なるアーキテクチャのメインフレームとワークステーション間で開発しているのが実状である。

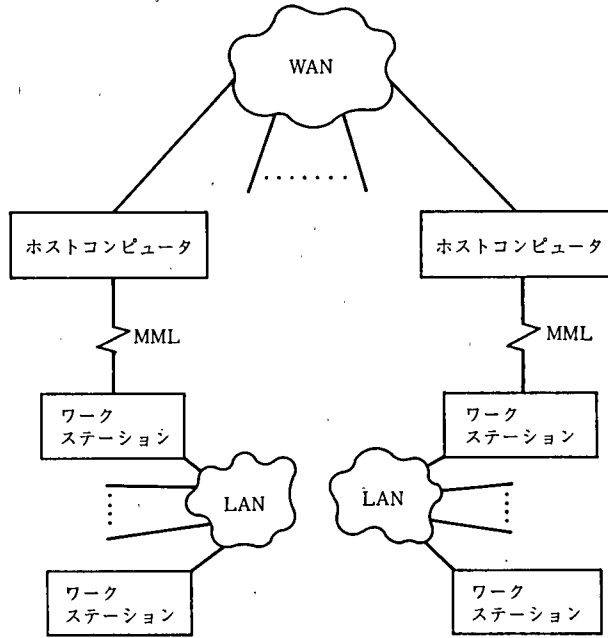


図 1 分散処理

Fig.1 Distributed processing

2. MML とは

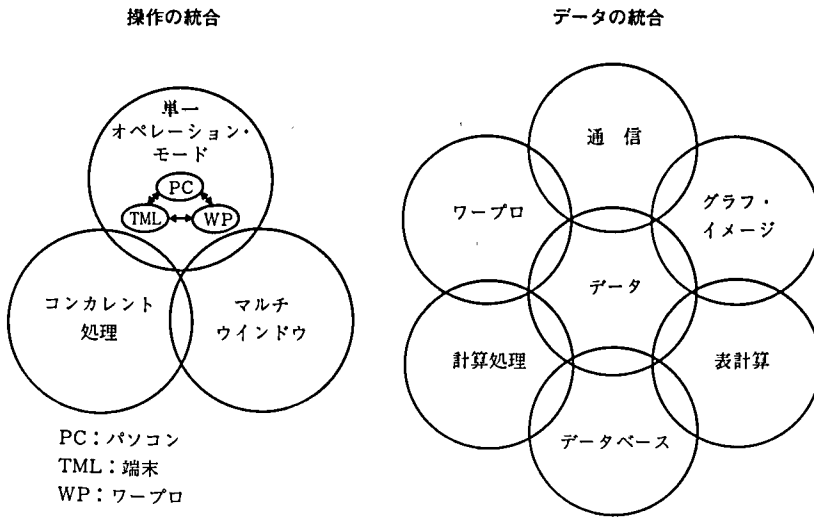
MML は、メインフレームとワークステーションの垂直分散を実現するための手法である。大型汎用機による集中処理が行われていた 70 年代、ワークステーションはまだ未発達な状態で、単にデータの入出力を行う端末としてのみ使われていた。80 年代になり、マイクロプロセッサ技術の進歩により、端末はインテリジェンスを持つようになり、ワークステーションとしての地位を確立した。そこでは、今まで大型機で行ってきた作業の一部が分担できるようになり、その分担の割合はますます増加しつつある。MML の目的は、この作業分担を円滑に行えるようにすることにある。

MML は、統合操作環境実現のための一翼を担うものである。統合操作環境とは、操作性の統合とデータの統合を意味する。ワークステーションの側から見た統合操作環境は、内部統合、水平統合および垂直統合の三つに分けて考えられるだろう。内部統合とは、1 台のワークステーションを端末、ワープロまたはパソコンと切り替えて使用するとき、その操作性を統一し、データの共有化を図るものである。水平統合とは、ワークステーション同志を LAN 接続して、プリンタ等のハードウェア資源やデータの共有化を図るものである。垂直統合とは、ホストとワークステーションの一体化を

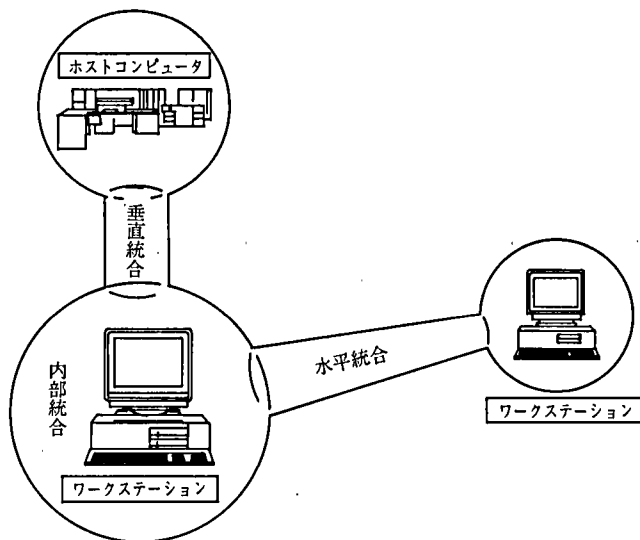
目指すもので、その手段がMMLである。

MMLの実現レベルは次の4段階に分けて考えられる:

- レベル0……ワークステーションは単なる端末機能のみで、MML以前のレベル。
- レベル1……ファイル転送レベル、まだMMLとは呼べない。
- レベル2……ファイル変換を行い、ホストとワークステーション間でデータ共有ができるレベル。このレベルからMMLレベルと言える。しかし、使用者はホストとワークステーションを意識して使い分けなければならない。



(a) 操作の統合とデータの統合



(b) 統合の形態

図2 統合操作環境

Fig.2 Unified operation environment

- ・レベル3……MMLの目標とするレベル。使用者にホストとワークステーションの違いを意識させない。

3. マイクロ・マップパー・リンク⁽⁴⁾

ホストとして UNISYS シリーズ 1100, ワークステーションとして UNISYS DS 7 を対象とした MML の実現手法について述べる。MML の核となる基本ソフトウェアは、ホスト側は第 4 世代言語 MAPPER 1100, ワークステーション側は U-MENU である。この MML を、とくにマイクロ・マップパー・リンクと呼ぶ。

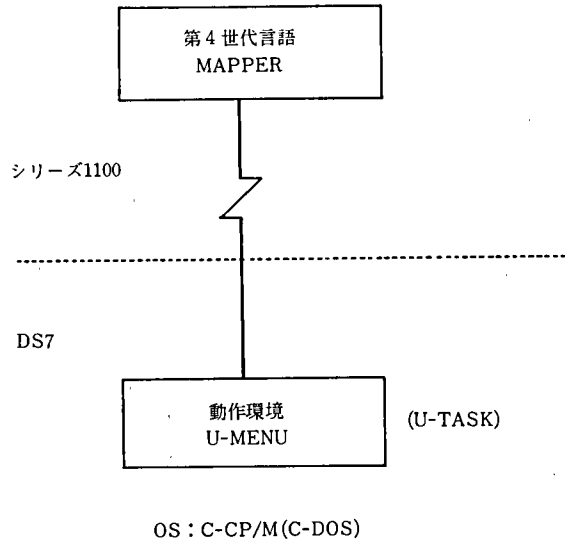


図 3 マイクロ・マップパー・リンクの基本ソフトウェア

Fig. 3 Basic software for micro-MAPPER-link

3.1 マイクロ・マップパー・リンクのねらい

マイクロ・マップパー・リンクのねらいは、次の 2 点である。

- 1) マッパー・コマンドで DS 7 固有の機能が使えること……本来、マッパー・コマンドは、インテリジェンスを持たない時代の端末を対象とし、ホストでの処理を前提としたものだった。マイクロ・マップパー・リンクを用いることにより、DS 7 固有の機能をマッパー・コマンドの延長として使用し、マッパーの機能拡張ができる。
- 2) マッパー・レポートを DS 7 からアクセスできること……ホストにあるデータを DS 7 上のアプリケーションから参照することや、逆に DS 7 上のアプリケーションで作成したデータをホストに転送することができる。

MML の実現手法として、ホスト主導型とワークステーション主導型が考えられる。1) はマッパー・コマンドを主体としたものでホスト主導型 MML (図 4(a)), 2) はワークステーションのアプリケーションを主体としたもので、ワークステーション主導型 MML (図 4(b)) と見ることができる。

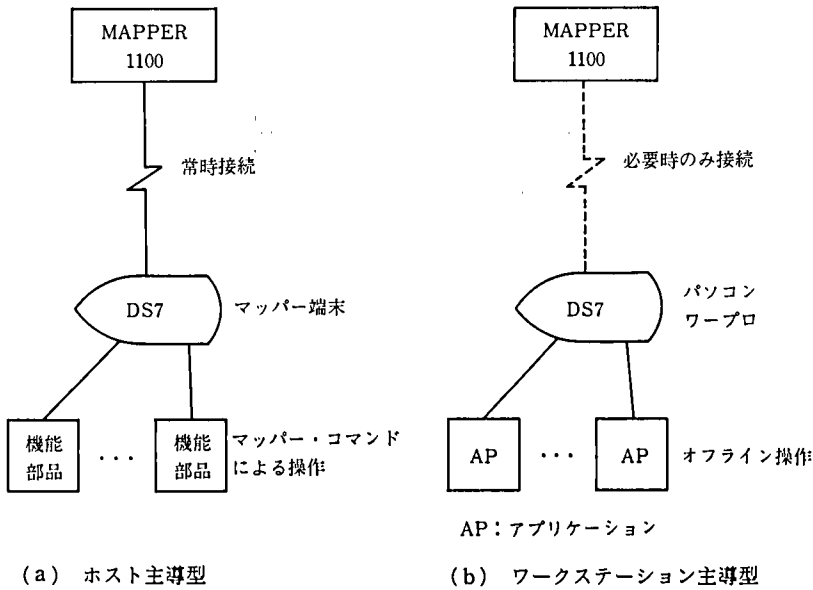


図 4 ホスト主導型とワークステーション主導型
Fig. 4 Host oriented and workstation oriented

3.2 マイクロ・マッパー・リンクのしくみ

DS 7 上に搭載されているデジタルリサーチ社製の C-CP/M は、インテル 8086 系マイクロプロセッサ上でいち早くコンカレント処理を実現した OS である。U-MENU には、この OS のマルチタスク機能が最大限利用されている。その結果、比較的容易にマイクロ・マッパー・リンクのための機能が実現できた。

C-CP/M は、機能強化されて C-DOS としてリリースされた。それに伴って、U-MENU も機能強化されて U-TASK としてリリースされている。本稿は U-MENU について述べるが、U-TASK についても以下の機能は同一である。

図 5 にマイクロ・マッパー・リンクのしくみを示す。マイクロ・マッパー・リンクを使う前提として、UTS エミュレータは U-MENU より起動されていることが必要である。このとき、U-MENU は親プロセス、UTS エミュレータは子プロセスである。

今、DS 7 よりマッパー・コマンドが入力された場合を想定する。コマンドが、UTS エミュレータを経由して MAPPER 1100 に送られると、MAPPER 1100 は、そのコマンドに対応するメニュー画面を出力する。これに対してパラメータをセットして送信すると、MAPPER 1100 はタスク・リンク・メッセージを送り返す。

U-MENU は、このメッセージを認識すると、UTS エミュレータ・プロセスを一端中断し、DS 7 上の適当なモジュールを子プロセスとして起動する。起動されたプロセスは、ステータスを MAPPER 1100 に送る。データ転送が必要なら、この時点から、このプロセスと MAPPER 1100 間で通信を行う。処理が完了すると、このプロセスは MAPPER 1100 に終了ステータスを送り、自分自身も終了し、親プロセスにコントロールが戻る。そこで、U-MENU は中断していた UTS エミュレータを再開する。

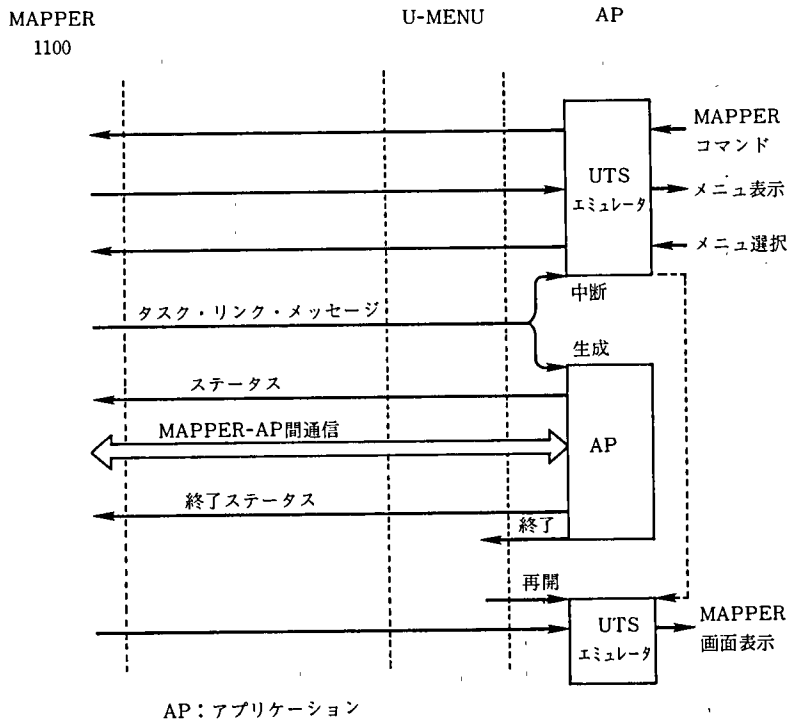


図 5 マイクロ・マップパー・リンク

Fig.5 Micro-MAPPER-link

4. イメージ処理のマイクロ・マップパー・リンク

この章では、マイクロ・マップパー・リンクの適用事例として、DS7イメージ処理プログラム U-IMAGE について解説する。

4.1 U-IMAGE の構造

U-IMAGE は、モジュール単位に単機能化することを目指してデザインされている。また、各モジュールは、パラメータ設定等のオペレータ操作が要求されるオペレータ介入モジュールと、要求されない実行モジュールとに分類されている。

具体的に述べると、U-IMAGE は、オフライン処理のために六つ、オンライン処理のために三つ、計九つのモジュールから構成されている。オペレータ介入モジュールは、オフライン処理主モジュール (UIMAGE) のみで、他はすべて実行モジュールである。とくに、オフライン処理の五つの実行モジュール(編集、文書作成、スキャナ、ファイルおよび印書) は、各々単独でも実行可能である。

4.2 U-IMAGE と MAPPER1100 の接続

U-IMAGE およびイメージ・データは、マップパー・コマンドにより、マイクロ・マップパー・リンクの機能を用いてホスト側から制御することができる。

DS7で作成したイメージを MAPPER 1100 に送るには二つの方法がある。

- 1) ファイル転送 (DKT コマンド)
- 2) グラフ画面を読み取ってファイルを経由しないで、直接 MAPPER 1100 に送信 (IMI コマンド)

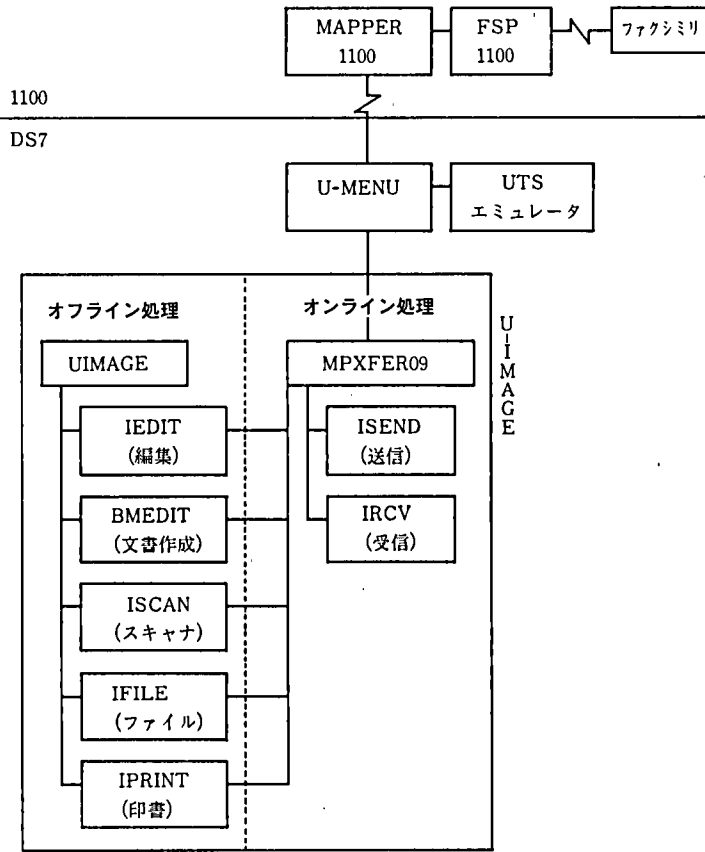


図 6 U-IMAGE のモジュール構造
Fig. 6 U-IMAGE module structure

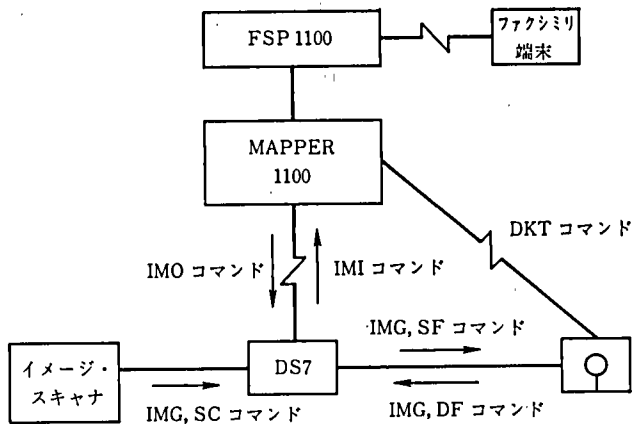


図 7 U-IMAGE とマップパー・コマンド
Fig. 7 U-IMAGE and MAPPER commands

逆に, MAPPER 1100 から DS 7 にイメージを送る方法も二つある。

- 1) ファイル転送 (DKT コマンド)
- 2) MAPPER 1100 から送信されてきたイメージ・データを, ファイルを経由しないで直接グラフ画面に書き込んで表示 (IMO コマンド)

オンライン処理の各実行モジュールは, IMG コマンドにより起動することができる。

- イメージ編集 (IMG, ED コマンド)
- 文書作成 (IMG, BE コマンド)
- イメージ・スキャナ (IMG, SC コマンド)
- ファイル保存 (IMG, SF コマンド)
- ファイル読み取り (IMG, DF コマンド)

図 6 の図式は可換である。たとえば, イメージ・スキャナより読み取って DS 7 グラフ画面に表示したイメージを IMG, SF コマンドによりファイル保存した後, DKT コマンドによるファイル転送を行ったイメージと, IMI コマンドにより直接転送したイメージは同一である。逆向きも同様である。

5. U-IMAGE とファクシミリ接続

イメージ・スキャナから読み取ったイメージを, ファクシミリ網または電話網に接続されているファクシミリ端末に送信したり, 逆にファクシミリ網または電話網に接続されているファクシミリ端末から送られてきたイメージを, DS 7 のグラフ画面に表示することができる。

U-IMAGE は, G 3 ファクシミリと互換性のある MH 圧縮と, ノーマル/ファイン

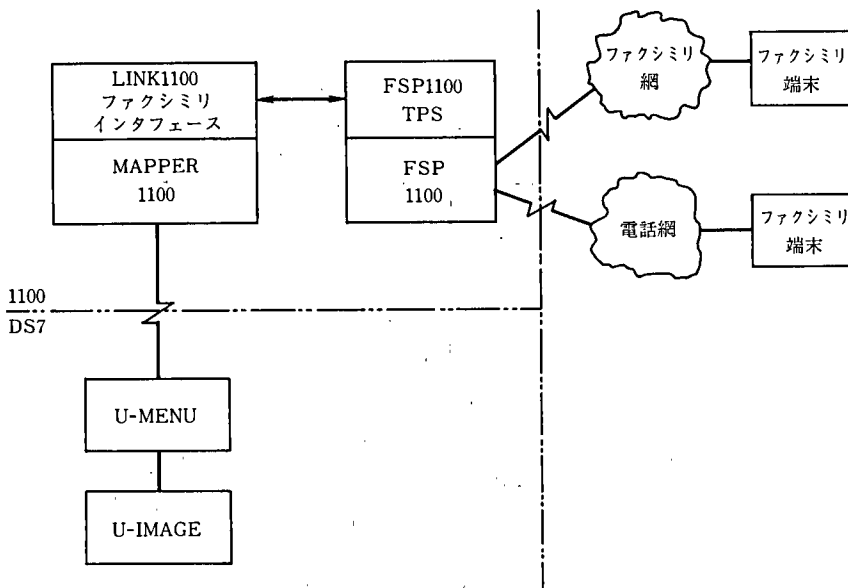


図 8 U-IMAGE とファクシミリ接続

Fig. 8 Communication between U-IMAGE and facsimile

両モードに対応しているため、U-IMAGE とファクシミリのデータ転送は効率良く行うことができる^[5]。

MAPPER 1100 とファクシミリの接続のためには、UNISYS シリーズ 1100 上に統合 OA ソフトウェア AOA/LINK 1100, およびファクシミリ支援プログラム FSP 1100 が必要である。FSP 1100 と MAPPER 1100 の間のデータ交換は、FSP 1100 のトランザクション・プロセスと AOA/LINK 1100 により行われる。

6. おわりに

ここで紹介した MML は、始めに述べた MML の実現レベルとしては、レベル 2.5 くらいであろう。イメージ・ファイルがマッパーと DS 7 で共有できる点、レベル 2 は達成している。一方、オペレータがマッパーを意識しなければならない点、まだレベル 3 に到達していない。しかし、この問題は U-VIEW/MAPPER^[6] や AOA によるメニュー選択を使うことにより、かなり解消できる。

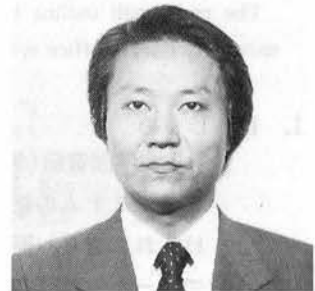
本稿は、分散処理実現のための一要素技術である MML を、DS 7 の U-IMAGE のケースを例として述べたものに過ぎない。DS 7 上でマイクロ・マッパー・リンクを活用している他の例としては、ファイル転送プログラム U-TRANS, 日本語ワープロ U-WORD/J, ビジネスグラフ作成プログラム U-GRAPH および統合 OA ソフトウェア AOA ソフトウェア・シリーズ等がある。

分散処理とは、単に機能と負荷を分散するだけでなく、それらを統合して協調処理することを要求するものである。分散処理の一部を成す MML も同様である。そう言う観点から、ホストとワークステーションの本当の意味の統合操作環境の構築がこれからの課題である。

-
- 参考文献 [1] 「分散処理技術」, 情報処理, 1987, Vol. 28, No. 4.
 [2] 「本格的な普及へ向かうマイクロ・メインフレーム・リンク」, 日経コンピュータ, 1987, 2, 16.
 [3] 「MML で機能分散を図る 4 GL と DBMS」, 日経コンピュータ, 1987, 11.9.
 [4] 佐々木茂, 「マイクロ・メインフレーム・リンク用統合操作環境」, 技報, 日本ユニパック(株), No. 13, 1987.
 [5] 河合昭男, 「イメージ・データの圧縮手法」, 技報, 日本ユニパック(株), No. 15, 1987.
 [6] 伊東春正, 「MAPPER におけるマンマシン・インタフェースの改良/直接操作の導入」, 技報, 日本ユニパック(株), No. 17, 1988.

執筆者紹介 河合 昭 男 (Akio Kawai)

昭和 22 年生。45 年大阪大学理学部数学科卒業 46 年日本ユニシス(株)(当時、日本ユニパック(株))入社、EXEC 保守、シリーズ 1100 性能評価、無人化システム開発を経て、現在ワークステーション・システム部に所属し、DS 7 のイメージ処理を担当。情報処理学会会員。



オフコンにおける大学図書館システムの実現

University Library System on a Small Business Computer

斎藤ひろみ

要約 大学図書館においては、冊数増加による事務量の増加と、新キャンパス開設による分館処理の機械化ニーズが顕在化している。中小規模大学図書館（蔵書数が数万冊から数十万冊）もその例外ではない。

汎用機での図書館システムの構築例は数多いが、中小規模図書館では予算的にみてオフコン・クラスのコンピュータによる機械化ニーズが圧倒的に多い。しかし、オフコンでの図書館システムの構築は、部分的に実現しているのが現状である。

オフコンによる機械化では、図書館業務をサブシステムに分割し、段階的なシステム化を行い、バーコード・リーダ、バーコード・プリンタ、蔵書点検用のハンディ・ターミナルといった新しい機器を導入して、操作性と運用性の優れたエンド・ユーザ指向のトータル・システムを構築すべきである。

本稿では、UNISYS シリーズ 8 における中小規模大学図書館システムの概要を述べたうえで、今後の大学図書館システムの方向について述べる。

Abstract University libraries have a problem of office work explosion caused by a rush of newly published books and newly opened branches in the campuses. Medium and small university libraries, which have a collection of books from 10 thousands to 100 thousands, are no exception to this phenomenon.

Although there are many library-office systems using general-purpose computers, medium and small libraries rather choose uses of small business computers by reason of budget.

There are some critical points in building "Total Library-Office System" by using small business computers. The first is division of library-office work into subsystems and stepwise systemization of them. The second is adoption of automatic equipments such as bar-code reader, bar-code printer, and hand-held portable computer to save the labor of library work.

The paper will outline the "Total Library-Office System" and finally point out the direction of university library-office system in future.

1. はじめに

J 大学図書館(筆者が開発に参加した神奈川大学、女子美術大学および東京家政大学の各システムの総称)は、13万冊を保有する開架式図書館である。年間に7,000冊を受け入れており、冊数増加による事務量の増大と、分館開設という業務への対応が必要であった。業務の合理化は、基本カード*、著者名・被伝者**名順のカードの整理や、

* 図書の情報に記載されているカード

** 伝記に書かれた人

各研究室の蔵書点検という観点からも見直しを図る必要があった。

一方、大学側は、入試・学籍・就職といった各分野にオフィス・コンピュータ*(以下オフコンと略す)を利用していることから、分散型統合システムを目指す必要上、既存機種と同じオフコン UNISYS シリーズ 8 に決定した。

筆者はこのオフコン上に、図書館システムをトータル・システムとして構築したが、その特長は概略次の通りである。

- 1) 予算システムから発注・受入、整理、蔵書点検、貸出システムまでを含む図書と雑誌に関する図書館の作業全般をカバーする
- 2) 各サブシステムに分け、段階的にシステム化を図れる
- 3) MARC**の導入やキーパンチの外注によって入力作業の負荷を軽減できる
- 4) バーコード・リーダーやハンディ・ターミナルといった機器を導入しており、操作性と運用性が高い
- 5) ユーザのニーズや業務をつかむための調査票から、システムの概略の仕様を設定できる。なお、この調査票はディスク容量の推定に使用されるだけでなく、ユーザにシステムの概要を説明するためにも使用される

本稿では、このシステムの概要を紹介するとともに、実際に図書館システムを構築し運用していく時の方法論、および考慮点を紹介している。

2. 図書館トータル・システム

本システムは、図書館業務を機能別に分類した9個のサブシステムから構成されて

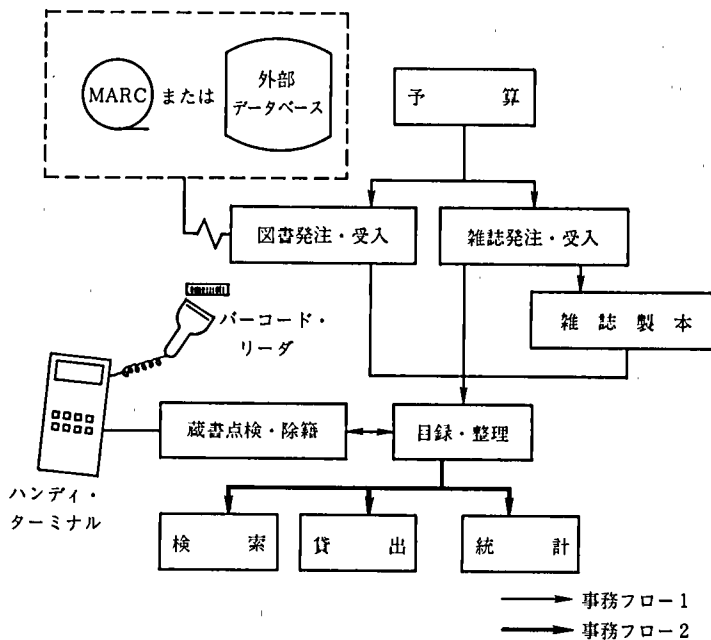


図1 図書館サブシステム関連図

Fig.1 Process flow of library-office system

* シリーズ 8

** MACHINE READABLE CATALOGING(機械可読目録)。 (3.1.1 MARC 参照)

おり、段階的なシステム化が可能である。1サブシステムは、4か月半でシステム化できる。これらのサブシステムは、図1の通りである。サブシステムは、本の収集と保管を支援する事務フロー1と、利用者への本の提供を支援する事務フロー2に大別される。

事務フロー1は、図書雑誌費を管理する予算サブシステムから始まり、図書の流れと、雑誌の流れに分かれる。つまり、図書発注・受入サブシステム、雑誌発注・受入サブシステムと雑誌製本サブシステムである。図書と製本済雑誌は、目録・整理サブシステムで基本カードに記載されるような情報が入力される。登録された情報は、蔵書点検・除籍サブシステムで保管管理される。

事務フロー1で管理されている情報は、事務フロー2の貸出サブシステム、検索サブシステム、統計サブシステムによって利用される。

以下、各サブシステムについて概要を紹介する。

2.1 予算サブシステム

図書館および研究室の図書費予算と、発注時・受入時の二つのタイミングでの数字をとらえ、随時残高を知ることができる。年度末の予算管理および次年度予算の参考資料にする。購入金額(定価ではなくサービス額)もとらえる場合がある。

2.2 図書発注・受入サブシステム

発注する図書を入力して、発注書を納入業者別に出力する。美術書等を収集している場合は、欠本*を見逃すと手に入らないことがある。そういう事態を防ぐため、欠本リストを納入業者別に年2回出力して書店へ渡し、廃刊、売り切れ、増刷中といった状況を調査させる。

本が納入された時点で受入の入力を行う。登録番号を自動発番し、資産として登録し台帳を出力する。また整理後、請求番号を付与してから台帳を一括出力することも可能である。あらかじめ出力してあるバーコード・シールを2か所に貼る(3.2 バーコード採用について参照)。

寄贈本や移管本については、別台帳で管理することも可能である。

2.3 雑誌発注・受入サブシステム

発注する雑誌の書名や出版社(出版社も含む)等を基本登録し、巻号数を所蔵登録していく。欠号については、図書よりも頻繁に欠号リストを出力しチェックする。

2.4 雑誌製本サブシステム

一年分、雑誌がまとまると製本して資産登録する。このサブシステムでは、製本の発注と受入を管理し製本済雑誌を図書として登録する。

2.5 目録・整理サブシステム

バーコード入力で、受入時の情報を呼び出し、整理した情報(基本カードの情報)を入力する。漢字入力が多く時間がかかるので、入力時間の短縮のためにMARCの導入とキーパンチの外注を行っている。(3.1 入力時間の短縮方法参照)。

分類番号は、一般的には表1のNDC**が使用されている。当該図書館における専門分野についてはNDCより詳細な、その図書館独自の分類を使用している場合が多い

* 全集等、継続して発刊されている本のうちで欠けた本

** Nippon Decimal Classification(日本十進分類法)

表1 日本進分類法

Table 1 Nippon decimal classification scheme

主 類 表 (MAIN CLASSES)		
0	総 記	General works
1	哲 学	Philosophy (哲学, 心理学, 倫理学, 宗教)
2	歴 史	History (歴史, 伝記, 地誌, 紀行)
3	社会科学	Social sciences (政治, 法律, 経済, 統計, 社会, 教育, 民俗, 軍事)
4	自然科学	Natural science (数学, 自然科学, 医学)
5	工 学	Technology (工学, 工業, 技術, 軍事)
6	産 業	Productive arts (農林, 水産, 商業, 交通)
7	芸 術	Fine arts (美術, 音楽, 演劇, 運動, 遊芸, 娯楽)
8	語 学	Language
9	文 学	Literature

ので、桁数・配列に注意が必要である。

増加図書目録を書名順に配列するのに、洋書の場合、冠詞種別コードを作成し、書名の頭の冠詞だけを取り出して配列した(図2)。

2.6 蔵書点検・除籍サブシステム

年に一度、本が紛失していないかどうかを点検作業をする。その際、本システムではハンディ・ターミナルを使用する。これを用いても点検のスピードは手作業と同じだが、基本カードを知らないアルバイトでも作業できるようになる。また、集計作業と報告書作りに2日から3日かかっていたのが、数時間でできるため事務の生産性が大幅に向上する。

不明がわかってから3年後には自動的に除籍する。また汚損や破損等による除籍は画面から人手で除籍する。

2.7 貸出サブシステム

館外に貸出した本の管理を行う。貸出時に学生証と本をバーコード入力する。学生証のバーコードは、学籍システムで毎年4月に発行して付けている。教職員と卒業生の貸出登録者には、バーコードの付いた貸出券を発行する。

返却時は、返却本をバーコード入力するだけである。予約本情報を入力しておく、返却時にプザーがなり、メッセージを表示する。督促状と統計表は定期的に出力される。

2.8 検索サブシステム

図書館員を対象とする図書・雑誌の検索を行う。学生や教職員等の利用者を対象とするシステムの場合は、端末機数とレスポンス面からの検討を詳細に行う必要があるため、今回はシステム化の対象としていない。

DSD134 NAME : DSM34MST 冠詞種別コード 9/08, TIME=13:08:31 PAGE 001

冠詞種別	冠詞種別	冠詞1 冠詞13	冠詞2 冠詞14	冠詞3 冠詞15	冠詞4 冠詞16	冠詞5 冠詞17	冠詞9	冠詞10	冠詞11	冠詞12
1	英語	A	AN	THE						
2	独語	DAS	DER	DIE	EIN					
3	仏語	L'	LA	LE	LES					
4	イタリア語	GL'	GLI	I	IL	UN	UN'	UNA	UND	
5	スペイン語	EL	LA	LAS	LO					
6	ポルトガル語	A	AS	O	OS					
7	オランダ語	DE	EEN	EENE	HET					
8	デンマーク語	DE	DEN	DET	EN					
9	ノルウェー語	DEI	DEN	DET	E					
10	スウェーデン語	DE	DEN	DET	EN					
11	ハンガリー語	A	AZ	EGY						
12	ルーマニア語	O	UN	UNA						
13	ウェールズ語	Y	YR							
14	アラビア語	AL-	EL-							
15	エスペラント語	LA								
16	特殊記号	.	r							

図2 冠詞種別コード

Fig.2 Table of articles

書名や著者名といった検索キーを入力すれば、ヒットした件数を表示する。表示内容は書名や出版者等の簡略画面と、価格や所在部所までを含む詳細画面から構成されている。

検索を高速化するため、検索専用ファイルを作成している。データの蓄積は、月1回、400~800件がバッチで追加される。

前方一致と後方一致の両方式の検索および、マスク文字(*, #, !)による検索が可能である。

中間一致はレスポンス・タイムが他より大きいため、検索途中でメッセージを出すようにしている。

2.9 統計サブシステム

各サブシステムで作成されなかった詳細な統計や、経営指針の参考となる統計表を作成する。

代表的なものに次の統計表がある。

- 貸出統計 登録者数一覧表 (登録率等)
- 貸出冊数一覧表 (ベスト100等)
- 所蔵統計 分類別*冊数・金額一覧表

* NDC(日本十進分類法)別の統計

分野別*冊数・金額一覧表

収集統計 年間受入冊数・金額一覧表

2.10 J大学におけるプログラム本数および工数

J大学図書館では、

- 1期 予算, 図書発注・受入, 目録・整理, 蔵書点検・除籍
- 2期 雑誌発注・受入, 雑誌製本, 貸出, 上記業務の1期分の見直し
- 3期 検索, 統計, 上記業務の1・2期分の見直し

の段階で、システム化を行った。

表2はJ大学のプログラム本数および工数を示すものである。なお、言語はPROGRESS-IIを用い、プログラムの平均ステップ数は230である。(COBOLの300ステップと同じ規模)。またハンディ・ターミナルで使用された言語はBASICであり、300~400ステップで、開発には約1人月かかっている。

図書館の仕事の大部分を網羅しているため、オフコン上での他システムと比較すると大規模システムとなった。

表2 J大学のプログラム本数および工数
Table 2 Number of programs and manpower at J university

サブシステム名	プログラム本数概算(本)	工数概算(人月)
予算	25	3~4
図書発注・受入	40	6~7
目録・整理	110	17~18
蔵書点検	10	1~2
初期マスタ登録	10	1~2
雑誌発注・受入	40	5~6
貸出	60	8~9
雑誌製本	30	4~5
検索	50	12~13
統計	30	4~5
合計	約300~400	約60~70

3. 本システム構築上の問題点と解決策

3.1 入力時間の短縮方法

整理した情報は漢字が多く入力に手間がかかった。操作員が不慣れな時の入力件数は、平均8件/時間である。そのため、MARCや外注パンチを採用し入力時間の短縮を図った。

3.1.1 MARC(Machine Readable Cataloging, 機械可読目録)

MARCによって、図書館のカードに記録されている書名や著者名等の目録情報が、磁気テープで提供される。

MARCには、表3の主なMARCの種類にもあるように和書にJAPAN MARCと市販MARC, 洋書にLC MARC等がある。

和書では、JAPAN MARCが標準だが、本が出版されてからMARCとして提供されるまでの期間が、3~6か月と長い。市販MARCは2~3週間位で届くので、市販

* 文部省によって定められた分野別の統計

表3 主なMARCの種類
Table 3 List of major MARC

和 書	JAPAN MARC 市販 MARC ・大阪屋 MARC ・TRC MARC ・日販 MARC	国立国会図書館 大阪屋 図書館流通センター 日本出版販売
	LC MARC (US MARC) UK MARC INTER MARC MAB 1	米国議会図書館 英国国立図書館 フランス国立図書館 西ドイツ国立図書館

MARCにあるデータは、まず市販MARCで運用してからJAPAN MARCに置き換えるのが望ましい。しかし予算の制限から市販MARCのテープを購入して、内容不足分は手で修正して使うようにした。MARCを使うと30件/時間で入力でき、約3倍の効率アップが図れた。

市販MARCは、業者から購入するのだが、どのMARCを使用するかが問題となる。MARCの値段は、本と同時に購入すると1件当たりの単価が安くなる。そのため、購入図書をより多く扱っている業者から購入するほうがよい。書店から出されている新刊案内からでは、どのMARCが最適なのか判断できないため図書館にある本から100冊程抜粋し、どのくらいMARCがあるか各社に調査を依頼した。選んだ本が専門性が強かったせいか、7割前後のヒット率であった。J大学では、一番ヒット率の高かったTRC MARCを使用することにした。

3.1.2 キーパンチの外注

J大学図書館では、すでに13万冊の蔵書がある。その情報を入力するのに、キーパンチを外注した。そして、必要最低限の項目をまず5年間で入力し、その後、順次使う項目を入力していく予定である。

入力原稿は基本カードを使用した。基本カードは属人性が強く、チェックに手間がかかった。たとえば著者名で、区切り記号が違う場合(“ラフカディオ・ハーン”と“ラフカディオ、ハーン”)、姓名の順序が違う場合(“ラフカディオ、ハーン”と“ハーン、ラフカディオ”)、本名とペンネーム等が違う場合(“ラフカディオ・ハーン”と“小泉八雲”)とさまざまである。図書館内で規則を決めているが、カードを整理した人によって多少の違いがある。

この問題は、MARCからの入力情報も同様である。1文字の相違でヒットしなくなるため、著者典拠や書名典拠等の辞書や、同義語や同意語の定義の機能を持たせ、機械で支援していく方法が望まれる。

3.1.3 整理伝票

整理した情報を入力する場合、当初は調べながら入力したため、入力操作に不慣れなこともあって2~3件/時間という結果となった。

そこで、入力時間の短縮のため整理伝票を起票してから、まとめて入力するようにしたところ、平均8件/時間となった。その後、操作員が慣れるに従って、15件/時間まで向上した。

表4 OCRとバーコードの比較表
Table 4 Comparison of OCR with bar-code

	OCR	バーコード	
		ペン型	ワンタッチ・スキャナ型
文字種	OCR-B 数字 アルファベットは限定される。 サインは限定される。	NW 7 数字 アルファベットは限定される。 サインは限定される。	同左
汚れ	ラミネートはかけること 文字のそばが汚れた場合正確に 読めない。	ラミネートはかけること 部分的に汚れていても汚れてい ない部分で読むことができる。	同左
読み取り エラー	読み取りエラーの場合読めるま で手でこすらなければならない または手入力	同左	自動的に読める部分 を探すためエラーは ほとんどない。
ラベル	現在機器構成のシリアルプリン タでプリント可能	専用のラベルプリンタを増設 または外注する。	同左
ハンディ・ ターミナル の容量 (1冊を 40Bで換算)	OCRのハンディ・ターミナルに は手頃な商品がない。	① 16 KB (約 200冊) ② 24 KB (約 400冊) ③ 32 KB (約 600冊) ④ 64 KB (約 1,000冊) ⑤ 128 KB (約 2,400冊) ⑥ 192 KB (約 4,000冊) ⑦ 256 KB (約 6,000冊)	① 64 KB (約 1,000冊) ② 128 KB (約 2,400冊) ③ 192 KB (約 4,000冊) ④ 256 KB (約 6,000冊)

3.2 バーコード採用

3.2.1 バーコードとOCRの比較

バーコードかOCRかの決定は、以下の二つの理由からバーコードが採用された。

- 1) OCRは1か所汚れただけで読めなくなる可能性が高いが、バーコードは横に読める場所が1か所あればよいから、バーコードの方が汚れに強い。
- 2) OCR用の手頃なハンディ・ターミナルがない。バーコードは、片手で持てるサイズだが、OCRは肩から下げて持つ大きさで重く値段も高い(表4)。

3.2.2 読み取り装置

読み取り装置は、読み取りエラー率の少ないワンタッチ・スキャナ型を採用した。ペン型だと操作員の走査の速さが不適切だと読めないこともありうるが、ワンタッチ・スキャナ型は走査線が自動的に走るのだから接触だけで読める。

3.2.3 ハンディ・ターミナル

ワンタッチ・スキャナの接続できるハンディ・ターミナルということで、エプソン製HC10を採用した。このシステムの最大のRAMサイズは、256KBで約6,000冊入力できる。

研究室で蔵書点検をしている際、他の研究室の本が混ざっていることがある。現状では、それがハンディ・ターミナルからシリーズ8に送り、不明リストを出力して始めて発見される。もし13万冊の蔵書をRAM内に持てれば点検中に発見できるようになり、その場で正しい部所へ本を戻すことが可能となる。これは今後の課題である。

3.2.4 バーコード・プリンタ

バーコード・プリンタは、シリーズ8で実績のあったアンリツ製K464Eを採用した。印字品質を保つためリボンとシールの紙質を検討し、極力汚れない材質にした。印字速度が3枚/分と遅いので、500枚くらいを単位として出力する運用とした。13万冊の本に貼る分はレイアウトを変えて、右側に漢字書名や請求番号を出力した(図3)。

3.2.5 ラミネート

何十年と保存するため、バーコードには必ずラミネートをかけることにした。布等の表紙にも貼るため取れないように、また何十年たっても変色しないのりを使っているドイツ製フィルムルックス(Filmolux)を選択した。

3.2.6 バーコード貼付

バーコードシールは、1冊に2か所貼るようにした(図4)。

1か所は、内表紙に蔵書印がわりに貼った。本来は蔵書印を押した後、登録番号をナ

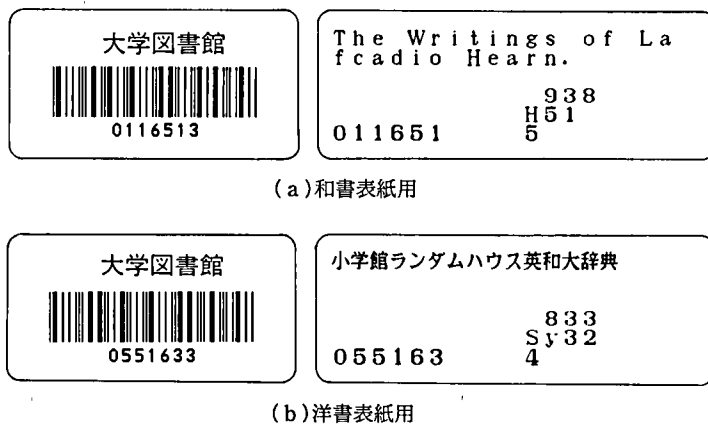


図3 バーコード・シールのレイアウト

Fig. 3 Layout of bar-code seal

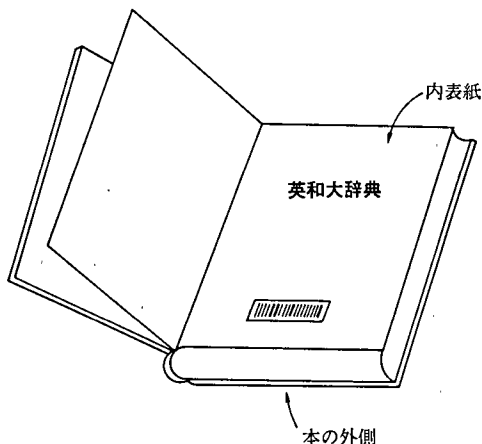


図4 バーコード貼付について

Fig. 4 Affixing a bar-code seal to a book

ンバリングで押していたのをバーコード・シールを貼るだけにした。

もう1か所は本の外側に貼った。貸出や蔵書点検時に読めない場合は、内表紙のバーコードを読むか、手で入力する。下地が黄色だと乱反射しやすく読みにくい、左側をあわせると正確に読むことができた。なお、下地に文字があると、すけて読み取りエラーを起こしやすいので、なるべく白いところに貼るようにしている。

カバーやハードカバーがあると読み取り時に手間がかかるので、貼るときに破棄することにした。カバー付きの方がきれいで利用者の興味をそそりやすいが、公共図書館のようにカバーの上からラミネートをかけると、ラミネート代がかさむためである。

13万冊の本にバーコード・シールを貼る作業は、アルバイトを採用した。1人平均100冊/時間程度の作業であるが、本が雑然と放置されている研究室等では、本を捜すのに時間がかかるため80冊/時間となった。

3.3 図書館員の機械化への対応

図書館員は、目録規則を頼りに情報を整理する。目録規則では、本の情報をできるだけ基本カードに盛り込むように定められている。そのため、機械化するにあたって同じように考えがちである。

このことによるディスク容量の増大や、システムの肥大化といった問題を避けるために本の書名を正確に知ることも重要だが、利用者の目指す資料をできる限り早く探すことの方が、より重要であるということを強調した。また漢字の入力が多いので、入力時間の短縮というのもその理由であった。

システム化の概要を決定する時には、目録規則と入力時間を考え合わせながら必要項目を決め、一部の入力項目は後送りした。

外字に関しても同様である。目録規則にあるように本をそのまま機械に入力すると、外字の数が膨大になる。それを避けるために、なるべく翻字してもらいJIS第1,第2水準に納まるようにした。

3.4 カードレスをめざして

図書館員にとって、基本カードはツールのすべてである。しかし、その保守に非常に時間を要する。機械化によって、カードをなくそうというカードレスの発想がある。カードレスによって図書館員の手間はかからなくなるが、利用者へのサービスが低下することは避けなければならない。

そこでJ大学図書館では、新着図書ニュースを月1~2回、冊子体目録(図書目録)を年1回作成し、製本して利用者に提供することにした。また5年に1回は、冊子体目録を5年分まとめて作成することにした。この目録を館内3か所に配置するため、プリント量が20,000ページになる。

そこで高速ライン・プリンタ(240 LPM)を使用して、時期的には学籍や就職等の他システムが比較的混み合わない5月に1週間くらいかけて出力することにした。

3.5 本システム設計上の工夫

13万件という大量データのため、ファイルを3分割して持つようにした。項目の分割法は次のようにした。たとえば著者は、共著者が多い本に合わせるとディスク容量が増大する。

そこで、図5で示す簡易分割方式をとった。著者を2分割し、必ず使うファイルに1名分、2番目に4名分、一番使わないファイルに5名以上の場合に、あふれフラグを

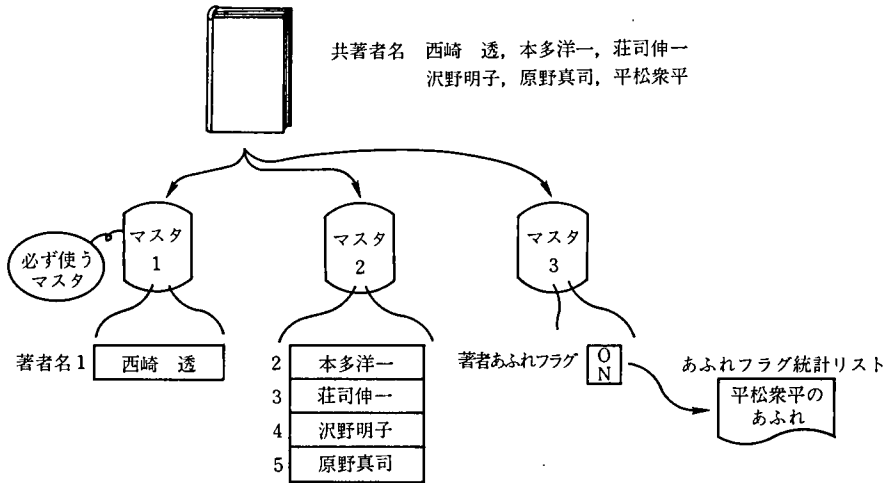


図5 簡易分割方式

Fig. 5 Method of simple partition system

表5 外字表

Table 5 Table of extra-standard character set

JIS 区点	UNISYS コード	JIS 区点	UNISYS コード	JIS 区点	UNISYS コード	JIS 区点	UNISYS コード
1 8801	F8A1 Å	2 8802	F8A2 Ä	3 8803	F8A3 Å	4 8804	F8A4 Å
5 8805	F8A5 Å	6 8806	F8A6 Å	7 8807	F8A7 Å	8 8808	F8A8 Å
9 8809	F8A9 Å	10 8810	F8AA Å	11 8811	F8AB Å	12 8812	F8AC Å
13 8813	F8AD Å	14 8814	F8AE Å	15 8815	F8AF Å	16 8816	F8B0 Å
17 8817	F8B1 Å	18 8818	F8B2 Å	19 8819	F8B3 Å	20 8820	F8B4 Å
21 8821	F8B5 Å	22 8822	F8B6 Å	219 9137	FBC5 Z	320 9138	FBB8 Z
321 9139	FBR9 Z	26 8826	F8B8 Å	323 9141	FBC9 Z	324 9142	FBCA Z
325 9143	FBCB Z	326 9144	FBCD Z	327 9145	FBCD Z	328 9146	FBCD Z
329 9147	FBCF Z	330 9148	FBD0 Z	331 9149	FBD1 Z	332 9150	FBD2 Z
333 9151	FBD3 Z	334 9152	FBD4 Z	335 9153	FBD5 Z	336 9154	FBD6 Z
337 9155	FBD7 Z	338 9156	FBD8 Z	339 9157	FBD9 Z	340 9158	FBD9 Z
341 9159	FBD8 Z	342 9160	FBD9 Z	343 9161	FBD9 Z	344 9162	FBD9 Z
345 9163	FBD9 Z	346 9164	FBD9 Z	347 9165	FBD9 Z	348 9166	FBD9 Z
349 9167	FBD9 Z	350 9168	FBD9 Z	351 9169	FBD9 Z	352 9170	FBD9 Z

付けることにした。最大5名が妥当かどうか今後検討するために、あふれフラグの付いたデータの統計もとっている。なお、書名表記や件名表記等も同様に処理している。

EBCDIC というコード体系では、英小文字とカタカナの同居が許されない。そのため、洋書は漢字コード(2バイト系)を使っている。ドイツ語の ð, フランス語の é は、ユーザ登録外字を使用して対応している。シリーズ8では、940文字までユーザ登録外字対応が可能であるが、他機種とのやり取りやディスク容量等の面で今後の課題が残されている(表5)。

4. 現状の図書館システム化における本システムの位置付け

図書館システムと一口にいても、トータル化されたシステムはオフコンでは極めて少ない。たとえば、図書目録サブシステムだけを図書館システムと呼ぶ例が多い。

また、トータル・システムと言っても、検索サブシステムの評価は高いが予算システムがない場合とか、国立大学で開発されたシステムを元にしていて私立大学の予算サブシステムがない場合とか、発注・受入サブシステムの入力データが目録・整理サブシステムでは使えず再入力となる場合とかがある。

以上から考えてみると、本稿のシステムはトータル・システムと言えよう。なお、このシステムは、ハードの制限上から中小規模の図書館(蔵書数が数万から数十万冊)に最も適しており、企業内の図書館でも使用できる。

5. 今後の課題

今後の課題としては、まず今まで述べてきたように、洋書の漢字コード対応、ハンディ・ターミナルのRAMの大型化がある。

それから、発注サブシステムの前に選書サブシステムを作成すれば、よりトータルなシステムが実現される(図6)。

選書サブシステムでは、全出版物の市販 MARC を購入し、文部省推薦等の基準で選定候補リストを出力し、そのリストを選定委員会の参考資料とすれば、購入が決定された本のデータを発注サブシステムへつなげられる。

また検索でのヒット率を高めるためには、検索キーの付け方を質のよいものにしなければならない。書名や著者名以外に、本の内容を表す件名の付け方が質を左右する。著者名・個人名・件名を付ける場合、従来の著者典拠を機械上にもち、機械でチェックできるようになれば現在の整理時間が大幅に減るであろう。KWIC*やソーラス(Thesaurus)**の作成も必要となる。一般件名を付ける場合、自然言語で表現しやすいように同義語・同意語定義の機能をもつことが望ましい。

また情報検索では、文献の検索以外に、情報そのものを直接検索したいという要望が利用者から出てきている。本稿の UNISYS シリーズ8での検索システムだけではなく、このシステムから他のデータベースや検索用パソコンを利用できるようにするなど、利用者のほしい情報を提供していく方向が望ましい。

大学図書館相互のネットワークの場として、学術情報センターがある。ここは各大

* H.P.Luhn が提唱したキー・ワード順の文献索引

** 検索のための用語辞典

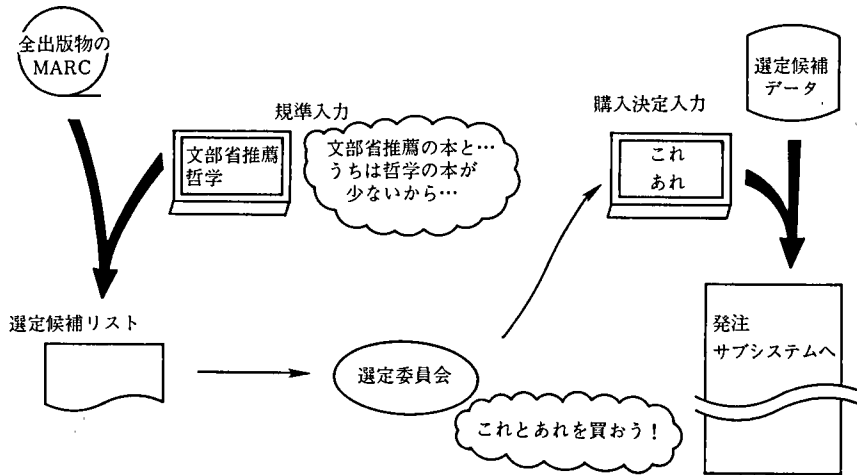


図 6 選書サブシステム

Fig. 6 Sub system for support of book selection

学図書館から集めた情報を一括管理し、使いたい図書館には提供するという機能である。そこで学術情報センターとつないで他図書館の本を検索し、ヒットしたデータを取り込むサブシステムを作れば入力時間の短縮となる。また学術情報センターでは、図書館同士の相互貸借システムの構想がある。UNISYS シリーズ 8 上で、学術情報センターの同システムと接続する相互貸借サブシステムを作成していけば、ネットワーク作りに有効である。

各研究室にパソコンを設置し LAN を経由して、発注・検索・貸出といったサービスをすると同時に、各研究室の予算管理もしていくようなシステムも作成中である。このほか、さまざまな情報媒体の変化への対応として、CD-ROM をはじめ、光ディスク、ビデオテックス等の画像情報装置や音声入出力機器との連携も必要となってきた。

必要な情報が手にはいりやすいからという理由でいままでは、蔵書数の多い大きい図書館の方が、サービスがいいと考えられてきた。しかし、外部の図書館や学術情報センターと接続し必要な情報の検索および、相互貸借の制度を充実させていけば、蔵書数の多さとサービスは関係がなくなってくる。

今後は、その図書館独自の専門性の高い図書収集に力を入れるとともに、図書館相互のネットワークを強化し、利用者の必要な情報を提供する図書館が理想の姿となっていくであろう。

- 参考文献 [1] 山下英男監修, 日本ユニバック総合研究所編著, 総合コンピュータ辞典, 日本経営出版会, 1970, p.433.
 [2] 黒沢正彦, 西村徹編, 「マークをうまく使うには」, 三洋出版貿易(株), 1986, pp.25~32.
 [3] 戸田光昭編, 情報ネットワーク論, 講座情報と図書館, 第6巻, 雄山閣出版, 1985.

執筆者紹介 齋藤ひろみ(Hiromi Saitou)

昭和34年生。56年立教大学文学部ドイツ文学科卒業(図書館司書取得)。同年日本ユニシス(株)(当時日本ユニパック(株))入社。主としてシリーズ8大学担当として客先サービスに従事。現在、日本ユニシス情報システム(株)に出向中。OAシステム統括2部システム8部に所属。



CAIシステム LEARN UP における オーサリングの特徴と今後の課題

The Authoring Function in CAI System "LEARN UP" and Its Future Enhancements

出 石 修

要 約 現在, CAI市場は活況の様を呈している。しかも, 昭和61年から文部・通産両省が CAI 利用に積極的な姿勢を示したことにより, 今後ますます発展しそうな状況にある。

このような状況下にあつては, CAIシステムは一段と厳しく吟味されることになる。とくに, コースウェアを作るためのオーサリング機能の充実は重要視される。

日本ユニシス(当時, 日本ユニバック)は, 昭和57年にオーサリング機能を備えた CAI システム「LEARN UP」を発表しているが, 以上のような理由から, さらにその能力を増強すべく昨年, 新 LEARN UP システムの開発を行った。

本稿では, そのオーサリング機能において, 以下を満足させるために講じた方法について述べている。

- 1) 簡単な操作でコースウェアを作成・加工・編集できること
- 2) しかも, 質の良いコースウェアを作れること
- 3) でき上がったコースウェアは移植性が良いこと

さらに, 今後必要になると思えるオーサリング機能についても言及した。

Abstract The CAI (Computer Assisted Instruction) market in Japan is active and strong today, and it will tend to grow rapidly even more in the coming years.

Its driving force comes from the governmental commitments to promote CAI systems, given by both the Ministry of Education and the Ministry of International Trade and Industry since 1986.

The current key issue in CAI system is lack of adequate authoring facilities, which prevents the spread of CAI systems.

The "LEARN UP" is a CAI system announced by Nihon Unisys, Ltd. (formerly Nippon Univac Kaisha, Ltd.) in 1982, which has an authoring facility.

Recently we have developed a new version of the "LEARN UP" system which has the much improved authoring capabilities.

This paper describes some design considerations on these capabilities as follows:

- 1) to improve its man-machine interface in courseware development.
- 2) to produce a high quality courseware.
- 3) to enhance the courseware portability to various systems.

Finally the future enhancements of this authoring capability will be also discussed.

1. はじめに

21世紀に向けた教育改革として, 昭和62年に臨時教育審議会が挙げたものは, 「個性の尊重」, 「生涯教育」, 「時代や社会の変化への対応」である。また教育課程審議会

が具体的手段として提言したのが、「情報活用能力の育成」、「教育施設の情報化」、「コンピュータ・グラフィックスなど発達した情報手段の教育への活用」である。

すでに、昭和 61 年には通産省と文部省により(財)コンピュータ教育開発センター(CEC)が設立され、68 年からのコンピュータを利用した学校教育の実施に向けて、パーソナル・コンピュータ(以下パソコン)のハードウェアやオペレーティング・システムなどの仕様が検討されている。さらに、通産省が情報処理技術者の不足を補う目的で、昭和 61 年から 65 年の予定で CAROL(Computer Aided Revolution On Learning)計画*をスタートさせている等、CAI(Computer Assisted Instruction)のニーズは高まっている。

CAIの内容を決定するのはコースウェア**である。そして、いかにしてコースウェアを作るかが問題となる。日本ユニシスは、昭和 57 年、他社に先がけオーサリング(コースウェア作成支援)機能を備えたパソコンによる CAI システム「LEARN UP」を発表した¹⁾。それまで、コースウェアは BASIC などの汎用プログラミング言語や、CAI 言語と呼ばれる専用プログラミング言語で作っていたが、LEARN UP のオーサリング・システムは、それらとは異なるものであった。それは、パラメータを与えコースウェアを生成するジェネレータ方式で、「先生自身が、自らの考え方にもとづいたコースウェアを手軽に作れるように支援すること」を目的とするものであった。

今回、「LEARN UP」の使用経験を踏まえて、さらに使い勝手が良く、良質のコースウェアを作ることのできるオーサリング・システムを目標に、新 LEARN UP オーサリング・システムの開発を行った。

本稿は、そこで考慮したことを説明するとともに、今後の課題について私見を述べたものである。

なお、本稿で扱っているオーサリング・システムは、LEARN UP システムのソフトウェア・コンポーネントであり、従来のものを「LEARN UP A 版ソフトウェア」、今回のものを「LEARN UP B 版ソフトウェア」として区別している。A 版と B 版との間でコースウェアの互換性はない(図 1)。

また、現在 B 版をもとにソフトウェア自体の移植性をも考慮した LEARN UP C 版ソフトウェアを開発中である。

2. 新オーサリング・システムの特徴

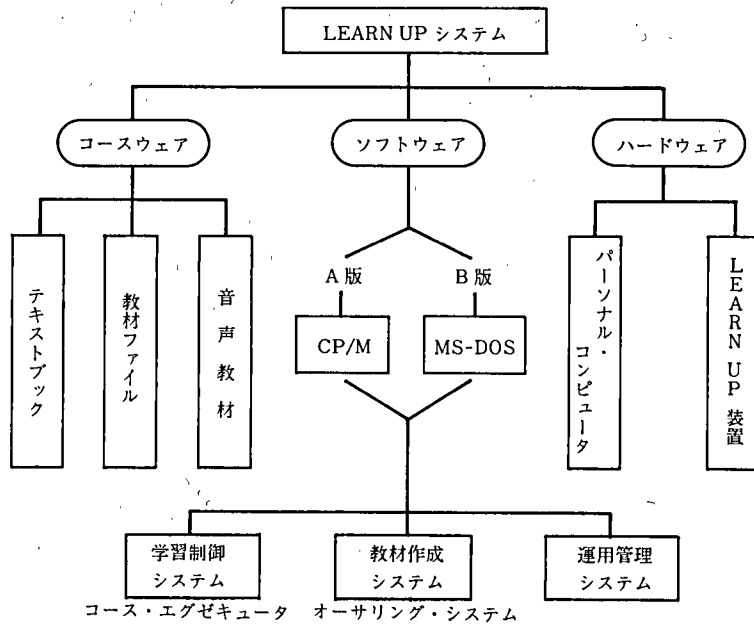
新オーサリング・システムは、生成するコースウェアについて、さまざまな考慮をしているが、その中の代表的なものを以下に示す。

2.1 移植性の向上

CAIの中核がコースウェアにあることはすでに述べた。そして、一般的に良いコースウェアほど、その制作に多大な労力・時間を費すものである。そうしたコースウェアは、さまざまなパソコンで使えることが望ましい。

* 通産省の外郭団体である中央情報教育研究所(CAIT)と情報処理振興事業協会(IPA)が、通産省の情報大学校事業構想の一環として、そこで使用される情報処理技術習得用コースウェア(CAROL コースウェア)を、企業に委託して揃えようとするもの。

** Courseware. CAI で使用される学習教材のことで、教授内容と教授方略からなる。



パーソナル・コンピュータ：当社のパソコンで LEARN UP 仕様のものを、とくに computent と呼んでいる。LEARN UP 仕様とは、メモリ 512KB (A 版は 256 KB)、ディスク装置×2' RS232C×1, CRT 解像度 640×400, カラー 8 色, 漢字 JIS 第一水準搭載をすべて満足するものを言う。

LEARN UP 装置：音声教材作成/再生装置には、音声教材のメディアごとに種々があるが、総称して LEARN UP 装置と呼んでいる。

図 1 LEARN UP システム基本構成

Fig. 1 System structure of LEARN UP

今回、次の手段を講じてコースウェアの移植性向上を図っている。

1) コースウェアのデータファイル化……プログラミング言語は機種ごとに異なるため、プログラム言語に一切依存しないように、コースウェアを——命令をすべてデータコード化した——データファイルとして生成している。なお、学習単位から見たコースウェアの階層構造とデータファイルの関係は、図 2 のようになっている。

① コース

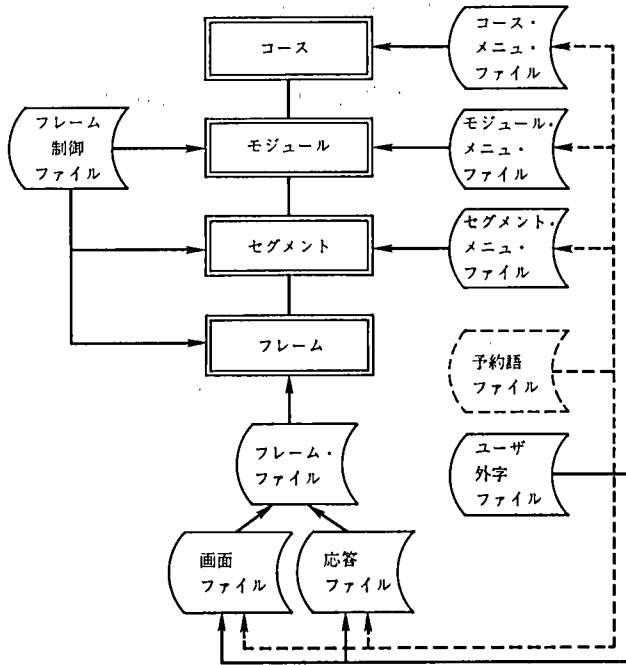
コースウェアの最大選択単位。1枚のコース・エグゼキュータ用ディスクに最大 10 コースまで登録できる。

② モジュール

単元 (通常 30~50 分の学習時間) の選択単位。基本的には、教材ディスク 1 枚, 音声教材 (カセットテープ) 1 巻が 1 モジュールである。1 コース当たり最大 10 モジュールまで登録できる。

③ セグメント

コースウェアの最小選択単位。通常 3~5 分の学習時間。



予約語ファイル：点線の矢印が示すファイルを作るときに使用される。

図 2 コースウェアの構造とファイルの関係

Fig. 2 Courseware hierarchy and data files

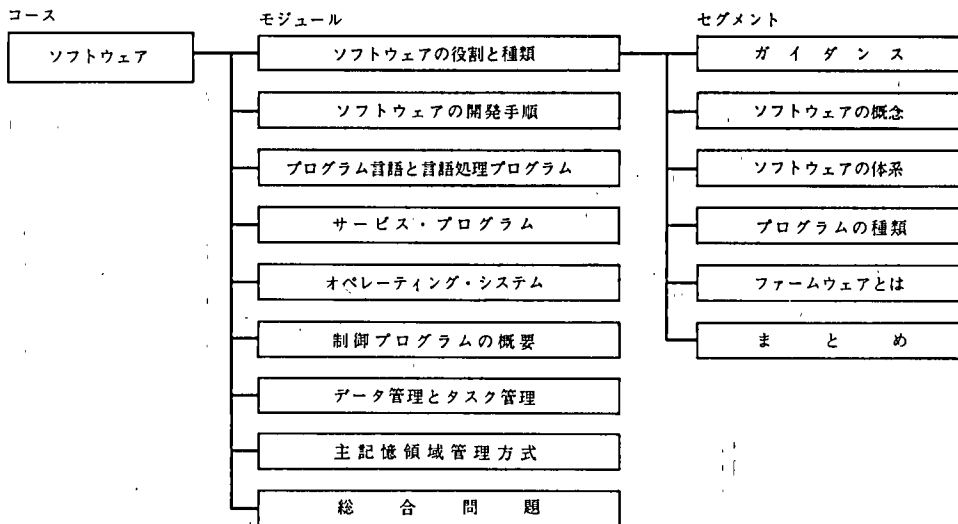


図 3 コースウェアの構造例

Fig. 3 The example of courseware hierarchy

1 モジュール当たり最大 10 セグメントまで登録できる。

④ フレーム

学習を進める上での基本単位で 1 教授項目（説明・問題・解説の組み合わせ）を内容とする。

1 セグメント当たり最大 10 フレームまで登録できる。

2) グラフィック・ドライバの採用……1)だけでは対処しきれないものに、ハードウェアが異なることからくるグラフィックスと外字コードの違いがある。そこで、この部分の処理ルーチンを機種ごとに用意することで、コースウェアの互換性を保証しようと考えた。この機種に依存する部分を処理するルーチンがグラフィック・ドライバである。グラフィック・ドライバは、図 4 に示す正規化された論理座標——画面ファイルのレコードは、コード化されたグラフィック命令と論理座標から作られている——を実際の物理座標に変換し表示するルーチンである。

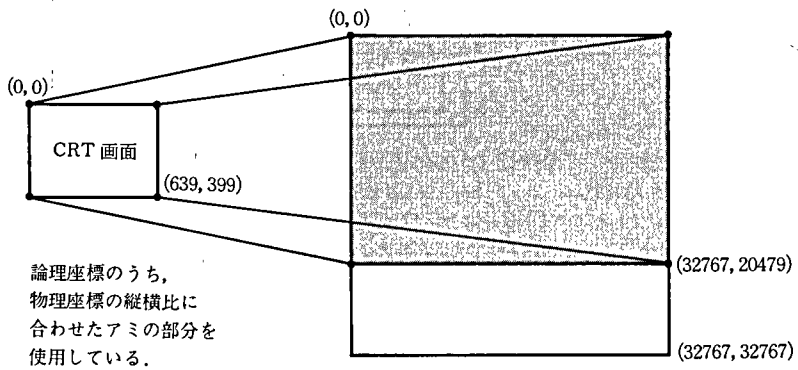


図 4 論理座標と物理座標

Fig. 4 Logical co-ordinates vs physical co-ordinates

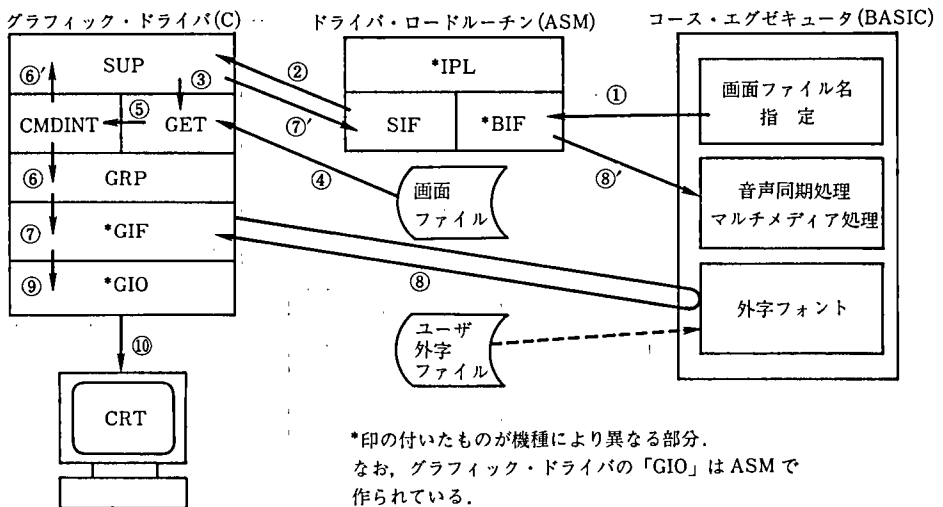


図 5 グラフィック・ドライバによる描画プロセス

Fig. 5 Painting process of graphic driver system

グラフィック・ドライバは、図5に示す構造をしており、ドライバ・ロードルーチン*によって読み込まれ、以後メモリに常駐する。

2.2 加工・編集の容易化

出来合いのコースウェアを一部変えて使いたいと考えている利用者は多い。そのため、以下を考慮し既存のコースウェアを容易に加工・編集できるようにした。

- 1) コースウェアの部品化……コースウェアのすべてが、個々の部品から構成されるようにした。したがって、コースウェアの修正は、それぞれ関係する部品に対して行えばよいし、新たに部品を作ってコースウェアを拡張することも可能である。なお、大量のコースウェア制作の際、部品を間違いなく操作・管理する手段として、ワープロにおける文書管理の手法を採り入れている(図6)。

図5におけるソフトウェア・コンポーネントは次のとおり。

IPL : グラフィック・ドライバをロードする。

BIF : コース・エグゼキュータとのインタフェースをとる。

SIF : グラフィック・ドライバとのインタフェースをとる。

SUP : スーパーバイザの役割を担う。

GET : 画面ファイルからレコードを読み込む。

CMDINT : コマンドを解釈する。音声同期やマルチメディアに関するコマンドのときは、一旦制御をコース・エグゼキュータに戻す。当該処理が行われた後は、再び制御が戻されるので、画面ファイルの続きを処理する。

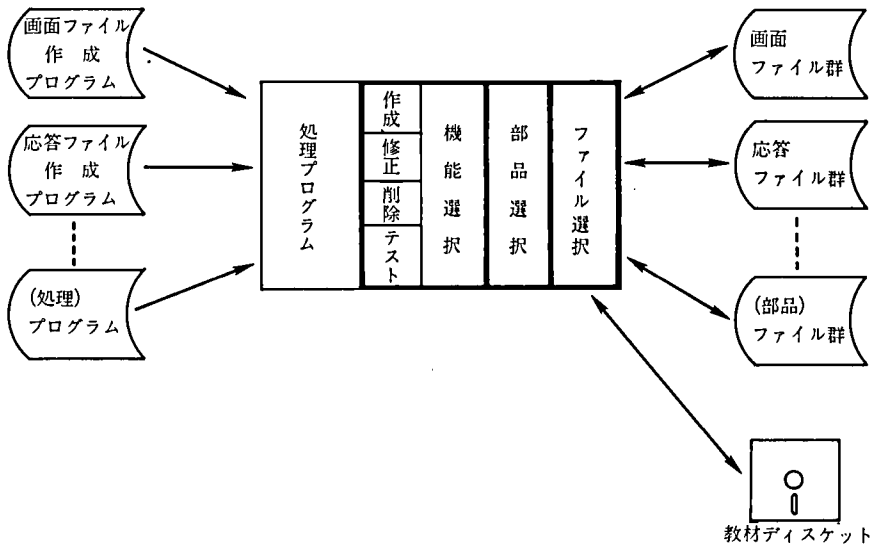


図 6 部品操作・管理の機構

Fig.6 Mechanism for management and manipulation of courseware

* ドライバ・ロードルーチンは、コース・エグゼキュータの初期化ルーチンから呼び出され、以後メモリに常駐する。このルーチンは、グラフィック・ドライバも含めてCルーチンを最大10個まで制御できるようになっており、現在、応答処理のマッチング・ルーチンなど計3個を制御している。ドライバ・ロードルーチンは、BASICからCを直接呼べないために準備されたもので、コース・エグゼキュータをCで書けば不要となる。

GRP：コマンド別の処理に区分けする。

GIF：物理座標への変換を行う。外字コードを見つけたときは、外字フォントの該当パターンに置き換える。

GIO：物理画面に描画する。

- 2) 画面作成の簡便化……コースウェアの開発において画面を作る比率は、他の部品のそれと較べて一段と高い。LEARN UP は、音声付きのシステムであり、ナレーションと同期をとりながら画面を動的に変化させることができるようになっている。したがって、効果のある画面を作ろうとすると、やはり試行錯誤的にならざるを得ない。そのため、画面は作りながら修正・確認できるようにし、さらに個々の画面構成要素についても、描画ではマウスを使ってのラバーバンド機能*を採用し、テキストでは、文字の追加・削除・書き換えといった基本機能に加えて、予約語の取り込み・センタリング・右詰め・置換・移動・コピーといったワープロ的機能を採り入れている。
- 3) 音声教材の修正・編集……従来、音声教材（カセットテープ）は、音声教材作成装置に付いているボタンを使って、すべて手操作で作っていた。新オーサリング・システムでは、コースウェアの作りやすさ——新規作成のほか、修正や変更が容易であること——を最重視している。そこで現在、新しい音声教材作成装置——ダブルカセットを内蔵したパソコンからの命令で読み書きができ、修正や編集が容易な装置——の開発を進めている。

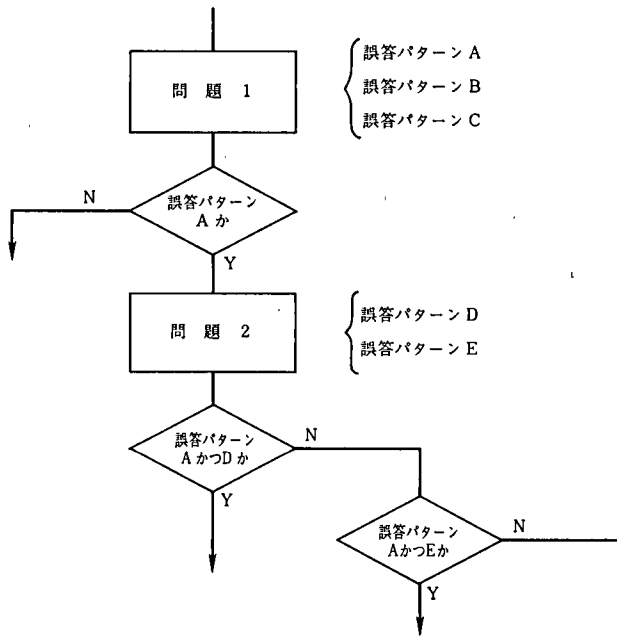


図 7 誤答パターンによるコース分岐

Fig. 7 Branching structure for adaptive learning

* 一箇所を固定されたゴムの端を持って、長さや方向を調整しながら位置合わせを行い、作図する機能を言う。

2.3 個人差への対応

コースウェアの良し悪しの判定基準はいろいろあるが、その第1は個人差にどれほど対応できているかであり、次の3点をもって対処している。

- 1) 応答履歴によるコース分岐……1モジュールの範囲内で、過去の応答内容をもとに適切なコース分岐(次に呼び出すフレームの決定)が行えるようになっている。仕組みとしては、モジュール内に設ける問題それぞれについて、その誤り原因を同定できる予想解答を用意し、それに誤答パターン記号を与える。この誤答パターン記号は、次の問題の予想解答のところに指定できるので、今までの経過をも含めた分岐の判断が行える、というものである(図7)。

ただし、コースの分岐は同一セグメント内でのみ有効としている。これは、LEARN UP のコースウェア構造が、1セグメントで通常一つのトピックを取り扱い、そのトピックを完全に理解させるために複数のフレームを通じて学習させるようになっていることによる。

- 2) 応答内容分析の柔軟性……誤り原因を同定するためには、個々の応答内容をきめ細かく分析できることが要求される。また、従来の限られたパターンによる予想解答の設定方式では、問題形式自体を限定してしまうことになり、真の評価を行うことはできない。そこで、オーサ(コースウェアの作成者)が予想解答のパターンを自由に作ってゆける方式*を採り入れ、分析の柔軟性を増すようにした。この方式により、これまで不向きであった記述式の問題への対応が可能となった。

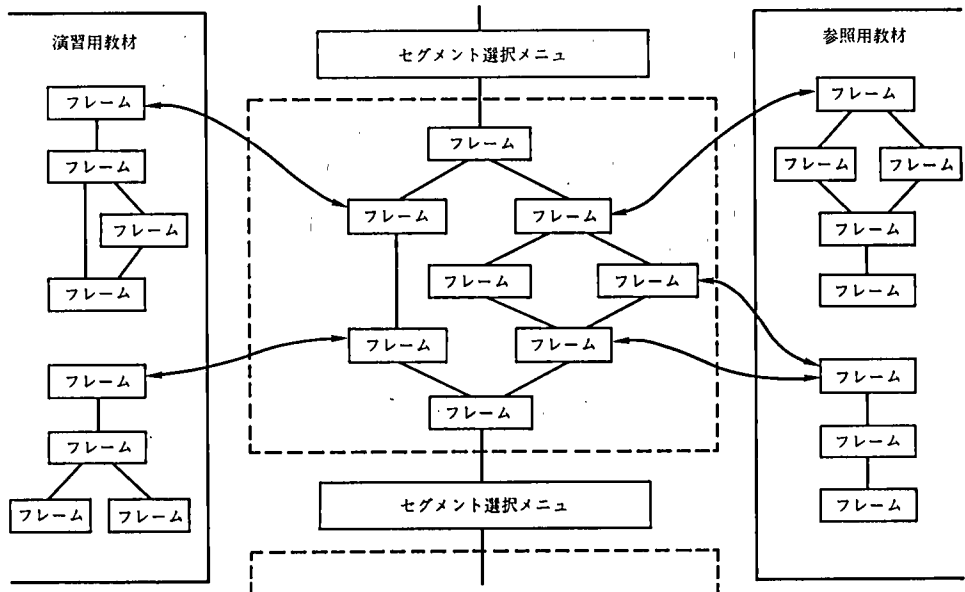


図8 補助教材による個人差対応

Fig. 8 Auxiliary sequence for individualized instruction

* たとえば、 $COMPUTE A = B * (C-D)$ のように COBOL の COMPUTE 命令では、算術式の演算子の前後には空白が一つ以上なければならず、演算順序を変える括弧と項目の間には空白はあってもなくてもよい。このような条件を一つのパターンとして定義することができる。

3) 補助教材での対応……一つのフレームから、最大10フレームまでからなる参照用教材や演習用教材が呼び出せるようになっている。1)の分岐を絡めて利用すれば、なお一層の個人差への対応が可能となる(図8)。

2.4 正当性の保証

コースウェアは、個々の部品の集合体であるとの考え方をとっている。そこで、個々の部品ごとの検証プログラム、および個々の部品とコースウェアの全体構造との整合性を調べるプログラムを用意することにより、集合体であるコースウェアの正当性を保証できるようにした。この関係を図示すると図9のようになる。

2.5 待ち時間の短縮化

学習時に学習者が感じる不満の一つとして、パソコンに待たされることが挙げられている。この待ち時間——待たされていると感じている時間——を以下のようにして短縮している。

1) コースウェアがすべてデータファイル化されているので、ディスクからまとめて読み込み実行する代わりに、レコード単位で1件ずつ読み込んで実行す

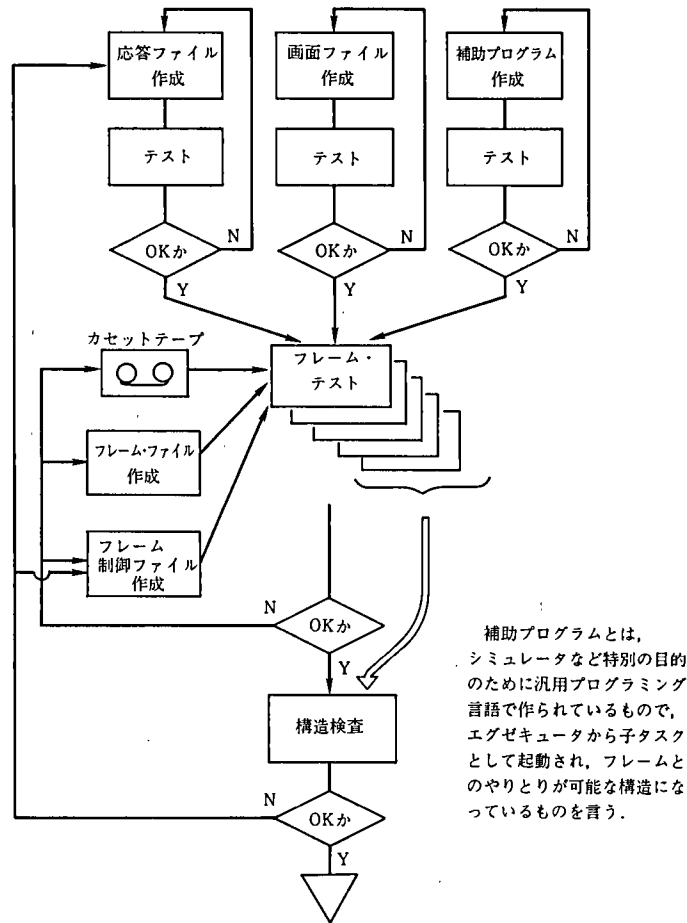


図9 コースウェア検証過程

Fig.9 The verifying process of the courseware

- るようにした。このため、待ち時間が分散され、学習者に待ちを意識させない。
- 2) グラフィック・ドライバにより描画スピードを向上させている。
 - 3) 予想解答の設定方式を AND/OR グラフの探索手法を適用しやすい形にしたことで、応答内容とのマッチングにおける真偽が最短のルートで確かめられる。

2.6 学習効果面での対応

パソコンによる CAI が普及した原因の一つに、その豊富なグラフィック機能がある。原理や概念といった抽象的な内容を教える場合、文字情報に較べて、絵や図の方が優れていることが多いし、本題から外れていても楽しい絵があれば学習者の心を和ませるといった別の効果もある。

すでに、LEARN UP は、学習効果を上げるために音声を用いているが、今回、さらに次のものをコースウェアに取り込めるようにした。いずれも、画面作成の過程で取り込めるようにしている。

- 1) アニメーションの取り込み……グラフィックを使った動画（アニメーション）は、次のようにしてコースウェアに取り込む。まず、動かす絵を通常の画面と同様に作成してアニメ部品として登録する。次にそれを「アニメ登録」で作成中の画面に取り込み、その移動位置や移動スピードを指定する。その際、アニメ部品を次々と取り換えれば、人の歩く様や形の変容する様を表すこともできる。

なお、移動位置を数式*で指定することもできるようになっている。この場合は、学習時に学習者の答えた値によって、その軌跡を描かせることができるので、簡単なシミュレーションとしても利用できる。

- 2) 画像と映像の取り込み……本物の画像や映像でないと効果が半減するという教育内容の場合には、TV-Photo**、VTR、レーザーディスクといったメディアが要求される。また、CAI 導入以前に上記メディアで教育していたところでは、制作工数を減らすため、それらをコースウェアに取り込めるようにとの要望が強い。

画像や映像は、その 1 コマ 1 コマをグラフィック画面と同じ扱いにして、「画面取込」でコースウェアの中に取り込む。その際、メディアの種類とコマ番号の From, To を指定し、画面ファイルにコマンド情報として記録する。グラフィック・ドライバは、このコマンドを見つけるとコース・エグゼキュータに一旦制御を戻す。コース・エグゼキュータは、RGB からビデオに切り換えて指定された範囲の画像や映像を表示する。このとき、メディア自体の音声を使うか、カセットテープの音声を使うかの指示も取り入れる時点で選択できるようにしている。

3. オーサリング・システムの今後

ここでは、オーサリング・システムの今後の課題について述べてみる。

3.1 データベース型 CAI への移行

コースウェアの分岐が、個々の学習者に適応するために必須であることは先にも述べた。今回の改良では、学習者の直前の応答内容だけでなく、過去の学習履歴をもと

* 式の場合は、基点、スケール幅、限界値などで移動範囲を指定する。

** 磁気ディスクに画像を入れておいてテレビ画面で表示する装置。

に次に進むべきフレームを選べるようにした。しかし、フレームの考え方を残しているために教授内容とコースとは完全には分離されておらず、プログラム学習型 CAI の域を出ていない。

データベース型 CAI では、教授内容とコースを完全に分離することにより、個々の学習者の理解度に合わせたコース（決められた枠組みの範囲内ではあるが）をコンピュータ内部で生成する。さらに、教授内容の構造を定義したスキーマを変えるだけで、教授内容の入っているデータベースを変えずに、個々の学習者に適した教授方略を適用できるといった利点を有している^[2]。コースウェアの個人差対応といった面では、この型の CAI は今回のものと較べて、はるかに実施しやすいシステムである。

今回はコースウェアをすべて部品化しているので、文字通りフレーム（枠）をはずすことで、比較的スムーズにデータベース型 CAI へ移行できると考えている。その場合に必要となるオーサリング機能としては、データベース中の個々の教授内容を学習者に合わせて最適な系列と構造に並べ換え、そのスキーマ・ファイルの作成を支援するものである。

3.2 コースウェア設計での支援

オーサリングは、コースウェア制作全般にわたって支援するのが理想の姿である。現在、設計済みの教材を作成し試行する機能は有しているが、設計を支援するものはない。コースウェア設計では、その作業の一つとして、コース・アウトラインの決定がある。教授項目が最適な系列に並べられているか否かは、学習の効果と効率の面からみて大変に重要であり、この部分を支援するツールを今後備えねばならないと考えている。その内容は以下のようなものである。

教授項目間の関連構造を分析する手法として ISM (Interpretive Structural Modeling) 法^[3]と IRS (Item Relational Structure) 法^[4]がある。ISM 法は、コースウェアを作成する前に適用される手法*で、一方の IRS 法は学習履歴（コースウェアの試行結果）をもとに教材構造を見直し、訂正するのに適用される手法である。それらを支援ツールとしてコースウェアの制作工程に取り入れたものが図 10 である。

3.3 DAT (Digital Audio Tape) の利用

複雑なコース分岐を可能とするには、音声を頭出しする速さの問題がある。現在のカセットテープでは、その要求に応じることができない。

CD (Compact Disc) やレーザーディスクは、瞬時に頭出しできるが、今の技術ではユーザが自由に書いて修正することはできない。

その点、DAT はカセットテープと同じように扱えるほか、カセットテープに較べて 10 倍近い高速で頭出しができる、録音時間が 120 分と長い、音声だけでなくプログラムやデータなどのデジタル情報が記録できる、といった長所を有している。

また、DAT を使うことにより、現在、教材ディスク、音声教材用カセットテープと二つのメディアに別れている教材を 1 本の DAT に納めることが可能となる。そして、教材ディスク分のコストが削減できるだけでなく、ディスク装置も 1

* ISM 法の場合は、IRS 法のように学習結果にもとづいて教材構造の妥当性を分析するものではないため、教授項目間の関連があいまいになりがちである。そこで最近では、ISM 法にファジィ (Fuzzy) 理論を取り入れた FSM (Fuzzy Structural Modeling) 法が試行されるようになってきた^[5]。

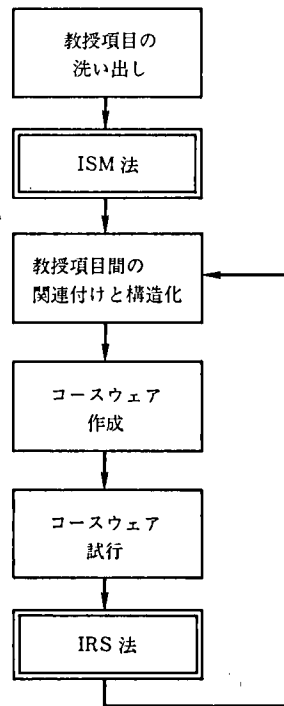


図 10 コースウェア制作工程における設計支援ツールの位置付け

Fig. 10 Designing methodology on courseware development

基——学習者個人のステータスを管理するためのディスク用——で済むことにより、パソコンのシステム価格をも下げることになる。

なお、上記のような DAT 利用を考えた場合のオーサリング機能は、教授内容とその説明のための音声コースの系列に合わせて記録・編集するものである。

3.4 ネットワーク化への対応

LEARN UP のような個別学習システムは、スタンドアロンによる利用が基本であるが、何十台かまとめて使用する場合は、運用面からネットワーク化することが要求される。また、コンピュータ・リテラシ（コンピュータ活用能力）の教育など、集団一斉授業の形態でパソコンを使用する場合でも、即時の個別対応を可能にする目的から、ネットワーク化が進められている。

そして、さらに発展させたトータル CAI システムでは、ホストコンピュータに汎用機やオフィスコンピュータを使い——学習者は、ホストコンピュータのリソースを共有できるので——プログラミングの実習や巨大なデータベースを利用した検索による学習、テストアイテム・バンク* によるドリル学習等が行われるようになる。

このような学習環境においては、従来の個別指導（チュートリアル）用教材のオーサリング・システムのほかに、検索用教材やドリル用教材のためのオーサリング・システムがそれぞれ必要となる。ともにデータベース化が鍵になる。

* Test item bank. よく練られたテスト問題や演習問題を、カリキュラムや学習目標に合わせて組織的に集めたもの。

4. お わ り に

昨今は、認知心理学を学習理論とする知的CAIの研究が人工知能の研究と相まってさかんに行われている^[6]。

現在の知的CAIは、学習領域に限定して——むしろ単一の課題に絞って——コンサルテーションやコーチなどを行うものである^[7]。したがって、課題が変われば、それに応じた専門知識（教授内容）、教授知識（教授方略）、学習者モデル、さらに学習者とのインタフェースをとるための対話知識と、そのほとんどを作り直さなければならない。しかも、先生一人でコースウェアを作れるような汎用オーサリング・システムは、まだ研究段階であり、知的CAIの普及には相当時間がかかると思える^[8]。

われわれの目指すオーサリング・システムは、あくまでも先生自身によるコースウェア作りを支援するものである。実用的なものを手軽に作れて、徐々に改良してゆければ、プログラム学習型やデータベース型であっても良いと考えている。要は、先生の授業におけるCAIの活用の仕方いかんである。

したがって、人工知能については知識表現や推論など、その要素技術の一部を用いて、さらにコースウェアを作りやすくすることが当面の課題と考えている。

なお、現在LEARN UP B版ソフトウェアに、さらにマシン・インタフェースに改良を加えたLEARN UP C版ソフトウェアを開発中である。使用の実績を見ながら、今後も改良を続けたい。

-
- 参考文献 [1] 細井 正・中新俊夫・岡田 豊，“パソコンCAI「LEARN UP」によるOA教育の実践”，電気通信学会，信学技報，ET83-10，1984。
 [2] 細井 正，“オーサリング・システムの動向”，テクニカルアイ，Vol. 4，No. 4，1987。
 [3] 佐藤隆博，千村浩靖，“キー概念と教材モジュールを関連づけて表わした教材構造グラフの作成法”，日本教育工学雑誌，Vol. 6，No. 4，1982。
 [4] 竹谷 誠，“IRS テスト構造グラフの構成法と活用法”，日本教育工学雑誌，Vol. 5，No. 3，1980。
 [5] 松原 勇，“ファジィ性を考慮したオーサリングについて——ファジィグラフによる教材構造の分析の一手法——”，CAI学会第12回研究発表大会論文集，B1-1，1987。
 [6] 管井勝雄，“CAI研究の可能性と今後の課題”，日本教育工学雑誌，Vol. 7，No. 4，1983。
 [7] 岡本敏雄，“知的CAI構築のための技術課題”，工業技術セミナー，T-1355，1985。
 [8] 大槻説乎，“知的オーサリングシステムの設計とその実際”，工業技術セミナー，T-1355，1985。

執筆者紹介 出石 修 (Osamu Izuishi)

昭和46年東京理科大学工学部経営工学科卒業。同年日本ユニシス(株)(当時日本ユニバック(株))入社。シリーズ1100のSEサービスに従事した後、同56年LEARN UPシステムの開発に参加。以後、主としてコースウェア開発を担当し、現在に至る。システム開発統括部ソリューションソフトウェア部CAIソフトウェア課に所属。CAI学会会員。日本教育工学会会員。



UNIX マシン UNISYS 5000/70 上での 第4世代言語 LINC の実現

Implementation of the 4th Generation Language LINC under UNIX

佐々木 政和

要約 本稿は第4世代言語 LINC (Logic and Information Network Compiler) の UNIX* マシン (UNISYS 5000/70 システム) 上での実現方法について述べている。

LINC は、「システム開発の品質向上」と「生産性の向上」を目的としてエンド・ユーザのために開発された言語で、現在は MCP (Master Control Program, UNISYS A/B/V シリーズの OS) の下で広く使用されている。

5000/70 システムの LINC は、今までの MCP 上の LINC とはまったく異なる構成となっている。

本稿では、次に示す主な機能を中心に紹介する。

- 1) 翻訳時システム
- 2) 実行時システム
- 3) データベース制御および障害回復システム
- 4) ユーティリティ
- 5) 他システム以外とのインタフェース
- 6) 機密保護
- 7) CENTIX** (5000/70 のために UNIX を拡張した OS) カーネルのカスタマイズ
- 8) システム開発と運用方法

とくに、LINC のために開発されたデータベース管理システムであるデータ制御システムは、五つのプロセス (イメージの実行^[8]) から構成された複雑なもので、ビジネス・ユースで欠かせない障害回復機能を持っている。

代表的な第4世代言語である LINC が、標準 OS と言われる UNIX 上で実現されたことは意義があるものと考えられる。

Abstract This is a description of how to implement LINC system on 5000/70 which is the UNIX system. LINC (Logic and Information Network Compiler) is the 4th generation language, which is very popular under existing operating system (MCP) on UNISYS A/B/V series.

This paper describes the major components of LINC on 5000/70 shown below;

- 1) Compile time system
- 2) Run time system
- 3) Database control and recovery system
- 4) Utilities
- 5) Interface with other system
- 6) Security

* UNIX は、AT&T ベル研究所が開発した OS の名前であり、同研究所の登録商標である。

** CENTIX は、UNISYS 社の登録商標である。

- 7) Customize of CENTIX kernel
- 8) System development and operation

Data control system which was developed as a database management system of LINC, is a complex one with 5 processes, provides very significant recovery functions for its business usage.

LINC is able to contribute to software productivity and quality improvement.

The implementation of LINC on UNIX which is very popular, will be significant in the proliferation of 4th generation languages.

1. はじめに

LINCは、UNISYS社が1982年9月に発表した第4世代言語である。LINCの基本機能は、LINC定義言語(LINC Definition Language: LDL)で記述されたオンライン・データベース・システムの仕様から、システムに必要なすべてのプログラム・モジュールを生成することである。これにより、オンラインやデータベース、オペレーティング・システムに関する知識を持った要員が参加しなくても、高度な情報システムを構築することができる。

また、システムの変更や拡張の保守作業もLDL言語による仕様を修正し再編成するだけで、各プログラム・モジュール間のインタフェースが自動的に保たれる。LINCは、第4世代言語に必要とされる次のような特徴をもっている。

- 1) 開発期間の短縮化(迅速性)……プログラム・バックログの解消
- 2) プログラム保守作業が容易(変化への対応性)……ソフトウェアの陳腐化の防止
- 3) ハードウェアの互換性(将来への発展性)……小型コンピュータから超大型まで
- 4) プロトタイピングによる開発(適合性)……開発途中での検査が可能
- 5) エンド・ユーザによる開発(ユーザ主導)……要求への前向きな対応

LINCでは、次の三つの要素を用いて、各種の情報とそれらの利用方法(ロジック)を定義する。

- 1) コンポーネント……ビジネス活動に必要な情報資源(顧客や商品、従業員などのマスク情報)を格納する。
- 2) イベント……売上・仕入・支払など日々のビジネス活動(トランザクション)を記録する。
- 3) プロファイル……コンポーネントやイベントに記録された情報をアクセスするための論理視点。

システム開発者は、これらのコンポーネント、イベント、プロファイルを記述することにより、データベースの生成を行うことができる。在庫元帳や入出庫伝票の入力画面を作成するだけで、データベースの物理構造の設計などの作業はLINCにより自動的に行われる。このようにLINCは、エンド・ユーザにとって重要な入出力画面の設計を中心とした開発システムであり、データベースの障害回復機能のような複雑な処理は、ユーザが意識することなくLINCが自動的に生成する。

最近、UNIXオペレーティング・システムを使用したマシンがビジネス・ユースで注目されている。本稿では、UNISYS社の5000/70のUNIX上に実現したLINCについて、その実現方法と機能を述べる。

2. 5000/70 の LINC システム実現方法

はじめに UNISYS A/B/V シリーズの LINC の概要を説明し、次に 5000/70 でのように実現したかを述べる。

2.1 A/B/V シリーズの LINC 概要

A/B/V シリーズの LINC について図 1 に示す。LINC コンパイラは、LDL ソースファイルから NDL ソース、GEMCOS ソース、COBOL 74 ソース、DASDL ソースを生成する。その後、生成されたソースファイルを各コンパイラがオブジェクト・プログラムに翻訳する。

また、データベース管理システムは DMS II (A/B/V シリーズ用のデータベース管理システム) を使用している。したがって、LINC システムは既存のコンパイラおよびデータベース管理システム上に構築された統合化システムの生成ツールと言える。

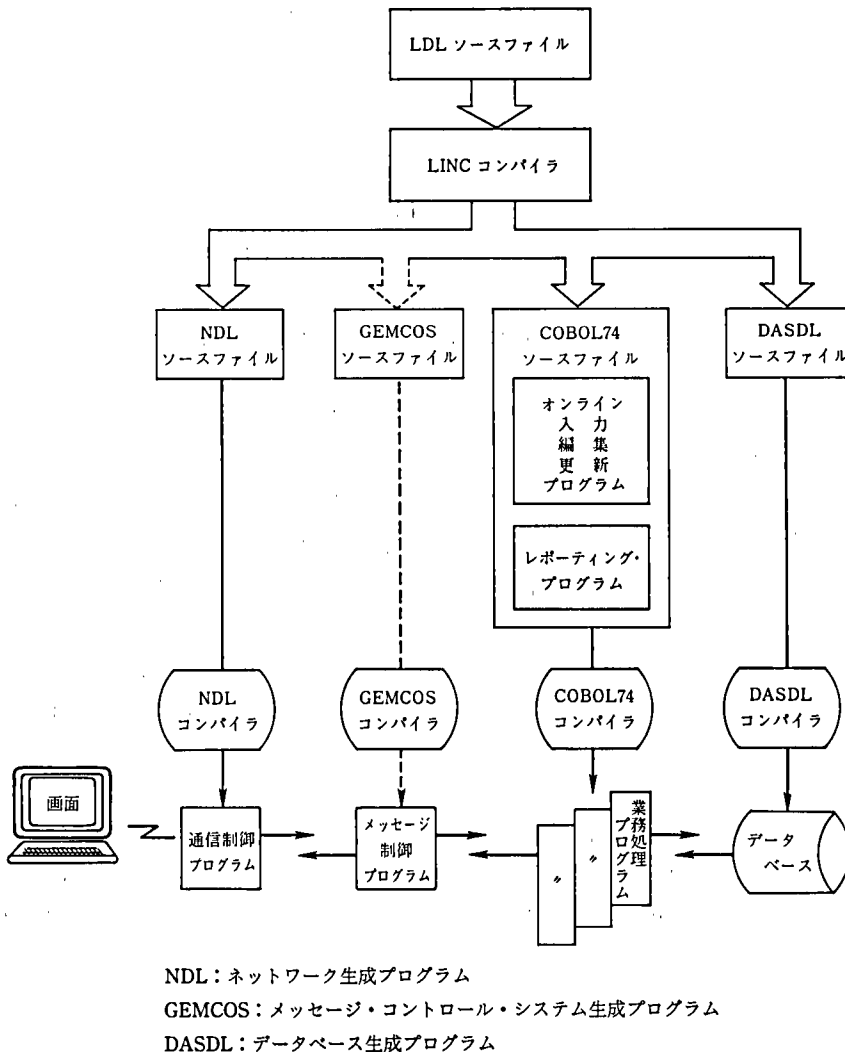


図 1 MCP 上の LINC I システム

Fig. 1 LINC I system under MCP

2.2 5000/70 システムにおける LINC の実現

5000/70 の LINC は、A/B/V シリーズの LINC I*(10.3バージョン)の LDL で記述された LDL ソースプログラムを 5000/70 上で翻訳し、実行することを目的として開発された。

2.1 節で述べたように、LINC にはデータベース管理システムが必要であるが、これをどのように実現するか、また生成するプログラムをソースコード、中間コード、オブジェクト・コードのいずれにするか、プロセスをどのように動作させるか(カーネルプロセス**、ユーザプロセス)ということがポイントとなる。

5000/70 では、図 2 のように LINC のサブシステムを開発している。端末の制御はスクリーン制御システム(Screen Control System : SCS)が行い、データベースファイルのアクセスはデータ制御システム(Data Control System : DCS)が行う。

2.3 コンパイラとインタプリタ

5000/70 では、LDL ソースプログラムを翻訳し、L-コード(L-code)という中間コードを生成し、それを仮想 LINC マシン(LINC インタプリタ)で解釈実行する方式を採用した。LINC/LIRC***のコンパイラおよびインタプリタをすべて開発しなければならないが、それは次のような利点を持っている。

- 1) 翻訳時間(LDL ソースプログラムから実行可能プログラムに変換するまでの時間)の短縮。A/B/V シリーズの LINC は一度ソースプログラムを生成し、それを翻訳するために無駄な時間を費す。5000/70 では中間コードに直接翻訳するために高速である。
- 2) 中間コードに翻訳するために、他の CPU への移植が容易である(ただし、インタプリタの移植は必要)。

一般的に、LINC/LIRC アプリケーション・プログラムは、ディスク装置や端末装置への入出力処理が多い。このようなアプリケーションを想定した場合は、インタプリタの内部処理(中間コードを解釈実行する)時間は問題とならない、むしろ SCS、DCS および各プロセス間のプロセス間通信(Inter-Process Communication : IPC)のオーバーヘッドがシステム全体のパフォーマンスに大きく影響する。

2.4 データベース管理システム

5000/70 では、データベース管理システムを持っていない。このため、複数ユーザが並行して同一ファイルにアクセスし、レコードを更新する機能を LINC 自身の内部に持たせる必要がある。

次の 2 通りの方法が考えられるが、LINC では 2)の集中化システムを採用した。

- 1) 複数プロセスが、それぞれ同一ファイルを更新する手法(分散化システム)(図 3(a))……ユーザ単位に複数同時に実行されているインタプリタから、直接ファイルにアクセスする手法(データベース管理システムをインタプリタに埋め込んだシステムと考えられる)である。CENTIX の持つレコードロック機能をカーネ

* LINC には、LINC I のほかに 1985 年 10 月に発売された対話型ですべて開発できる LINC II がある。

** カーネルモードで処理される特権プロセス

*** LIRC(Logic and Information Report Compiler)はバッチ処理により、LINC のデータベースをアクセスしたり帳表を出力するためのプログラミング言語である。

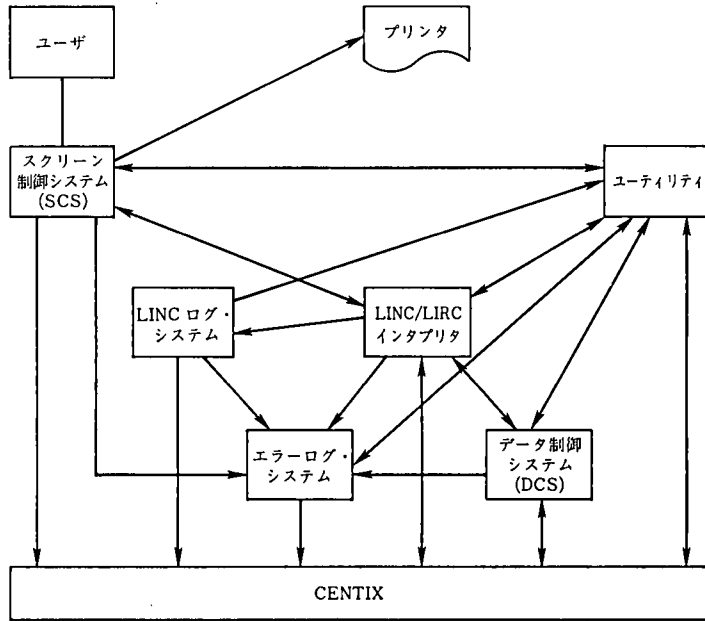


図 2 5000/70 での LINC サブシステム

Fig. 2 5000/70 LINC subsystem

ル内部で動作させることにより、データ・ファイルへの並行同時アクセスが可能である。実現レベルでは、比較的容易であるが大きなメモリを使用するプロセスが、ユーザ数だけ実行されるために、多くのメモリを消費する(実メモリ容量が少ない場合は、ページアウトやスワップアウトが多く発生する可能性がある)ために、LINC では採用していない。

- 2) データベースを管理するサーバプログラムにより同一ファイルを更新する手法(集中化システム)(図 3(b))……ファイルを制御するために DCS というサーバをユーザ・プロセスとして実行する方法である。インタプリタ(クライアント)からファイルをアクセスする要求が出された時は、すべて DCS サーバが処理を行う。インタプリタとスクリーン制御プロセス(SCP)は、ユーザの起動分だけプロセスとして動き、DCS は同一 LINC システムでただ一つだけ実行される(複数の LINC システムを実行する場合は、その LINC システムの数だけ DCS が動く)。これによりメモリの使用量は、最小限におさえられインタプリタと DCS は、プロセス間通信を使いメッセージの送受信を行う。

この集中化システム(クライアント-DCS サーバ方式)の最大の課題は、「いかに効率的に IPC を実現できるか」という点である。LINC ユーザ(端末)数が増加すれば、多くのインタプリタがクライアント・プロセスとして実行されるため、DCS サーバがボトルネックになる可能性が高くなる。高度な IPC と効率的なサーバの設計が不可欠となる。

この点は、5000/70 はマルチプロセッサ・システムであるために、これらの LINC プロセスを複数の CPU にそれぞれ分散させることで多少回避できる。

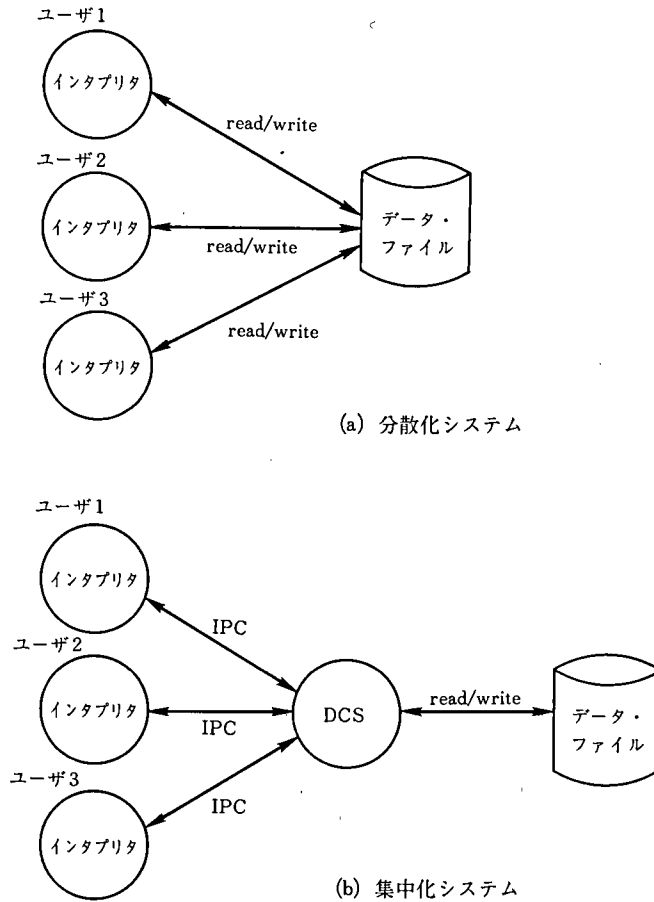


図3 分散化システムと集中化システム

Fig. 3 Distributed system and concentrated system

たとえば、アプリケーション・プロセッサ (AP) が 2 台設置されたシステムでは、1 台の AP で DCS と端末 2 台分のインタプリタ、SCP、ログプロセスを実行し、他の AP で DCS 以外の端末 5 台分のプロセスを実行するのである。DCS のようなバックグラウンド・プロセスは、高速なプロセッサで実行するのが望ましい。

この集中化システム方式では、カーネル内のレコードロック機能を必要としない。システム V R 2.0 以前の UNIX では、レコードロックのためのシステムコールを持たなかったため、ビジネス・システムで UNIX を利用する場合の一つの欠点として「ファイル/レコードロック機能を持たないこと」が指摘されてきた。これについては、文献^[1]の第 7 章で解説されている。

3. 翻訳時システムの機能

コンパイラは 2 パス方式であり、パス 1 では LDL ソース (LINC/LIRC ソース) を読み込んで字句解析、構文解析を主に行い、パス 2 はパス 1 が正常に終了した段階で

実行され、データ構造をより詳細に拡張してディスク上に保存する。図 4 に翻訳時システムの関連図を示す。

コンパイラにより、ネットワーク定義ファイル*、L-コードファイル、アプリケーションシンボル・ファイルが直接生成される。画面ファイルはスクリーン定義システム (SDS) を通して作成され、データベース関連ファイルは DCS を通して作成される。また、LDL ソースのオプションの指定により、LDL リストファイルが出力される。プロトタイピングによるシステム開発等の場合には、画面の数、項目の数、データベースの型等を変更する必要が発生することがある。このときは、LDL ソース上で再編成のオプションを指定することにより、既存のデータを残してデータベースを再編成することができる。

コンパイル時に CENTIX ファイル・システムのツリー構造を有効に使い、いくつか

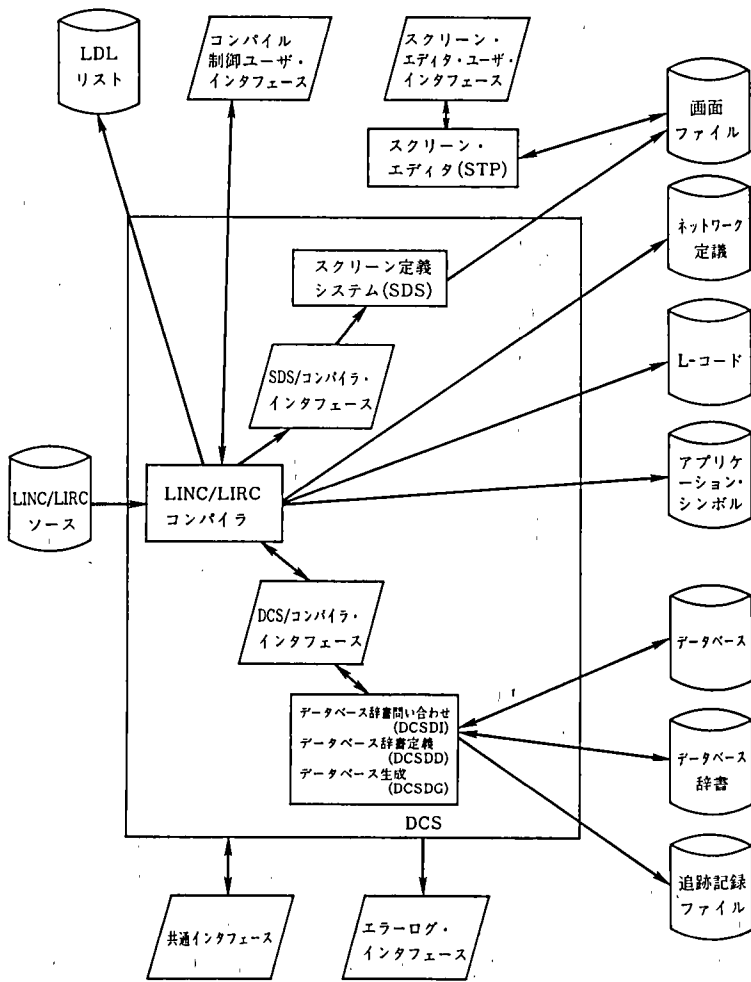


図 4 翻訳時システム

Fig. 4 Compile time system

* ログイン名と LINC 用の優先番号を関連付けたり、LIRC の出力データの処理を定義するファイル

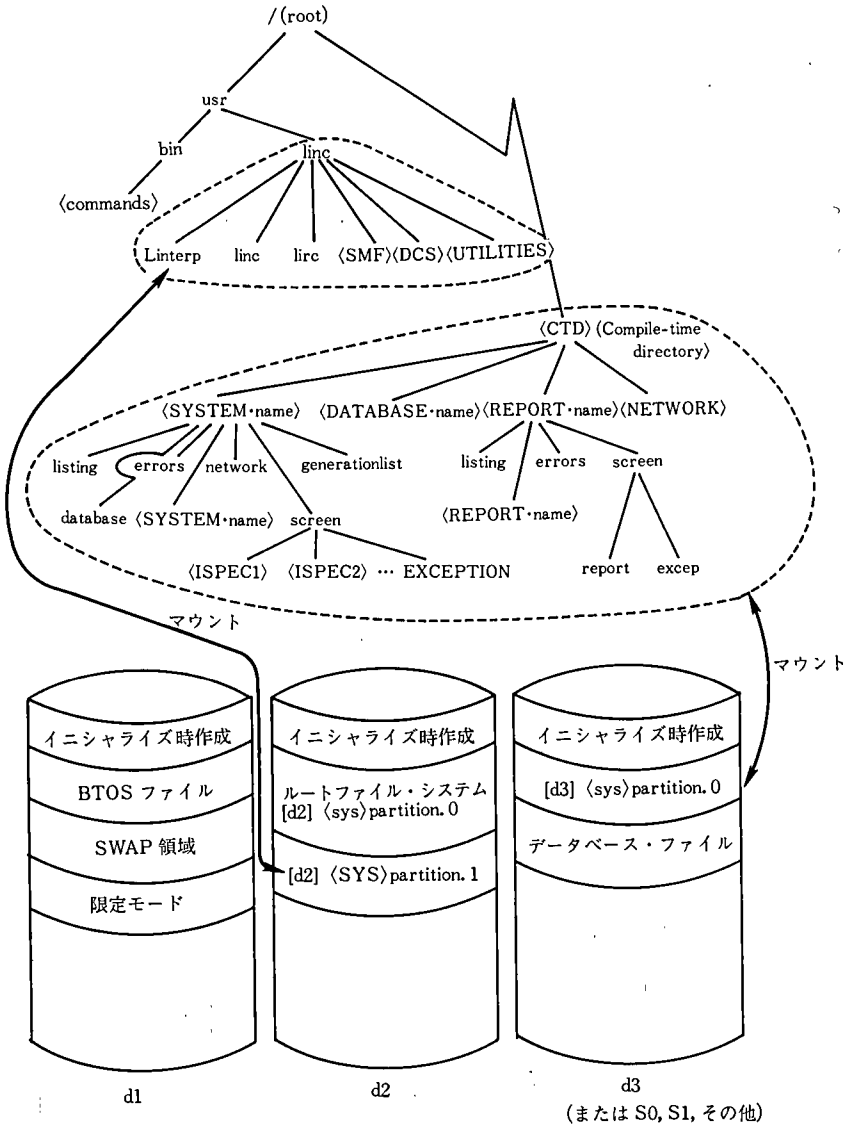


図5 LINC ファイルとディスク・ファイル
 Fig.5 LINC file and disk file structure

のディレクトリおよびファイルが生成される。図5にLINCシステムに関するファイルを示す。たとえば、ルート・ディレクトリの下に翻訳時ディレクトリ(<CTD>)でLDLソースを翻訳した場合に、そのプログラムに記述されたシステム名のディレクトリが作成され、その下にlisting, database, errors, L-コード(<SYSTEM.name>), network, generationlistが作られる。また、screenディレクトリの下にISPEC*の分だけ画面ファイルが作成される。またLIRCの場合も同様に、レポート名のディレクト

* イベントとコンポーネントの総称

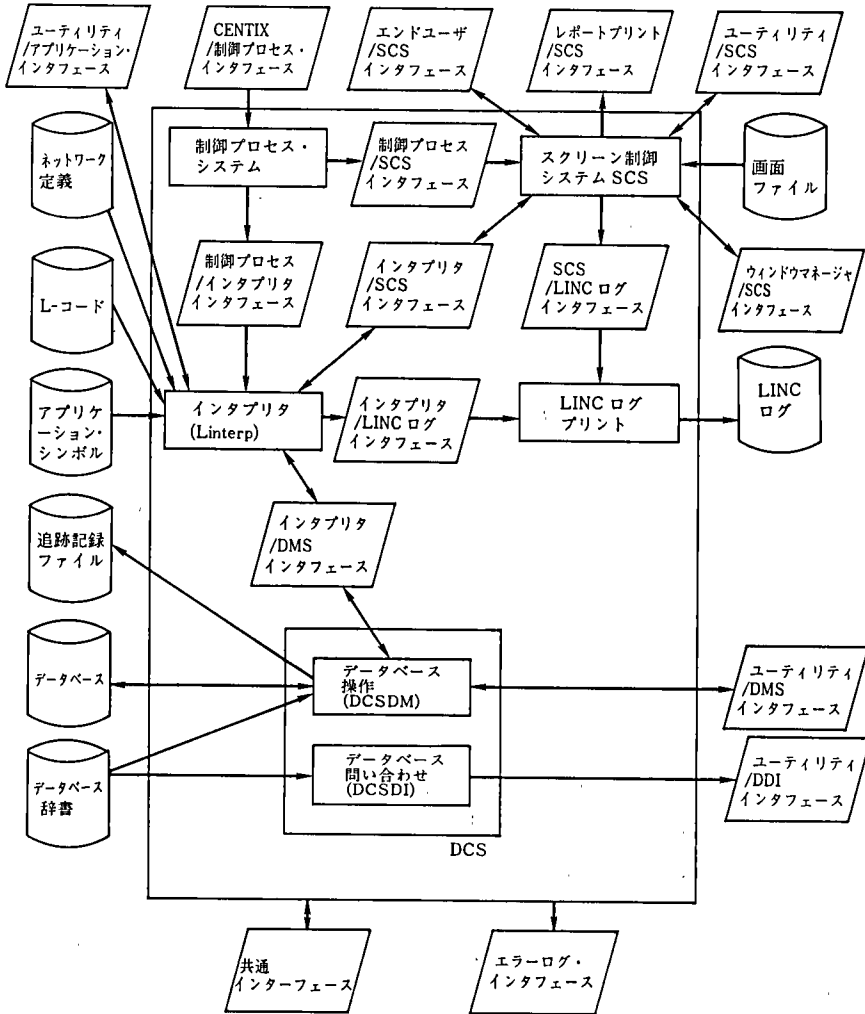


図 6 実行時システム

Fig. 6 Run time system

りが作成され、その下に LINC と同様のファイルが作成される。

一方 /usr/linc ディレクトリの下には、LINC システムを導入したときにコピーされた LINC システム用のファイル群が存在している。

なお、ここでは、LDL 仕様および生成される各ファイルの説明は省略する。それについては、5000/70 の LINC 関連のマニュアルを参考にされたい。

4. 実行時システムの機能

図 6 に LINC 実行時のサブシステムの関連図を示す。ここでは、制御プロセス・システム、スクリーン制御システム、データ制御システムについて、その機能を述べる。

4.1 制御プロセス・システム

制御プロセス・システムは、LINC 実行時の各プロセスを制御するためのものであ

表1 ステアリング・テーブル
Table 1 Steering table

linc	ErrProg	q1	q2	lrun	ErrProg	q1	q2
	Linc				LogMgr	q3	q4
					SCP	q5	q6
					Linterp	q6	q5 q7 q8
lirc	ErrProg	q1	q2	backup	ErrProg	q1	q2
	Lirc				SCP	q3	q4
					BackUp	q4	q3
gli	ErrProg	q1q	q2	lincarchive	ErrProg	q1	q2
	SCP	q3	q4		RmDb		
	DataConv	q4	q3				
linctrans	ErrProg	q1	q2	rmarchdb	ErrProg	q1	q2
	LincTrans				RmArchiveDb		
datatrans	ErrProg	q1	q2	generatedb	ErrProg	q1	q2
	DataTrans				GenerateDb		
ErrProg : エラーログ・プロセス				lcp	ErrProg	q1	q2
LogMgr : LINC ログ・プロセス					LogMgr	q3	q4

る。LINC は、複数のプロセスが並行して動作するように実現されているため、これらのプロセス制御をするのが目的である。

LINC 用のコマンドは /usr/bin ディレクトリに含まれていて、これらはすべて ACP (Application Control Program) にリンクされている。また、LINC のコマンドの実体は /usr/linc ディレクトリに存在する。

ここでは、ASYS というシステム名を持つ LINC システムを実行した場合の制御プロセス・システムの動きを例として説明する。

- ① 操作員は CENTIX にログインし、ウィンドウ・マネージャを起動する(通常は、. profile に設定する。ウィンドウ・マネージャが実行されていない場合は SCP が直接画面を制御する。)
- ② 次のコマンドを使用して LINC を起動する。

```
$ lrun ASYS [Return]
```

- ③ lrun にリンクしている ACP が、lrun ASYS を引数として実行される。ACP はステアリング・テーブル(表 1)を読む。このステアリング・テーブルは、産み出されるプロセスの名前、各プロセスに必要とされるキューの数、どのプロセスがキューにリンクされるべきかという情報からできている。lrun の場合は、ErrProg, LogMgr, SCP, Linterp(インタプリタ)プロセスを実行する。なお、ErrProg と LogMgr はロギングのためのプロセス(ログプロセス)であり、ASYS の LINC 実行時は、ただ一つのみ実行される。

ログプロセスは、すべての LINC ユーザを管理するために新しいプロセスグループを作り、起動された端末から独立したプロセスとなる。

- ④ 各プロセス間で使用するメッセージキューを作成する。

- ⑤ インタプリタからデータベースへの要求が出された場合は、関連する DCS プロセスを作成する (DCS 関連のプロセスは、ステアリング・テーブルからは産み出さない)。
- ⑥ プロセス相互間で、プロセスがどういう状態で終了したか (正常または異常)、再生成の必要があるか (ログプロセスが異常終了した場合は再び生成される)、関連するプロセスを終了するかなどを適切に管理する。

4.2 スクリーン制御システム

スクリーン制御システムは、操作員と LINC を結ぶために、端末を制御するプロセスである。インタプリタと同様に、端末の数だけ実行されるサーバプロセスであり、インタプリタから端末に対する要求が出された場合に、必要な画面ファイルを表示したり、キー入力されたデータをインタプリタに渡したりする。インタプリタ、LINC ログプロセス、エラー・ログプロセスと IPC を介し通信している。

4.3 データ制御システム

この DCS は、その機能により、次の 5 種類の DCS 関連プロセスに分類される。

- ① データベース定義 (DCSDD, DCS Data Definition)
- ② データベース生成 (DCSDG, DCS Data Generation)
- ③ データベース操作 (DCSDM, DCS Data Manipulation)
- ④ データベース問い合わせ (DCSDI, DCS Data Interrogation)
- ⑤ データベース・クラッシュ・リカバリ (DCSCR, DCS Crash Recovery)

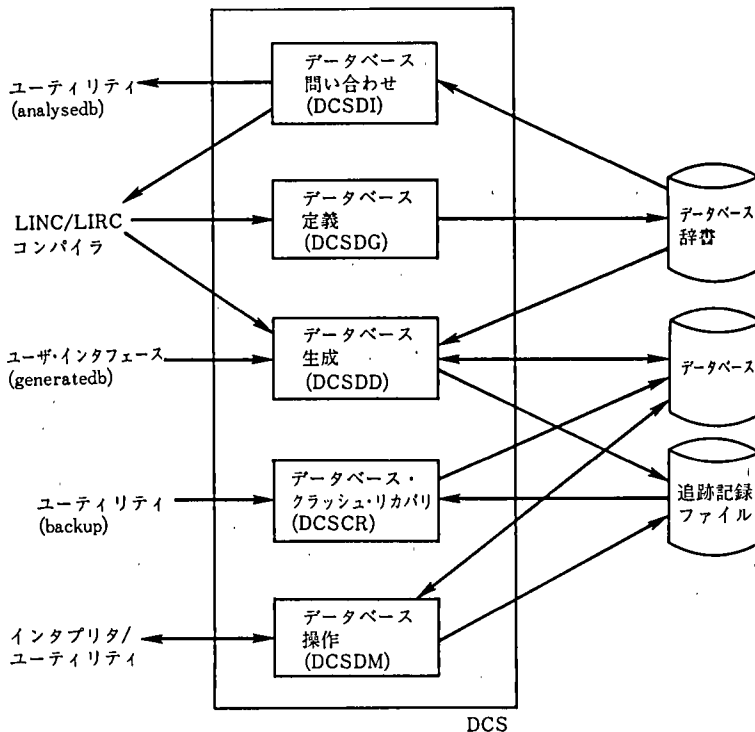


図7 データ制御システム(DCS)

Fig. 7 Data control system(DCS)

DCS はインタプリタやユーティリティからの要求により、サーバプロセスとして同一 LINC システム上で一つ実行される。DCS のプロセス関連図を図 7 に示す。

たとえば、LINC/LIRC コンパイラは、DCSDD, DCSDG プロセスを通じて、データベース辞書、データベース、追跡記録ファイルを生成する。

データベース分析用の analysedb ユーティリティは、DCSDI, DCSDM によりデータベース辞書とデータベースを検索する。また、generatedb ユーティリティは、データベースの再編成を実行するために DCSDG プロセスを産み出す。ディスクの入出力エラー等が発生し、データベースが破壊された場合は backup ユーティリティを実行し、バックアップのデータベース・ファイルを復元した後に、DCSCR プロセスが追跡記録ファイルからデータベースを回復する。アプリケーションやデータベースを更新するユーティリティは、DCSDM を通じてデータベースの検索・修正・削除を行う。

5. DCS の並行制御処理と障害回復機能の実現方法

5.1 DCSDM の機能

ここでは、DCS の中で重要な役割を持つ DCSDM についてその処理を述べる。DCSDM は、クライアント(インタプリタ、ユーティリティ)からのデータベースに対する読み出し/書き込み要求を制御するためのプロセスである。

LINC では、データ・ファイルへのアクセスは、集中化システムを採用している。このため、DCSDM は複数ユーザからの要求に対し、整合性を保ちながら並行して制御する機能を持たなければならない。

LINC では、1 トランザクションは 1 画面の処理単位に相当する。各トランザクションは、排他的でなければならない。つまり 1 トランザクションは、すべて実行されるか、もしくはまったく実行されないかのどちらかである。

DCSDM は、並行処理を実現するために各トランザクションの実行処理を、次の 4 段階に分けて管理している。

- 1) 動作段階……インタプリタからの要求(トランザクションの開始・終了、レコードの読み出し/書き込み、追加・削除等)により、メモリ上にトランザクション管理用のテーブルを作成する。
- 2) 確認段階……動作段階で作られたトランザクション管理用テーブル内で、トランザクション間での不整合(デッドロック等)がないか検査する。
- 3) 保管段階……トランザクション・データを追跡記録ファイルに書き込む。
- 4) 書き込み段階……データベースにトランザクション・レコードを書き込む(書き込み要求の場合)。

インタプリタから見た場合は、自分のプロセス(クライアント・プロセス)が DCS サーバを独占しているように見える。

5.2 DCS の障害回復機能

データベース・システムにおいては、ソフトウェアまたはハードウェアの故障により、失うデータを最小限にし、障害発生直前の状態まで復帰回復する機能を持つことは不可欠である。

DCS は、リスタート・リカバリ (Restart Recovery) とクラッシュ・リカバリ (Crash

Recovery) の 2 種類の障害回復機能を持ち、データの保全性を完全なものとしている。

- 1) リスタート・リカバリの実現方法……LINC を実行中に、システムが異常終了 (メモリ上のクラッシュ) した場合に最終トランザクションの直前まで回復する機能である。クラッシュ後に再度実行した場合、データベース・ステータス・ファイルを読み、データベースが前回正しく終了しているかを検査し、異常終了の場合は次の処理をする。
 - ① 画面上にそのメッセージを表示する。
 - ② データベース・ファイルより、最終の追跡記録ファイルの番号を得る。
 - ③ 追跡記録ファイルのヘッダ・レコードを読み、最終トランザクション・レコードの位置を得る。
 - ④ 最終トランザクション位置より元に戻ってチェックポイント・レコード (データベース・ファイルをクローズすることにより、メモリとディスクの内容を確実にする区切り) を探す。
 - ⑤ チェックポイント・レコードより、最終トランザクションまでのデータを使用しデータベースを回復する。
 - ⑥ 最終トランザクション・レコードの次に、チェックポイント・レコードを書き込む。

操作員は、最終トランザクションがデータベースに正しく格納されているか検査し、万一失われたデータがあれば再入力する。

- 2) クラッシュ・リカバリ……リスタート・リカバリが成功しなかった (たとえばディスクの故障、データベース・ファイルの入出力エラー等が発生した場合)、**backup** ユーティリティを使用して前回に取ったデータベースのバックアップを復元し、バックアップ後に発生した追跡記録ファイルから、エラー発生直前トランザクションまで回復させる。これをクラッシュ・リカバリという。

6. LINC ユーティリティ

UNIX オペレーティング・システムは、比較的ソフトウェア開発環境が整備されているために、システム・プログラマや研究者に支持されているが、このコマンド名、オプション、出力メッセージは、一般のエンド・ユーザには不向きであると言われている。

実際に、UNIX のコマンドやライブラリ関数/システムコールを修得するためには、大変な労力と期間を必要とする。

このため、LINC では必要最小限のシェルコマンドを身につけるだけでアプリケーション・システムを構築できるように、21 種類のユーティリティ (表 2) を提供している。これらは、一般のシェルコマンドと同様に `/usr/bin` ディレクトリに含まれているために、ユーザからは他のコマンドと同一に見える。

たとえば、LINC のデータベース・ファイルをテープにバックアップ取る場合は、**cpio** や **tar** のような一般のシェルコマンドではなく、**backup** ユーティリティを用いるようにしている。

表2 ユーティリティ一覧表
Table 2 Utility list

ユーティリティ(コマンド)	概 要
analysedb	LINCで作成したデータベースの構造を知る。
backup	LINCデータベースのバックアップおよびリカバリ。
lincuptrans	B 1000 LINCのLDLソーステープを5000/70のディスクに入れる。
lincuptrans	5000/70 LINCのLDLソースをB 1000/Aシリーズ用に交換する。
dictutil	現在のトランザクション番号とアカウント月を変更する。
generatedb <システム名>	LINCデータベースの遅延や再編成を行う。
gli	LINC以外からのファイルおよびプログラム・インタフェース
gloss	画面を通してISPECフォーム、オプション仕様を設定する。
datatrans	B 1000 LINCデータベース・テープを読んでディスクに蓄える。
dataconv	ディスク上のB 1000 データベースを5000/70 LINCデータベースに変換する。
linc <ソース名>	LINCのLDLソースを翻訳する。
lirc <ソース名>	LIRCのLDLソースを翻訳する。
lrun <システム名> または lrun <リポート名>	LINC, LIRCを実行する。
lincdh <ソースファイル名>	ソースファイル中に特殊文字を入れることにより、見やすい書式リストを作成する。
lincquery	条件を設定してレコードの検索を行う。
loganal	LINCログの分析
rmdb <システム名>	システム名に関連したデータベース・ファイルをすべて削除する。
rmarchdb <システム名> <ボリューム識別子>	ボリューム識別子上のすべてのアーカイブ・ファイルを削除する。
setlogp	ロギングモードとログファイル・サイズを設定する。
stp	画面のフィールド以外の静的な部分を修正する。
xel	glossで作成したデータをLDLソースに変換する。

7. 他システムとのインタフェース

LINCのデータ制御システムは、LINC用に開発されたデータベース管理システムである。このため、LINC以外のプログラム(たとえば、COBOL, C, Pascal, シェル等)から直接アクセスすることは許されていない。そこで、この目的のためにLINCには、外部ファイルや外部プログラムとインタフェースするための機能(gli ユーティリティ)を持たせている。これより、LINCプログラムを変更せず、テープに蓄えられたデータをLINCのデータベースに格納することが可能となる。

また、データベース・レコードを抽出する機能(LIRCのextract文)を使用することにより、データベース・ファイルをCENTIXファイルにコピーしたり、その逆の実行が可能である。

図8は、COBOLプログラムとLIRC(抽出ファイルを利用)プログラムのインタフェースとしてFIFOファイルを使用した例であり、COBOLプログラムからLINCの

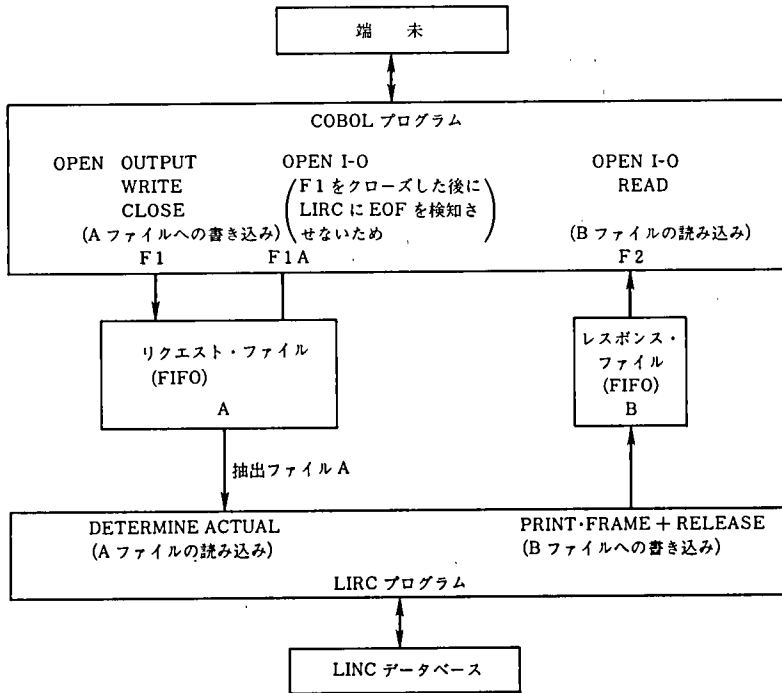


図 8 LIRC と COBOL インタフェース

Fig. 8 Interface between LIRC and COBOL

データベースをアクセスしている。

8. LINC 機密保護

LINC ユーザは、CENTIX の実ユーザ ID および実効ユーザ ID とは別に、LINC システム用の優先番号を持つ。このため、特定のユーザしかアクセスできない画面が定義でき、これにより LINC データベースへの機密を保護している。

9. CENTIX カーネルのカスタマイズ

DCS, SCP, インタプリタ等の LINC プロセスはすべてユーザ・プロセスとして実行される。したがって、カーネル内に LINC 用のソフトウェアは組み込まれていない。

ただし、LINC システムを効率よく動作させるために、次のカーネル・パラメータを変更している^[4]。

- | | |
|-------------------------|--------------|
| 1) ディスク I/O キャッシュバッファの数 | buffers=400 |
| 2) バッファのテーブルサイズ | hashbuf=128 |
| 3) スワップエリアのテーブルサイズ | nswbuf=144 |
| | swapmap=1800 |
| 4) 非スーパーユーザの最大プロセス数 | maxproc=50 |
| 5) メッセージキューの数 | msgmni=128 |
| 6) 一つのキューのメッセージ数 | msgtql=128 |

- | | |
|---|---|
| 7) シェアドメモリ・セグメントの
最大サイズ(4 K バイト/ブロック) | shmmmax=1024 |
| 8) シェアドメモリの合計 | shsmall=1024 |
| 9) ヒープ割り当てのブロック数
(4 K バイト/ブロック) | nlrg=8 |
| 10) カーネル仮想メモリパラメータ：
ページングとスワッピングに関する
パラメータを右のように変更する。 | minfree=16
desfree=50
lotsfree=64 |

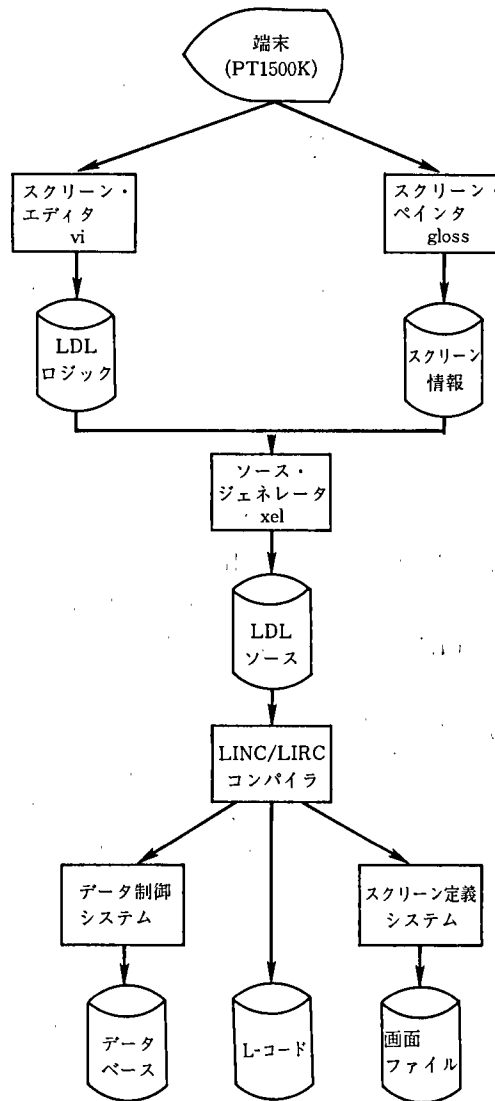


図9 gloss と xel を使用した LINC/LIRC プログラムの開発

Fig. 9 Generation of a LINC/LIRC program using "gloss" and "xel"

10. システム開発と運用方法

LINC では、図 9 に示したように画面設計(データベースの設計も含まれる)は、会話型の gloss ユーティリティを使用し、ロジック部分のみ vi コマンドで作成する。

LINC は、プロトタイプング手法で開発するために、エンド・ユーザはモジュールが完成する都度システムのシミュレーションができ、ソフトウェア開発の生産性と品質を高めている。

また、操作に自信が持てない場合は、ヘルプキーを押すことにより、不慣れな操作員を助ける機能を備えている。

11. おわりに

5000/70 システム上における LINC の実現方法をまとめた。LINC ソフトウェアは約 8 M バイトの大きさを持ちディスク上に存在する。このソフトウェア・プロダクトは、1984 年～86 年の間にイギリスのスコットランドの技術者 30 数名により開発され、日本で日本語処理機能を付加したものである。インタプリタ、コンパイラを含め、すべての LINC ソフトウェアは MODULA 2 言語で開発され、高度にモジュール化されていて、保守を容易にしている。

5000/70 は UNIX マシンであり、同時に小型の汎用オフィス・コンピュータとして位置付けられている。小型機で LINC を使用できる利点が評価され、日本では 1986 年に初めて顧客に導入され稼働している。現在では種々の業種(32 業種)で利用されており、LINC の汎用性が実証されている。なお、LINC の詳細は、文献^{[9]~[12]}を参照されたい。

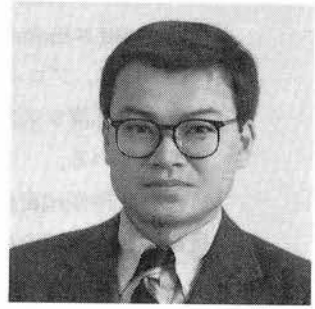
本稿が、今後予想される UNIX 上の同種のシステムの開発に参考になれば幸いである。

-
- 参考文献 [1] M. J. Rochkind 著, 福崎俊博訳, “UNIX システムコール・プログラミング” アスキー出版, 1987.
- [2] A. Filipiski, J. Hanko, “UNIX の 防 御 能 力” 日 経 バ イ ト, 1986 年 7 月 号, pp. 177~192.
- [3] P. Jackson “Unix System V 互換の OS を搭載したマルチマイクロプロセッサ・システム” 日経エレクトロニクス, 1983 年 11 月 7 日号, pp. 233~247.
- [4] “5000/70 CENTIX Kernel Customizer Operations Guide” (Release Level 6.0) UNISYS, 1987.
- [5] “UNIX SystemV プログラマ・ガイド, リリース 3.1”. 日本ソフトバンク出版事業部, 1987.
- [6] 塩谷修, “実用 UNIX システムプログラミング”, 日刊工業新聞社, 1986.
- [7] “5000/70 CENTIX Operations Reference Manual Volume 3 : System Operations Part 1” UNISYS, 1987.
- [8] D. Ritchie, K. Thompson, 石畑清, 小野芳彦訳 “UNIX タイムシェアリング・システム” 共立出版, bit, Vol.13, No.9, pp.1079~1094.
- [9] E. E. Rudolph, “Productivity in Computer Application Development”, バロースコンピュータセミナー資料.
- [10] 福間, “第 4 世代言語 LINC 初心者者が採用の“先べん”, 日経コンピュータ, 1986 年 3 月 31 日号.
- [11] 井尻, 戸田, 豊田, 福間, “第 4 世代言語がもたらすアプリケーションへのインパクト”, 日経コンピュータ, 1987 年 6 月 22 日別冊.

[12] 山下, 曾我部, “プロトタイピングによるシステム設計方法”, 日経コンピュータ, 1987年6月22日別冊.

執筆者紹介 佐々木 政和 (Masakazu Sasaki)

昭和58年明治大学工学部電気工学科卒業. 同年日本ユニシス(株)(当時パロース(株))入社. 5000/70システムのサポート活動に従事. 現在プロダクトサポート本部, プロダクト・サポート統括二部システム・プロダクト三部所属.



統合ネットワーク指向インテリジェント端末の開発 ——伊藤ハム(株)における事例

The Development of a Dedicated Intelligent Terminal for the Companywide Network——The Case of Ito Ham Foods Inc.

堀 籠 順 之 助

要 約 昨今の大企業は統合ネットワークとして、全国規模の営業所システムを展開しているが、その営業所内販売活動および商品在庫管理等を成功させるためには、必要な管理情報が最適な情報交換によって得られ、かつ適切な意思決定が各々の営業所で行われなければならない。したがって、通常エンドユーザはホスト・システムが一元的に管理しているデータベースをできるだけ活用すべく、ネットワークを利用したターミナル・システムをユーザ環境に立脚したシステムとして構築することになる。

今回紹介するシステムも、ネットワーク形態そのものはホスト集中型であるが、端末システムとしては多機能化を指向しており、基幹業務としての営業情報システムと付帯業務としての営業情報検索システムの2面性を持ったローカル・システムを構築している。なお、開発にあたっては、顧客要求を反映した専用端末を開発し、現業部門のニーズの吸収に努めた。

本稿においては、専用端末開発の全般的な紹介と将来像について述べる。

Abstract Recently the big enterprises develop the office system of national scale as the unified network. Their success in the sales activities and the stock control is owing to the fact that necessary management information is available by pooling information properly, so that the excellent decision making is put in practice in their each office.

Therefore end users usually develop the terminal system making use of a network, with the thought that it should be both based on the user's environment and able to utilize the database on a host computer. Though the form of a network is a host-concentrated one, the terminal system is oriented to the multi-functions and a local system including 2 subsystems is constructed on it; the business information system and the sales information inquiry system.

We have developed a dedicated terminal to comply with the needs of our customer.

The paper describes implementation of the dedicated terminal and what they will be in the future.

1. はじめに

昨今のターミナル・システムは、ハードウェアおよびソフトウェア両面にわたり進歩し多機能化してきている。文書処理を始めとし表計算、グラフ化等のOA的ソフトウェア・パッケージも年々高度化したものが市販されている状況である。

また、ローカル・システムの機能および機構が整備されるに従い、ホスト・システムとの結合を目的としたマイクロ・メインフレーム・システムも開発されている。さらに見のがすことができない要素として、ネットワークの制度的・技術的進歩がある。昭和60年4月1日付の第3次回線自由化とVAN法案の制定にしたがい、より自由に通信ネットワークを構築できるようになり、技術的にもより高速かつ高品質な通信

ネットワークが実現され、有効なターミナル・システム作りが可能となってきた。反面さまざまな機能が利用できるがゆえにホスト・システムとの分散処理を指向し、過度な環境設計を行うことにもなりかねない。

その結果として、利用者が使いにくい仕組みとか、パフォーマンスの低下をまねくシステム環境ができ上がってしまうことも起こりうる。本稿のような営業所システムの場合、柔軟な操作性、高度な経営戦略を旨とした管理情報の視覚的提供等を実現することによって、より使いやすい、そして必要なデータが容易に手に入るシステム作りが可能となる。

そのような背景を踏まえ、今回はユーザ・ニーズを重視し、専用端末を開発し、その上でアプリケーション・システムを作成した。開発期間は約10か月であり、開発完了後は伊藤ハム(株)の全国営業所に設置されている。アプリケーション・システムのなかでも、とくに営業情報検索システムは経営戦略システムの立案、および営業所での営業戦術システムの立案のために効果をあげている。

このシステムでは、情報検索、ならびに進捗管理データの検索およびビジュアル化を実現している。

以下、本システムの概要および開発に当たって考慮した点について述べる。

2. 営業所専用端末システム構築の考え方

営業所システムは、全国規模の総合ネットワークであり、本社のホスト・システム UNISYS B 7900 内のデータベースの有効活用および端末への一部機能の分散、そして将来におけるネットワークの拡大と品質の改良を目的としている。当面は、①旧営業所端末の質的改善、②営業情報検索システムの導入を基本方針とした。

2.1 基本的開発目標

1) 営業所旧端末の改善項目

- ① 日本語システムへの全面切り換え
- ② 入力操作の容易性および迅速性の追求
- ③ 処理の標準化

2) 営業情報システムの導入

- ① 営業管理情報のためのグラフィメージ処理の導入
- ② ホスト・システム・リンクによる情報検索システムの実現
- ③ ローカル機能の拡大

以上の基本目標達成のため、ハードウェアの設計段階より基本ソフトウェアに至るまで次の設計概念を取り入れた。

2.2 主要な設計概念

- 1) ハードウェア設計……cpuは80286タイプとし、クロックは10MHzに設定した。ディスプレイの表示機能は、キャラクタ、グラフィック、および合成表示を可能にし、漢字はJIS第1、第2水準および外字ROM対応とした。また、外部インタフェースの種類としては、セントロ、シリアル、ハンディ端末仕様ジャックイン、MCA(Multi Channel Access)およびSDLCフレーム・インタフェースを用意した。

このほか、補助記憶装置としてフロッピー・ディスク、および 20 MB の固定ディスクを接続した。

- 2) ソフトウェア設計……ソフトウェアの開発を実施するに当たり、その開発環境としてオペレーティング・システムは MS-DOS 系、開発言語は COBOL と BASIC をベースとした。グラフィメー処理は BASIC で開発し、その他は機能別に言語を選択する方針を取った。また、システム構造は機能別階層構造を採用し、それに基づきプログラム間のリンクプロトコルを決定した。ネットワーク機能では、将来のネットワーク拡大に備えて、専用プロトコルと汎用プロトコルの両方を搭載する方向に決めた。

3. 専用端末の開発

開発に当たっての基本目標および設計概念は、ユーザの要求の分析結果によるものであるが、とくに今回のローカル・システムを構築するに当たり配慮しなければならない要件として、戦略的な情報検索システムの開発があった。しかし、ターミナル・システム上で大規模なシステム開発を行うためには、処理効率の向上、操作の容易性、リソースの最大限の活用が要求される。

たとえば、既存の端末を活用する方法では端末接続入出力の強化・拡充および端末台数の増加等、すべてにシステム・コストの増大が不可欠となる。このほか、ターミナル・システムが複雑化するため開発の長期化が予測され、ソフトウェア投資コストも増大する結果となる。

したがって、コスト・パフォーマンスに見合うシステム環境を構築するためには、専用端末を開発する必要がある。

投資効果をあげるよう、端末のハードウェアはできるだけ一般的な仕様に基づくものとしたが、一部に専用化するための改造を行った。

3.1 ハードウェアの専用化

キーボードは、旧端末のティーチング機能をそのまま生かすと同時に、標準的要素を取り込むべきファンクションキーの多様化(12種類)と、キー配列の再設計を行った。

表示画面についても操作性の面から角度調整が上下、左右共に自由に行えるように設計した。また、表示色は人間工学的に見て目に疲れない程度の鮮明度に限定した。画面の枠組は、グッドデザイン賞を受けたものを改造し顧客のロゴをマークしている。

端末設置台は入出力機器および障害切換えスイッチを含む専用台として、とくに営



写真 1 専用端末

Photo. 1 The dedicated terminal

業所内の設置環境を意識して設計した。

プリント・システムは、基幹業務での大量な帳票出力に対処するため高速性が要求されるが、コスト対比により 330 行/分の高速プリンタを採用した。また、グラフおよび表等を出力するプリンタは、表示色が画面色と変わらないこと、表示速度が速いこと、各種のグラフの表示が自由に行えることを目標に実現している。

3.2 ソフトウェアの専用化

後述する経営戦略システムの実現のため、ソフトウェアを次のように専用化した。

大量の検索情報および数百種類のグラフ情報を処理するため、処理効率の向上が要求される。

その実現手法として第 1 にデータ圧縮技法を採用した。データ圧縮は、①ホストシステムとの通信プロトコルにおいては PTP-BATCH ハンドラの改造により通信効率の向上で実現した、②端末システム内のファイル構造はすべてデータ圧縮形式を採用し、ディスク容量とアクセス効率を上げている、③ CPU から端末への情報出力においては、圧縮データを通常データへ変換するハンドラを設計した。

その結果、現状ではディスクの使用率を 60% (120 MB) に抑えることができ、将来のシステム拡張にも対応できるようになった。

第 2 の実現手法として、通信プロトコルの伝送制御手順の効率化のために一般的に使用されている BTAM を採用せず、PTP-BATCH 手順を開発した。この手順の特徴は、①余分な通信制御コードの送信を一切除去する半二重方式の手順であること、②公衆回線を意識した回線切断のアルゴリズムが、ユーザ・アプリケーションに完全に依存できることがあげられる。このため通信時間の短縮、ホスト・システムとのリンクの操作の容易性が実現できた。

第 3 の実現手法として、グラフィック処理能内を向上させるため基本 OS 環境およびシステム環境を一部改造し、数種のグラフィック・モードを追加した。これらによってグラフィック処理能力は向上し、グラフ表示による情報検索システムの目的に適合させることができた。

なお、上記変換アルゴリズムは、図 1 のごとく基本的に二つのフェーズより構成されており、第 1 のフェーズではレコード・セパレータを、第 2 のフェーズではフィールド・セパレータを判別し圧縮データを処理する。レコード・セパレータには、出力キャリッジ制御コードが付随しており同時にその変換も行う。

4. 開発プロジェクト

プロジェクトはハードウェアおよび基本ソフトウェア、応用ソフトウェア別に組織体制および組織間インタフェース・チャンネルを設定し、担当範囲および責任体制を明確にした。ハードウェアおよび基本ソフトウェアの開発は、製造担当部門としての本部組織が担当し応用ソフトウェアの開発は、現業部門として現場システム組織が担当することとした。

前記の体制のもとに 60 年 10 月より本格的に開発を進め、61 年度に全国営業所への納入が完了している。ハードウェアおよび基本ソフトウェアの開発は、本部と現場部門を併せ約 10 名の直接担当要員にて 61 年 4 月まで作業を実施、応用ソフトウェア部

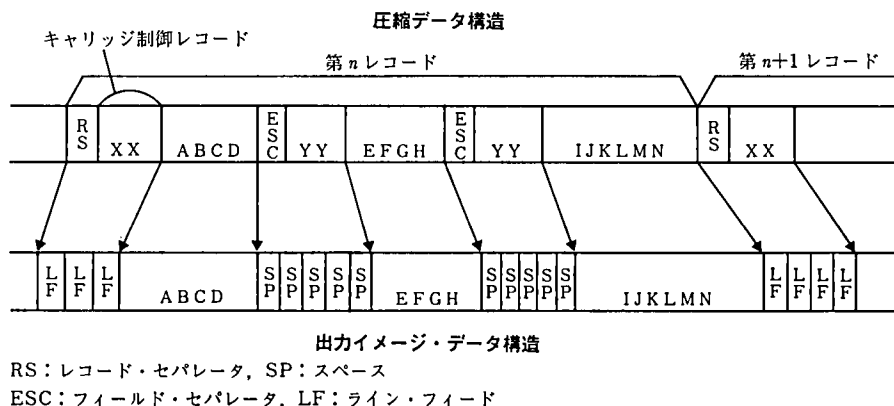


図1 変換アルゴリズム
 Fig.1 Translation algorithm

門は顧客および現場部門にて開発分担を行い、約8名の直接担当により61年7月まで開発を進めた。ただし、グラフ検索システムの拡張開発は顧客側にて現在も開発進行中である。

納入結果としては、営業所の管理者および担当者の評価は高く、ホスト・システムへの情報検索件数も日々増加している状況である。とくに優れている評価項目は端末の操作性および有効性であり、当初の設計目標は達成されたものとみなしてよいと思われる。

5. 処理システムの開発

処理システムは機能別に処理ユニットに分割し、各ユニットの結合を階層化する方向で構築した(図2参照)。各処理ユニットは、基本的に動的にリンクされる処理プログラムとなるが、利点としてメモリの有効利用およびシステム構造に影響を与えることなく開発や修正が可能となることがあげられる。それによって本システムでは、情報検索グラフ・システムの表示グラフごとに処理プログラムを分割することができ、新規要求に対してはその処理ユニットの開発と登録を行えば使用できる。また、処理ユニット別に、機能と効率を意識した開発用言語の選択も可能となる。

第2の利点としては、障害時の対応が迅速に行えることが挙げられる。これは階層化によって、その処理ユニットの位置付けが明確となり操作上のリスタート・ポイントが明らかになったためである。

反面難点としては、各処理ユニット切り換え時にオーバーヘッドが生じ、部分的に効率が低下することがあげられる。基本処理システムの構造およびリンクの形態を図2に示す。

5.1 汎用画面処理の実現

基幹業務のエントリ処理および各種メニュー処理は、端末システムの画面より営業所担当操作員により入力される。その画面イメージは共通要素の多い比較的統一されたものと、独自のイメージに大別される。前者は汎用画面として登録される。このた

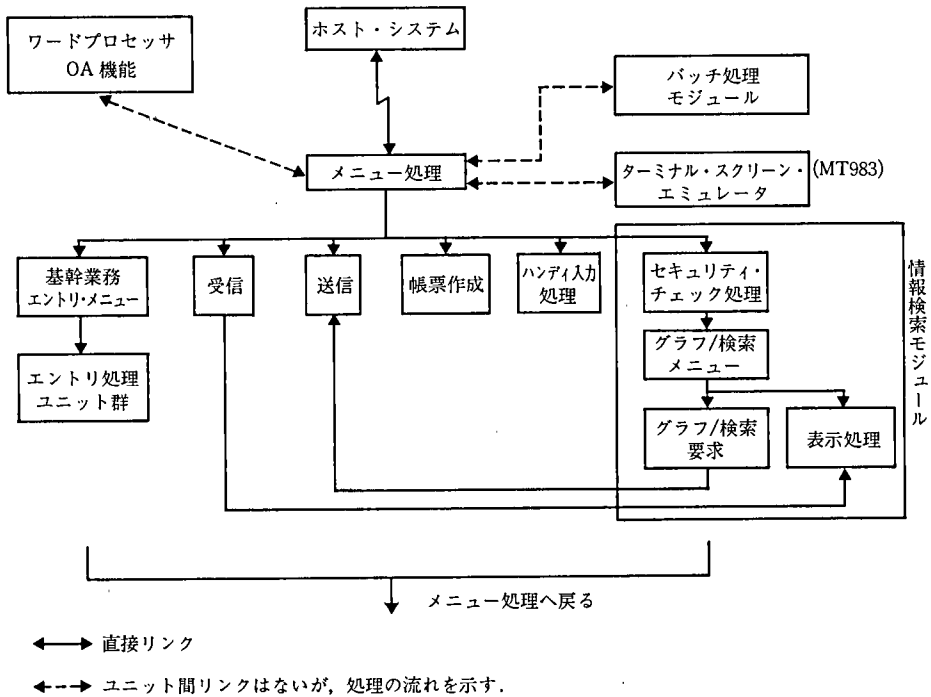


図 2 処理システム構成図
 Fig.2 The construction of transaction system

めに画面の各種属性をパラメタとして入力することによって、自動的に画面を生成する処理モジュールを開発した。

この方式の利点は、①入力項目の基本検査(数値・範囲)が属性の入力により決定でき、プログラムの修正・追加を必要としない、②新規画面および修正画面が簡単な操作で作成でき、処理プログラムの変更を必要としない等である。新たな管理情報の出力に伴い、グラフ/検索システムのエントリ画面の新規追加は必然的に行われるため、汎用画面処理は現状および将来共々ますますその有用性を高めている。

5.2 情報検索システムの構成

処理モジュールの作成に当たり、設計上留意した要素は第1に情報の機密性の保持であった。経営戦略管理情報は基本的にはホスト・システムのデータベースを加工・編集して得られるが、その情報のすべてが企業機密に属する。そのため2段階のセキュリティ・チェックを設計時点で検討し採用した。

第1段階は端末システムにおけるもので、ホスト情報を要求する入力時点で行う入力資格者のチェックである。実際の入力チェックは、まず職種コードおよびシークレット番号によって検索担当者のレベルを知り、その検索可能な範囲を決定する。次にその検索可能範囲内の項目のみを表示する。

次にホスト・システムとのリンク確立時点には、通信プロトコルにもたせた相手識別機能によって不正な接続形態を排除する。実際の相手識別は端末アドレスおよび管

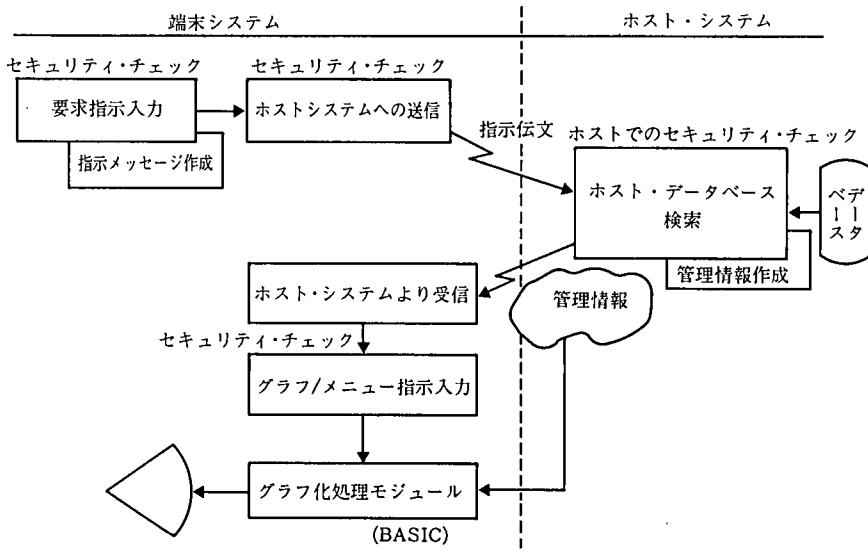


図 3 管理情報検索システムの仕組み

Fig. 3 The construction of management information inquiry system

業所コードの組み合わせで実現している。

第2段階はホスト・システム内でのセキュリティ・チェックである。不正伝文の入力をOSレベルおよびコントロール・システム・モジュールにおけるチェックによって防止している。

情報検索の方法は図3の情報フローによるが、併せてグラフ化をローカル・システム内で実現している。

5.3 経営戦略的管理情報検索システムの実現例

操作員が情報検索グラフ・システムを基本メニュー画面より選択入力した場合、図4(a)のメニューが表示される。検索者の職業レベルを表す職種コード2桁とシークレットNo4桁の入力を行う。なお、シークレットNoは登録されているもののみ有効であり、その入力フィールドは保護されているため第3者が知ることはできない。

職種とシークレットNoが有効であれば、その職種が検索できるメニューとして図4(b)が表示される。なお営業所コードと処理日付は、当日の初期設定されたものが表示される。入力誤りのあるものは、画面の右上にメッセージを表示する。

処理ステップは、画面下位のF1からF12までのファンクションキーで指示するようになっている。入力画面に共通して表示されているVER.001は、画面および処理ソフトウェアのバージョンNoであり、改善等新旧の変更時の混乱を未然に防ぐためのもので、各営業所ごとのバージョン管理を可能にしている。

次に、現在必要としている管理情報を01から12の範囲で選択入力する。図5は、01の営業進捗管理情報を選択した場合のサブメニュー画面の例である。サブメニューでは各種のデータ選択ができ、メニュー画面に表示されている番号を入力することになっている。たとえば88を入力した場合は、営業進捗管理データ選択エントリ画面が次に表示される。

営業所コード=100 処理日付=63/03/31

=====

グラフ(メニュー)の入力 Ver. 001

=====

職種【02】 シークレットNO【 】

[F1] 選択画面 [F2] 職種入力 [F3] [F4] [F5] [F6]
 [F7] アップ [F8] ダウン [F9] [F10] [F11] [F12] 終了

初期メニュー

(a)初期メニュー

営業所コード=100 処理日付=63/03/31

=====

職種=02 グラフ(メニュー)入力 Ver. 001

=====

- | | |
|-------------|----------|
| 01. 営業進歩管理 | 12. 在庫管理 |
| 02. | |
| 03. 商品管理 | |
| 04. 商品分析 | |
| 06. 得意先管理 | |
| 07. 得意先分析 | |
| 08. 売掛金管理 | |
| 09. 営業所利益診断 | |

番号を選んで下さい【 】

[F1] 選択画面 [F2] 職種入力 [F3] [F4] [F5] [F6]
 [F7] アップ [F8] ダウン [F9] [F10] [F11] [F12] 終了

(b)検索可能項目のメニュー

図 4 グラフ・メニュー

Fig. 4 The graph menu

```

=====
営業所コード=100 処理日付=63/03/31
=====
職種=02          グラフ(メニュー)入力          Ver. 001
=====
    
```

- 88. 営業進歩管理データ選択
- 99. 売上推移データ選択

番号を選んで下さい[]

```

[F1] 選択画面 [F2] 職種入力 [F3]      [F4]      [F5]      [F6]
[F7] アップ  [F8] ダウン [F9]      [F10]     [F11]     [F12] 終了
    
```

図 5 サブメニュー画面

Fig. 5 The sub graph menu

63年03月31日現在 *営業進歩管理表
加工品 万円単位 目標標準達成100.0%

	部	目標	当月実績	目標%	前年実績	前年%	売上予測	目標%	前年%	1日当り売上		
										目標	実績	前年
長嶋 一史	1	100	90	90	110	110	120	120	109	4.5	4.1	5.0
原 辰夫	2	70	75	107	80	114	90	129	113	3.2	3.4	3.6
清原 和彦	1	90	90	100	70	78	80	89	114	4.1	4.1	3.2
掛布 雅広	3	100	70	70	80	80	90	90	113	3.2	5.5	3.6
高橋 慶夫	2	130	100	77	100	100	85	65	85	4.5	4.5	3.9
合計		3,000	3,210	107	2,950	98	3,050	102	103	136.4	145.9	134.0

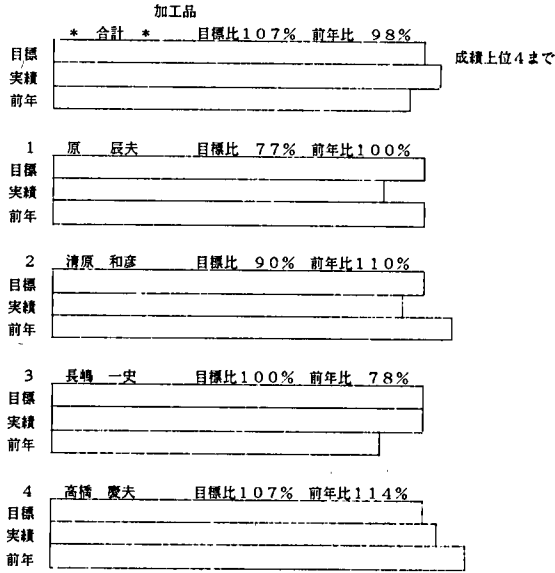
図 6 表の例

Fig. 6 The sample of table

データ選択画面は、必要管理情報の期間と営業所の範囲(指定営業所情報、グループ営業所情報、全社)等を指定する画面であり、ホスト・システムに用意されているデータベースの抽出範囲を指定するものである。

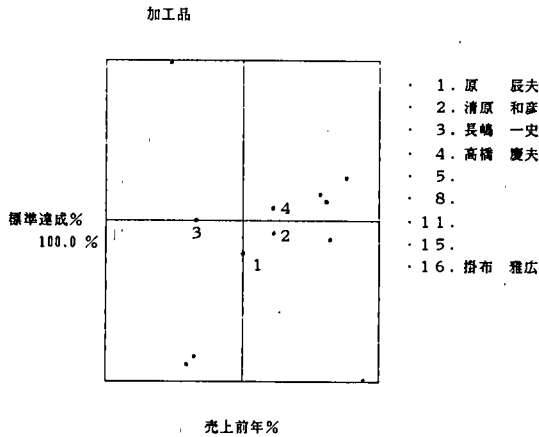
データ選択エントリを完了すると、図3によるホスト・システムへの端末入力情報

63年3月31日現在



(a)売上推移ラップグラフ

63年3月31日現在



(b)売上進歩グリッド図

図7 グラフの例

Fig.7 The sample of graph

の送信が行われる。検索が完了し端末システムへのグラフ情報が準備されると、ホスト・システムより自動 CALL にて回線が接続され、操作員は必要時点で指示入力によりホスト情報を端末システム内に取り込む。

その後は自動的に図6の表および、グラフ表示指示入力により図7のグラフが表示される。また必要であれば、表およびグラフのハードコピーをOAプリンタに出力することも可能である。

図6は、営業所内の各担当者または管理者別の営業達成進捗度を表にしたものであり、図7はそれを2種類のグラフで示した例である。

以上の例で見ると、メニュー画面を介して段階的に詳細な情報に落とし込んでいく方式を取ることによって、セキュリティの段階的保護を実現している。また、現在検索可能情報の組み合わせは数百種類に達している。とくに、最近では経営戦略を目的とした高度な情報のニーズが増加しており、検索画面も日々増加している状況である。

6. 通信システムの実現手法

6.1 通信システムの構成

ネットワーク・システムの拡張拡大に伴い、通信システムもまた多機能でしかも高い品質を要求される。本システム（図8参照）では、ホスト・システム対端末間リンクは現在公衆回線網を利用しているため、低速低品質が避けられない。

そこで、次の改善を予定しているところである。それらは、①現行の回線制御手順専用 PTP-BATCH 手順を、より品質の高い専用 BSC 手順へ切り換える、②公衆回線網を一部 DDX-P 網へ切り換えコスト・品質・速度を改善する、などである。さらに MCA リンクの採用検討中であるが、当面実現の予定はない。

通信システムのボードはオプションであり、最大4ポートまで任意の組み合わせが可能となっている。通信プロトコルは、ハンドラにより処理されるが、実際のプロトコル制御は送受信処理ユニットにて行い、各種コマンドおよび属性パラメータをハンドラに与えることにより実現している。

ハンドラは、端末の初期設定によりスーパーバイザ・コールされる。またターミナル・スクリーン・エミュレータとして、MT 983 エミュレータがある。このエミュレータは、リンク可能な通信手順が POLL-SEL 方式および HDLC 方式手順のため、両手

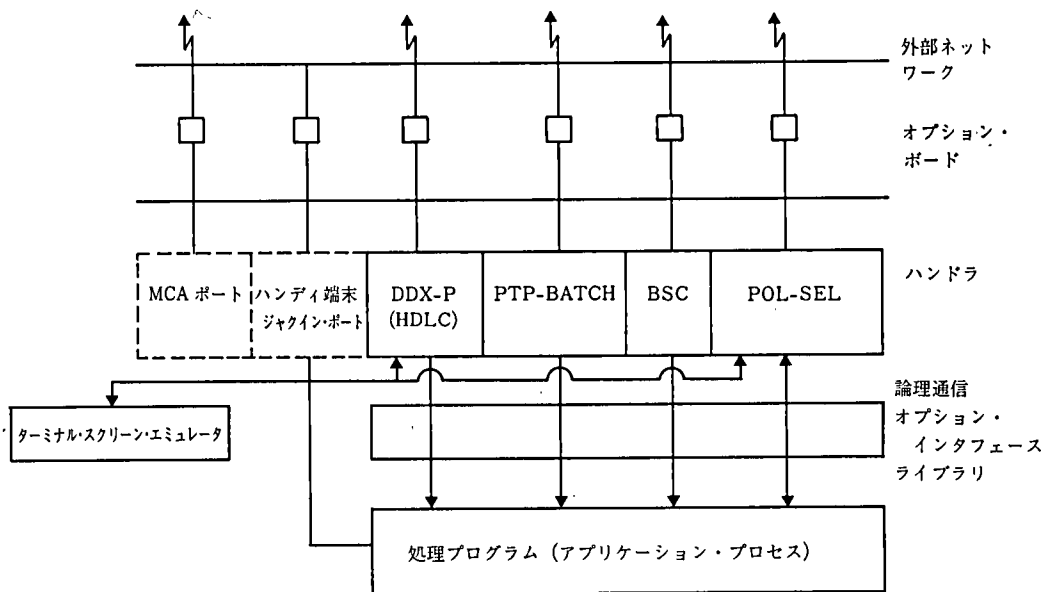


図 8 通信システム構成図

Fig. 8 Construction of communication system

順共搭載を予定しているところである。

ハンドラと処理プログラム間には、オプションとして論理通信インタフェースが介在し相互の指示と応答の制御を行っている。

6.2 ネットワーク体系の考え方

伊藤ハム(株)における総合ネットワーク体系(図9参照)は、従来より一貫した思想のもとに構築されており、西宮本社のホスト UNISYS B 7900 システム内での各処理層と東京、豊橋、九州における拠点としての UNISYS B 1900 中型ホストシステムの処理層、および営業所システム端末の各処理層は OSI 標準に基き、層別に標準化されている。したがって、ある処理の部分的システムの変更は、ネットワークを通じて対応する層への影響に限定され、限界はあるが比較的变化に対応しやすい体系となっている。

OSI 標準	B7900	B1900	端末(IE600)
アプリケーション	アプリケーション・システム		アプリケーション
プレゼンテーション			送受信処理
セッション	コントロール・システム	コントロール・システム	
トランスポート			
ネットワーク	GEMCOS	SMCS	
データリンク	NDL	NDL	ハンドラ
フィジカル	NSP	MLC	ポート

GEMCOS: メッセージ制御汎用ソフト
SMCS: 同上

NDL: 通信制御プログラム生成言語
NSP: 物理的通信制御装置
MLC: 同上

図9 ネットワーク体系図
Fig.9 Network system

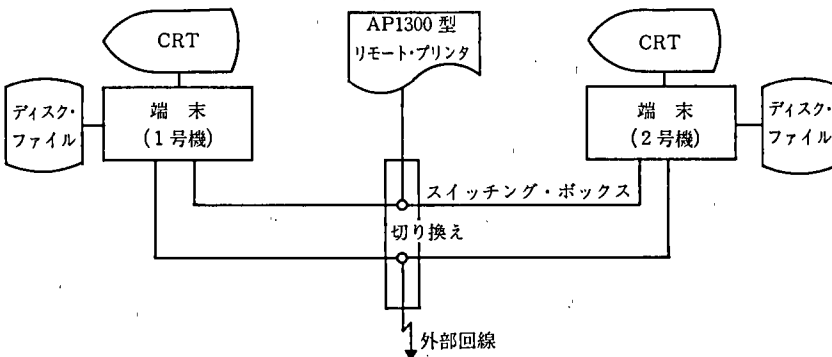


図10 障害対策構成図
Fig.10 Fault tolerant counter-measure of the terminal system

7. 端末システムの障害対策

営業所システム内における端末は、大部分複数台数(2~3台)の構成であり、障害発生時には故障端末を正常端末に瞬時に切り換え、システム全体の停止を避けている。

スイッチング・ボックスは基本的にはハードウェア機構のみで構成されており、機能としては本体および端末接続入出力機器の切り換えが可能であり、図10の構成図のごとく本体の切り換えは通信回線のスワッピングで行う仕組みになっている。したがって1号機および2号機共に同時処理が可能であり、ディスク・ファイルの分散によりホスト・システムとの交信は、適時外部回線を切り換えることによって二重に活用することができる。

8. おわりに

本稿では、専用端末とその上でのアプリケーションの開発事例を紹介した。システムの設計段階よりエンドユーザ・ニーズを意識したハードウェアとソフトウェアの体系が要求され、いかに顧客ニーズを取り入れ分析し実現するかが最大の課題であった。専用端末開発のむずかしさは、第1の点はニーズの反映と同時に将来を見通した標準化とのトレードオフであり、第2の点は直接製造部門とのセマンティック・ギャップに始まる納期等のずれが発生しがちであることにある。

その意味では当プロジェクトは比較的短期間に開発を遂行し、優れた評価を得ており一応成功したものと見てよいだろう。

反省点としては、ユーザ・ニーズを完全に実現するに至らず将来構想に含めていること、および資源とコストに関する開発見積りと実績値の間の差異がまだまだ大きいことが挙げられる。

最後に本プロジェクト遂行に当たり、関係各位の多大な努力および協力があったことを申し添え感謝の意を表したい。

執筆者紹介 堀 籠 順之助(Junnosuke Horigome)

昭和19年生。43年大阪教育大学教育学部数学科卒業。同年日本ユニシス(株)(当時パロース(株))入社。計算機システムエンジニアとしてシステム開発に従事。その後オンライン・ネットワーク部門の設計開発およびニューメディアの調査・研究に従事するかたわら、端末およびホストシステム設計開発に従事。現在、大阪システムサポートセンター所属。近畿技術士会情報処理部門(KICK)会員。



花王(株)における人事システム専用端末

Dedicated Terminal for Personnel Information System in KAO Corp.

南 瀬 末 彦

要 約 花王(株)における新人事情報システムは、UNISYS VIDEOTEX1100*(NAPLPS**)を中核とした先進的システムである。各事務所の端末機から、社員の出勤時刻や夏休みの予定を入力したり、個人の財形貯蓄や自社持ち株の現在額、過去の給与額などを閲覧できる。このような個人向けの人事情報を社員自身が簡単に操作して得られるように、専用端末機が開発された。

本稿は、この専用端末の機能について紹介する。

Abstract The new personnel information system in KAO Corp. is an advanced one which is composed of UNISYS VIDEOTEX1100 (NAPLPS) as a host system and the dedicated terminals distributed to every offices. It offers the various services to the members of the company, for example, their inputting everyday attendance records and submitting summer vacation notices, and obtaining information on the amount of their treasury stocks and past salaries by it.

A new dedicated terminal has been needed for this system, because every members of the company can easily operate the machines on receiving various services.

This report introduces the function and specification of the terminal.

1. は じ め に

企業環境の変化およびテクノロジーの進歩により、先進ユーザでは人事情報システムの再構築を推進中である。筆者が、システム構築に参加した花王(株)におけるシステムは、このような動きを示す好例であり、人事情報業務の省略化、効率化を狙いとするものである。この新システム開発の一環として、全従業員が直接操作できるビデオテックス (NAPLPS) 対応の専用端末 (通称パーソナル端末) が開発された。

本稿では、この専用端末の持つ機能、および構造についての特徴を紹介する。

2. 人事情報システム概要

花王(株)における新人事情報システムは、ホストに UNISYS2200/200D を使用し、各事務所にビデオテックス VIDEOTEX 1100 で稼働する専用端末を配置した構成となっている (図1)。

このシステムは、一般従業員の人事に関する社則や福利厚生等の諸制度を従業員一人一人に身近なものにし、有効活用できるよう情報の提供を行うものである。

* VIDEOTEX 1100 : 文字・図形情報をデータベース化し、ビデオテックス端末と会話処理を行い、必要な時に必要な情報を提供するシリーズ 1100/2200 のソフトウェア・パッケージ。

** NAPLPS : North American Presentation Level Protocol Syntax. 米国とカナダが 1982 年に採択したビデオテックス (双方向文字図形情報システム) の統一方式。

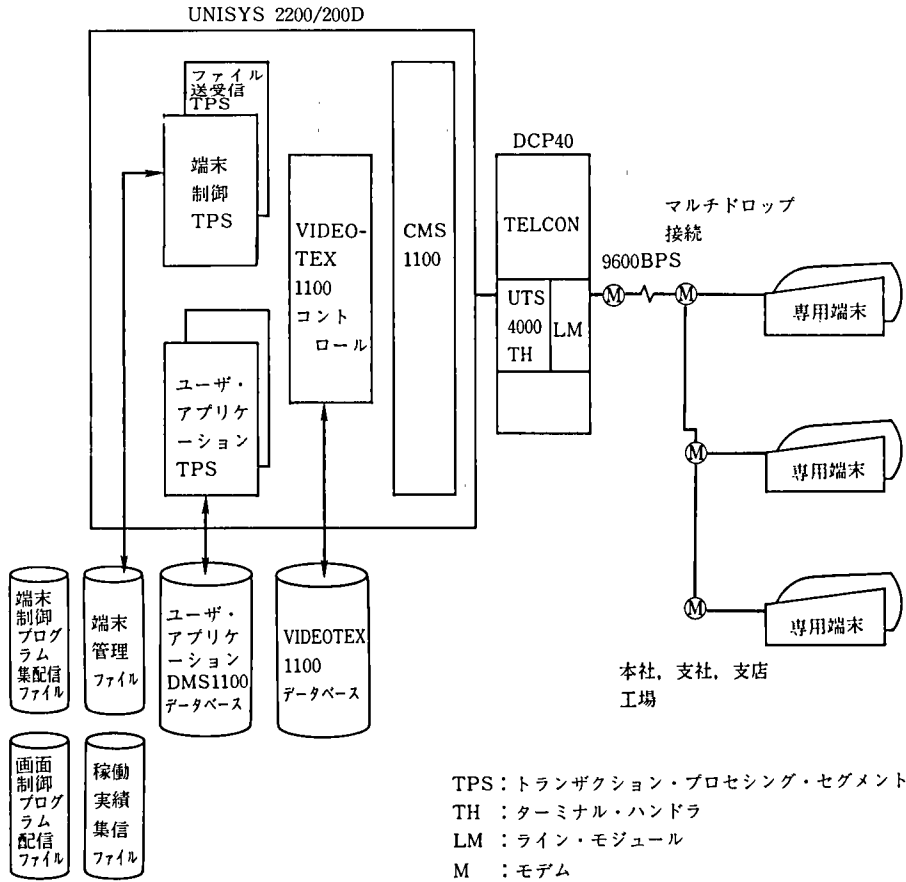


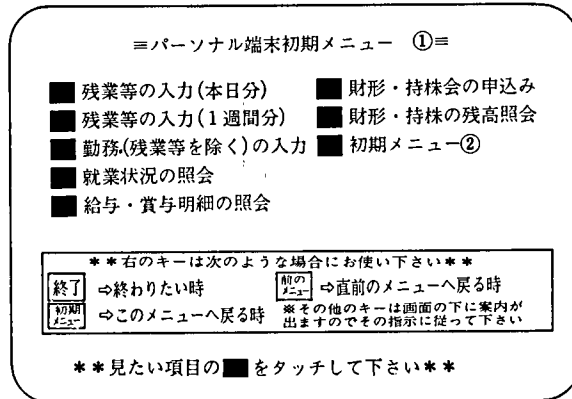
図 1 システムの構成

Fig.1 System configuration

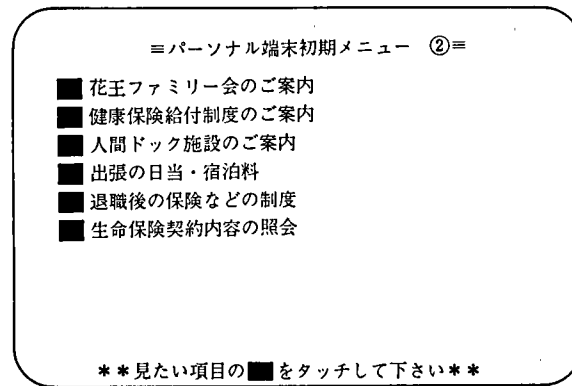
特徴は、個人に関する諸情報、たとえば給与をはじめ賞与の照会、財形、持ち株の申し込みと残高照会、さらに勤労データ（残業・休日出勤・休暇取得等）の入力と照会が、図 2 に示すような専用端末のメニュー画面から自由にできることである。この専用端末は、簡便な操作で誰もが使用できるようにフルキーボードを廃除し、画面に触れて指示できるタッチパネルと必要最小限のキーボードを装備している。また、多くの人が利用するために個人情報の機密保護が必要であり、磁気カード（通称パーソナル・カード）を使用するとともに暗証入力方式を採用している。

すべての情報の表示は、VIDEOTEX 1100 の画面を使用しており、回線経由で専用端末に伝送される。画面のデータ量が多いため、少しでもレスポンス・タイムを軽減できるよう、転送不要な画面は専用端末のハードディスク上に常駐させている。また、画面制御プログラムは、ホスト側で集中管理されていて、変更があれば専用端末の起動時に自動的に転送される仕組みになっている。

また、専用端末は専任オペレータがいないため、無人運転を行う必要がある。無人運転を行うために専用端末内に週間タイマー内蔵の電源コントローラがあり、自動的に電源のオン、オフを行う。また、ホストからのコマンドにより電源のオン、オフ時



(a) 人事システム専用端末の初期メニュー (その1)



(b) 人事システム専用端末の初期メニュー (その2)

図 2 サービス・メニュー

Fig.2 Service menu

間設定および強制電源オフも行える。

端末側では使用状況把握のため稼働ログを記録しており、ホストコマンドによりホスト側へ転送して一括して分析できる。

専用端末のソフトウェア構成を図3に、その外観図を図4に示す。

3. 専用端末に要求される機能

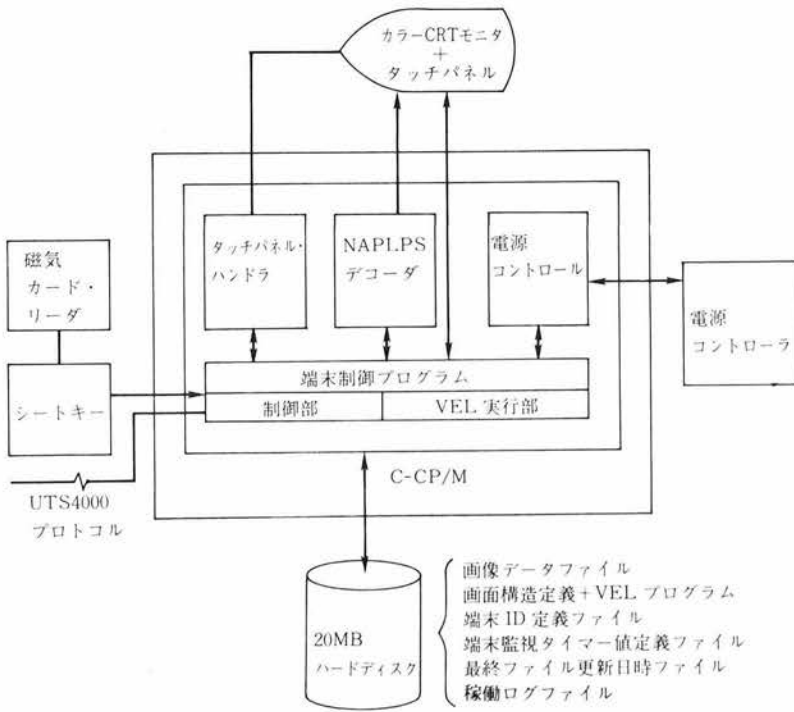
前述のシステム概要を満たすべく、人事情報システム専用端末に要求される機能は以下の通りである。

3.1 入力部

- 1) 一般従業員が使用するため、入力操作は簡単なものであること（一般的なフルキーボードではキーの数が多く使用しづらいので、それに替わるものを考える）。
- 2) セキュリティ確保ができるよう、磁気カードを使用できること。

3.2 表示部

- 1) 多色表示ができること。
- 2) レスポンス・タイムが2～5秒位であること（ただし、回線とホストシステム処



VEL: Videotex Easy Language, 画面等の簡易制御プログラムで端末制御プログラムのもとで動作する。

図 3 専用端末のシステム構成

Fig. 3 System structure of the terminal

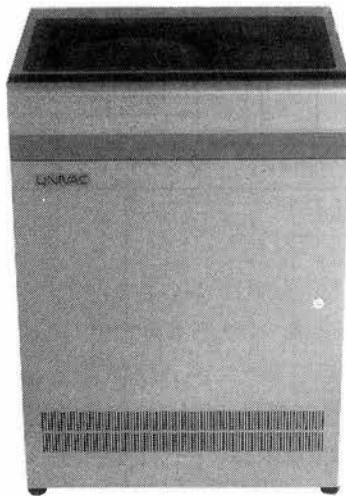


図 4 専用端末外観図

Fig. 4 External view of the terminal

理に要する時間も含める)。

- 3) 画面は個人の情報も表示されるため、操作者以外から見えにくい構造であること。

3.3 電源制御部

- 1) 端末専任のオペレータがないため、無人運用ができること。

3.4 筐体

- 1) 操作は立った姿勢でできること。
- 2) 会社(花王(株))のイメージに合った色合いにすること。
- 3) 外形寸法は、できるだけコンパクトであること。

4. 専用端末のハードウェア構成

本端末のハードウェア構成は、図5に示すようにCPU装置にDS7を使用し、DS7のオプション・ボードに各ハードウェアを接続し、これらを同一筐体内にすべて収めたものである。DS7は、FDD(フロッピー・ディスク・ドライブ)1台、HDD(ハード・ディスク・ドライブ-20MB)1台の構成である。

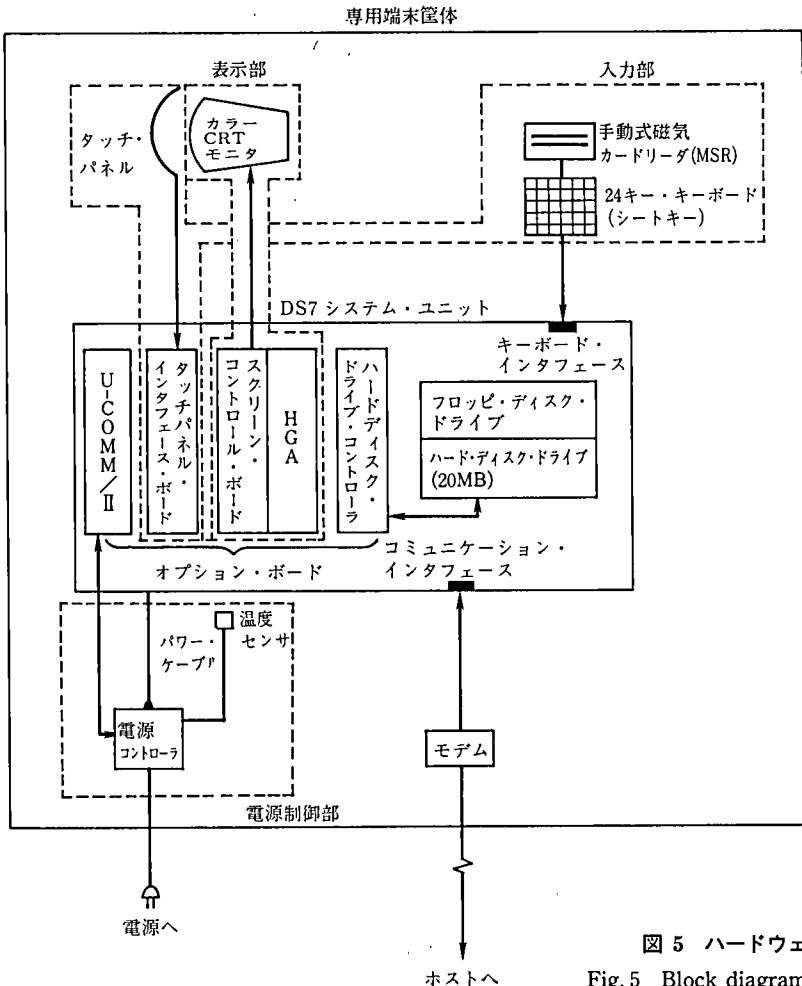


図5 ハードウェア構成

Fig.5 Block diagram of terminal

オプション・ボードとして、HDD を制御する HDD・コントローラ、タッチパネルを制御するタッチパネル・インタフェース・ボード、画面表示を制御するスクリーン・コントロール・ボード、高速のイメージ処理を行う HGA (High speed Graphic Adapter)、そして電源コントローラを制御する U-COMM/II (RS 232 C インタフェース) ボードを内蔵している。

入力装置には、タッチパネル、24 キー・キーボード (以後シートキーと呼ぶ) および手動式 Magnetic Stripe Reader (以後、磁気カードリーダーと呼ぶ) を使用しているため簡単に入力できる。

表示部は、カラー CRT モニタ (以後モニタと呼ぶ) と、前述のオプション・ボードであるディスプレイ・コントロール・ボードおよび HGA ボードから構成され、多色表示と高速イメージ処理を行う。

電源制御部には専用の電源コントローラを設置し、一日一回任意の時間に電源のオン、オフをタイマーで制御することにより無人運転を可能にした。筐体は、これらの機器を一つに収めたもので、保守性と人間工学的な観点からの操作性とを考慮したデザインとなっている。

次に、個々の構成部についての詳細説明を述べる。

4.1 入力部

入力部には、操作の簡便化およびセキュリティ確保のために、タッチパネル、シートキー、および磁気ストライプリーダーを使用している。これらの入力装置の詳細を次に述べる。

4.1.1 タッチパネル

タッチパネルはモニタの管面に装着されていて、タッチパネルの上面を指先等にて押すことにより、X 軸、Y 軸の座標が出力され、CRT 管理上のどの位置を指示したか感知できる装置である。したがって、画面上の選択したい項目の部分を直接触れるだけでその項目 (位置) が入力できるので、キーボード操作に不慣れな不特定多数のエンド・ユーザが容易に使用できるようになり、また操作性が向上する。

タッチパネルのセンス方式として、以下の五つの方式が考案されている。

- 1) 抵抗シート方式 (デジタル・タイプ、アナログ・タイプ)
- 2) 静電容量方式
- 3) 電磁誘導方式
- 4) 光学方式
- 5) 音響方式

この中で現在商品化されているものは、1)の抵抗シート方式、2)の静電容量方式、および4)の光学方式である。静電容量方式は、商品化されたばかりで、あまり実績がないため信頼性が乏しい。光学方式は、センサ部分が CRT 管面より少し離れているため、また抵抗方式のデジタル・タイプは、形状が平面型であり CRT モニタの周辺部で密着しないため、視差が生じやすい。とくに本専用端末のように、水平状態で使用する場合には、視差が大きくなり誤操作しやすくなる。

したがって、専用端末で採用したのは、この3方式でなく抵抗シート方式のアナログ・タイプを採用した。

この方式の特徴は次のとおりである。

- 1) 分解能が高く (X 軸-256, Y 軸-256), 表示パターン設計の自由度がある。
- 2) 指・棒等, 押圧があれば入力操作ができる。
- 3) CRT とほぼ同じ曲率なので CRT 管面に装着した場合のずれがなくなる。
- 4) 項目を選択する場合, 3) により視差が少なく選択ミスが低減する。
- 5) 他方式に比べ, 比較的, 低価格にて入手できる。

抵抗シート方式 (アナログ・タイプ) のタッチパネルは, ポリエステル・フィルムおよびガラス上に透明な抵抗膜を蒸着し, その 2 枚を微小な点状のスペーサではさみ, 張り合わせたものである (図 6)。この抵抗膜の X 軸と Y 軸に, それぞれ電圧を加えておくと, 上部からの指押圧等で機械的に接触し, そこで生じた電圧差がオプションボードのタッチパネル・インタフェース・ボードに入力される。この電圧差は, インタフェース・ボードで AD (Analog to Digital) 変換され 256×256 の精度をもつ座標値となる。ブロック図を図 7 に示す。

また, タッチパネルを CRT に取り付ける場合には取り付け誤差が生じたり, また長い間使用していると経年変化により出力データが変化する可能性がある。これらの誤差をカバーするため, 専用端末ではタッチパネルの出力データをプログラムで補正し,

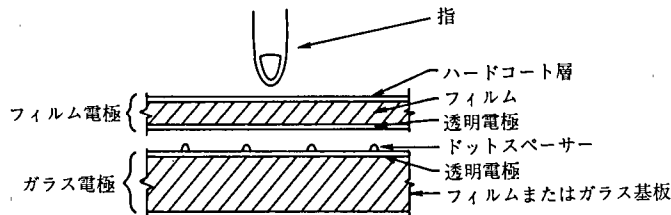


図 6 抵抗シート方式

Fig. 6 Resistance sheet method

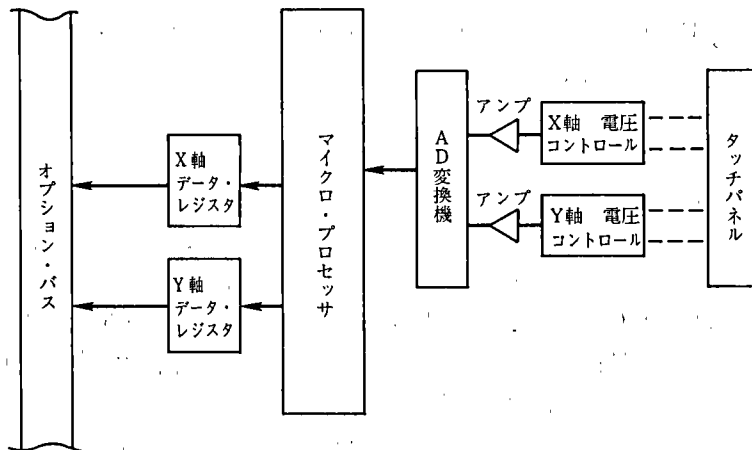


図 7 タッチパネル・インタフェース・ボードのブロック図

Fig. 7 Block diagram of touchpanel interface board

常に正しいデータが入力できるような考慮もしている。

4.1.2 シートキーと磁気カードリーダー

シートキーは簡易キーボードで、24個のキーが装備されている。このキーボードは、数字の入力、初期メニューの選定等、入力が頻繁に行われる項目のキーが配置されている。また、磁気カードリーダーは磁気カードを読み取る装置であり、個人情報の機密を保護するために使用される。

シートキーと磁気カードリーダーは、キーボード拡張インタフェース・ボード（以後、拡張ボードと呼ぶ）によりコントロールされる。また、拡張ボードはDS7のキーボード・インタフェースに接続されて、シートキーおよび磁気カードリーダーそれぞれのコードをDS7に入力する。シートキーのコードは、0～9までの数字とPF1～PF13までのファンクション・コードで構成される（図8）。そしてキーのもつ意味を自由に設定できる。なお、本専用端末での実際のキー配列は図9のようになっている。

磁気カードリーダーの仕様は、読み取りカード記録密度210BPI、読み取り速度3～40インチ/sec、記録文字数72文字以下である。

4.2 表示部

表示部は、多色表示（4096色中16色）とイメージ処理の高速化を可能にするためモニタとスクリーン・コントロール・ボードおよびHGAボードから構成される。

4.2.1 モニタ

モニタは多色で、かつ鮮明な絵を表示するために、1120×720ドットの高分解像度アナログ・モニタを採用した。

モニタは、立って操作したとき見やすくするためと、逆に操作者以外から見えにくくするために表示面を水平に配置した。このように配置すると、通常のモニタ（DS7のモニタ等）では色ずれの弊害がでる。これは、次に述べるモニタの動作原理より起因している。モニタの動作原理は、熱励起され、放出された電子ビームを偏向コイルに流す偏向電流の磁界によってCRT管面を走査させるもので、その原理上外部磁界の影響を強く受ける。

PF1	PF2	PF3	7	8	9
PF4	PF5	PF6	4	5	6
PF7	PF8	PF9	1	2	3
PF10	PF11	PF12	0	.	PF13

図8 シートキー・コード

Fig. 8 Sheet key code

前の メニュー	初期 メニュー		7	8	9
前の ページ	次の ページ		4	5	6
		訂正	1	2	3
OK		終了	0	.	

図9 シートキー印刷内容

Fig. 9 Printed characters on sheet key

外部磁界の一つとして地磁気があり、地磁気は図 10 に示すように地球が一つの磁石として考えられ、それぞれの地域および設置方向により、異なる影響をモニタに与える。したがって、モニタには色純化磁石やコンバーゼンス・ヨークなどの装着部品を取り付け、それぞれの地磁気による影響の補正を行っている。この補正は、通常は机上で使用されるような置き方（管面を垂直に置く）で調整されており、管面を水平に置いた場合は、管面の周辺部に色ずれが起こり実用上問題となる。

したがって、モニタの電源投入時に消磁を行うとともに、水平に置いた場合の調整を製造メーカーで行い、地磁気による影響を極小化して色ずれを防止した。

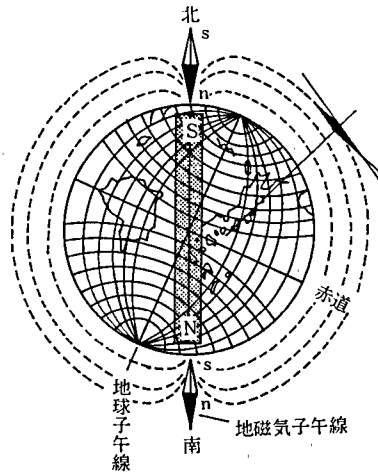


図 10 地 磁 気

Fig. 10 Terrestrial magnetism

4.2.2 スクリーン・コントロール

スクリーン・コントロールは、表示タイミングとキャラクタ表示および多色表示の制御を行う。

本ボードと標準 DS 7 のスクリーン・コントロール・ボードと異なるところは、多色表示のためにパレット IC を装備していることである。このパレット IC は、色の選択を行う IC(Integrated Circuit)で、本ボードの場合 4096 色中 16 色の選択が可能である。

4.2.3 H G A

HGA はイメージ処理を行うボードで、高性能なグラフィック・コプロセッサ（インテル社 82786）を使用し高速度の処理を実現するとともに、ハードウェアで文字フォントの実数倍拡大表示のための拡大補助ロジックを追加し高速化を図った。この補助ロジックは、16 ビットデータを 8~32 ビットの任意のデータ長に拡大または縮小するための回路であり、一次元の変換しかできないが、CPU のビット操作によるプログラム変換に比較して数倍のパフォーマンスを得ることができる。

最近のグラフィック・コントローラには文字の拡大縮小機能を有する物も出現しているが、倍率の制限や縮小アルゴリズムの違い等により、繊細な文字を必要とする本人事システムのようなアプリケーションを満足させる LSI はまだ少なく、コスト的に

問題があるため上述の補助ロジックを考案した。

これの改善により、標準 DS 7 のイメージ/グラフ・アダプタ・ボードに比較して文字の等倍表示で 18.5~25 倍、文字の拡大表示で 7~12 倍、文字以外の描画で 3.5~44 倍のスピードが得られた(表 1)。また、HGA を使用することで専用端末で利用する NAPLPS 画面の表示速度は、回線伝送時間を除いて 2 秒から 6 秒であり、実用上問題のない処理スピードを実現できた。

表 1 グラフィックス・ベンチマーク・テスト

Table 1 Graphics benchmark test

単位：秒

テスト・プログラム	HGA	DS 7 標準 スクリーン・ボード
① 画面の直線分割と塗りつぶし	1.1 (11.0)	12.1
② 画面の直線分割とハッチ・パターンによる塗りつぶし	2.7 (4.5)	12.2
③ 楕円の描画	5.6 (3.7)	20.7
④ 八分円の描画	.2 (3.5)	.7
⑤ 楕円内の塗りつぶし	255.4 (1.9)	475.5
⑥ 楕円内のハッチ・パターンによる塗りつぶし	255.4 (1.9)	475.7
⑦ 四辺形領域の塗りつぶし	8.5 (43.9)	373.2
⑧ 円の描画	1 (12.3)	12.3
⑨ 円内部の塗りつぶし	105.5 (20.7)	2183.3
⑩ 円内部のハッチ・パターンによる塗りつぶし	250.1 (8.7)	2186.1
⑪ 半角文字列表示 (1 倍)	1.4 (18.5)	25.9
⑫ 全角文字列表示 (1 倍)	1 (25.1)	25.1
⑬ 半角文字列表示 (1.3 倍)	4.7 (11.9)	56.1
⑭ 全角文字列表示 (1.3 倍)	3.9 (13.6)	53.2
⑮ 半角文字列表示 (2 倍)	2.2 (7.2)	16
⑯ 全角文字列表示 (2 倍)	2.1 (7.5)	15.8

() 内の数値は、標準スクリーン・ボードを 1 とした時の処理スピードである。

4.3 電源制御部

専用端末は、専任オペレータがいない場所に設置されるため、無人運転を可能にする必要がある。このため、次の五つの機能をもつ電源コントローラを開発した。

- 1) タイマーによる一日一回の電源オン、オフ(ただし、一週間単位)
- 2) 曜日ごとに異なる電源オン、オフ時間の設定および変更
- 3) 異常温度時の警告または電源オフ
- 4) 電池による停電時のタイマーの保持
- 5) ホストからのコマンドによるタイマー値の設定および電源のオフ

これらの機能を用いて、週単位でホストよりタイマー時間を変更し専用端末の自動電源オン、オフを可能にし、また異常温度においては専用端末内部の温度が摂氏 50 度で警告音を発生させ、60 度で電源オフとする。

このようにホストよりの管理と専用端末自身の安全機構により、無人運転を可能としている。

4.4 筐体

筐体は各装置とモデム(構内モデム)一台を一つにまとめて収める金物であり、同社指定のカラーリング・コンパクト性、および従業員が立ったままの状態を使いやすいように考慮した。

デザインの特徴は、従業員全員が使用することを考え、通常の情報機器のイメージを脱した親しみの持てるものとした。筐体は板金主体であるが、前方に大きな丸味をもった部位を設け、単なる箱にならないよう考慮した。

配色は前面扉とトップカバーをパールグレーに、フレーム、リアカバーをチャコールブラウンに塗装して逆L字のラインを強調し、オリジナリティを出すとともに、磁気カードリーダーとブルーグリーンのラインをアクセントラインとした。

モニタとシートは、見やすさ、使いやすさを考慮し7.5度傾斜している。入力部の高さは、身長の高い人から低い人まで操作できるよう、入力部の中央で850 mmとしている。また、操作時に身体が専用端末に触れるまでに近づいた場合のことを考慮に入れ、つま先が専用端末に当たらないようにスペースを確保してある。

シートキーのデザインは、薄暗い場所でも見やすいよう黒の背景に白い文字にしてある。印刷の文字は、限られた範囲内で文字を大きくするために、文字数は漢字2文字カタカナで4文字以内となる。

表面シートは操作感を重視し、0.3 mm厚ポリウレタンシートを採用して適度なクリック感をもたせている。シートの表面は梨地タイプで反射を防ぎ、離面印刷であるため印刷がはがれないようになっている。

磁気カードリーダー部は、スライド式の使い勝手を優先して、カードがスムーズに挿入でき、スライドしやすく溝を切っている。材質は、カードを傷つけることがないように黒色ベークライトを使用している。磁気カードリーダーは右手操作を基本とし、左から右の方向に読ませる。このため、機器の右側に壁などがあつた場合、カードを読ませた後カードが壁にあたって傷がつかないように磁気カードリーダーのヘッドは中心から50 mm左にずらし、読み取り後のスペースを確保している。

5. 仕様

専用端末の仕様を機能別に表2に示す。

6. 製品の評価

人事情報システム専用端末を開発するために、各種の構成部品を検討した結果、要求にあつた端末を開発することができた。

しかしながら、今後、情報量の増大とともに処理速度の高速化、見やすさや操作性の向上等の要求の高まりが予想される。これらの要求に対応するために具体的な検討事項として、以下のものが考えられる。

- 1) 個人情報をご個人で保持できるよう、磁気カードに替わりICカードを採用する。ICカードを使用することにより専用端末以外の端末でも、個人情報を参照することが可能となる。
- 2) 画面をより自然画に近づける。表示色を16色以上に多色化(256色, 512色…)し、使用者に自然画に近い画像を提供する。
- 3) 回線および表示速度の高速化。利用者への応答時間を短縮する。
- 4) 画面の鮮明化。現タッチパネルは、画面の鮮明度を落としているので透明度の高いタッチパネルを採用して、鮮明な画面を提供する。

表2 専用端末仕様
Table 2 Terminal specifications

項 目	仕 様			
C P U 部 (D S L)	プロセッサ	16ビット・マイクロ・プロセッサ (i80286:8MHz)		
	主記憶 (RAM)	2 MB (メモリ・ディスク:768KB)		
	補助記憶装置 (内蔵型)	ドライブ A	5.25インチ・ディスク 640KB(2DD), 1M/1.2MB(2HD) 共用	
		ドライブ B	20MB ハードディスク	
	漢字フォント	非漢字, JIS第1水準および第2水準漢字		
	インタフェース	鍵盤 (磁気カード読取装置 付加可能) RS-232C (UTSプロトコル)		
スロットル接続品	HGA U-COMM/II (RS-232C 電源コントローラ用) タッチパネル・インタフェース			
表 示 部	画面サイズ	14インチ		
	表示寸法	水平250±5mm, 垂直171±5mm		
	表示色	アナログ RGB 4096色中16色		
	解像度	1120 (横) × 720 (縦) ドット		
	走査周波数	水平27.4725KHz, 垂直68.77Hz (インタレース方式)		
	ドットクロック	40MHz		
入 力 部	タッチパネル	アナログ抵抗シート方式, 分解能256×256		
	磁気カードリーダー	磁気ストライプ読取装置, JIS C9561 II型で記録された 磁気ストライプ付きプラスチック・カードの読取可能		
	シートキー	10キー+PFキー, 全24キー		
電 源 制 御 部	電源コントローラ	週間タイマー設定可能, 各種機能あり		
		外部温度センサ	50°C±3°C	アラーム音にて警告
			60°C±3°C	電源スイッチ切断
内部温度センサ	55°C	電源スイッチ切断		
電 源	消費電力	動作中	330W	
		非動作中	30W	
	入力電圧	100V±10% (単相 3線式)		
	周波数	50/60Hz±2%		
	接地	第1種または第3種接地		
	ケーブル長	250cm		
温・湿度許容範囲	温度	10°C~34°C		
	湿度	44%~86%		
電波障害自主規制 (VCCI*)	第1種10dB緩和レベル適応			
外寸 (cm)	60.0 (W) × 51.0 (D) × 90.0 (H)			
重量	80.3kg			

* VCCI : Voluntary Control Council for Interference by Data Processing Equipment and Electronic Office Machines (情報処理装置等電波障害自主規制協議会)

- 5) ホストコマンドにより電源をオンすることにより、内部タイマーによる時間の管理を不要とする。

7. おわりに

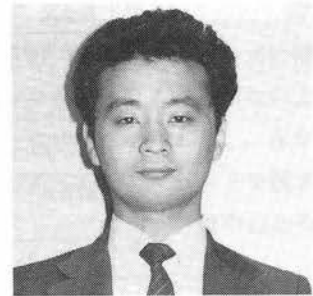
専用端末を採用した人事情報システムは、新しい試みである。今後このようなシステムは、ペーパレス化の推進、事務の合理化、タイムリーな情報の提供等の要請に応えるため他社においても採用されよう。このシステムの要は、専用端末であると言っても過言ではなく、そのより良い操作性が決め手となろう。

今後もユーザの意見を可能な限り反映し、トータルなシステムとして最も使いやすい専用端末を目指して努力していきたい。

本端末の開発に際し、ご助言いただいた花王(株)「新人事事務システム・プロジェクト」の方々に感謝の意を表したい。

執筆者紹介 南 瀬 末 彦 (Suehiko Minase)

昭和34年生、55年国立熊本電波工業高等専門学校電波通信学科卒業。同年日本ユニシス(株)(当時日本ユニバック(株))入社、現在ワークステーション開発部応用機器開発課所属、マイクロプロダクトの開発に従事。



UNISYS DS 7用プリンタでのバーコード印字

Bar-Codes Printing of UNISYS DS 7 Workstation Printer

北 崎 勲

要 約 DS 7に接続されるプリンタ (0469-01) にバーコード印字機能を追加した。

各種バーコードの規格に合ったバーコードを効率良く印字するためには、プリンタの選定や制御方法、バーのドット幅等検討し実証を得るべき事項が多々ある。

本稿では、この作業内容、とくに各種バーコード印字パターンの最適化を中心に報告する。

Abstract Bar-code printing capability has been added on DS 7 dot line printer (TYPE 0469-01). In order to realize bar-code printing in correspondence with various bar-code specifications (2 of 5, Code 39, NW-7 and JAN), few important points were investigated about such as printer selection, print control method and dot width of bar (space) element.

This paper describes mainly the method to form optimum dot patterns for various bar-codes on the 0469 printer.

1. はじめに

近年 FA 関係において、部品管理、工程管理のためのデータ入力手段として、バーコードを利用する動きが活発化してきている。これは、バーコード入力を行うことによって、キーボード入力による煩わしきや入力ミスがなくなるという点にある。

当社においても FA 関係を中心として、DS 7 に対してバーコード・リーダ、バーコード・プリンタを接続することが必須となり、今回バーコード・プリンタとして0469-01型プリンタにバーコード印字機能を付加する作業を行った。

本稿では、各種バーコード規格に対する印字パターンの最適化を中心に作業内容を紹介する。

2. 開発対象の選択

バーコード・プリンタに対する開発目標および条件は以下の通りであり、バーコード印字機能を付加する対象プリンタの選択とバーコード機能を制御する方法について検討を加えた。

- 1) DS 7 の I/O として接続され、バーコードの入った伝票をプリントできること。
- 2) バーコード種類は、CODE-39, NW-7, JAN, 2 OF 5 が取り扱い可能なこと。
- 3) 複写紙付連続帳票印字が可能なこと。

2.1 プリンタの選択

DS 7 に接続されるプリンタは次の 3 種類であり、この中からバーコード印字機能を付加する対象プリンタの選択を行った。

- 1) 0473-00 プリンタ (ドット・シリアル・プリンタ)……縦 24 ピンの印字ヘッドを左右に移動して印字するタイプのプリンタである。印字密度：180 DPI (Dot Per

Inch), 印字速度: 漢字 56 CPS (Character Per Second) である。

- 2) 0469-01 プリンタ (ドット・ライン・プリンタ)……横 1 列に間隔を置いて並んだ 91 本の印字ピンを左右に動かして、横 1 ドット・ライン単位の印字を行うタイプのプリンタである。印字密度: 180 DPI, 印字速度: 漢字 140 LPM (Line Per Minute) である。
- 3) 0474-00 プリンタ (レーザ・ビーム・プリンタ)……電子写真方式のノンインパクト・タイプのページ・プリンタである。印字密度: 240 DPI, 印字速度: 約 6 枚/分である。

まず機能上の必要条件として、伝票発行可能なこと、タックフォーム (シール)・プリント可能なことが挙げられる。0474-00 プリンタは、レーザ・ビーム・プリンタであるために、カットシートしか扱えないため対象外になる。

次に 0473-00 プリンタと、0469-01 プリンタを比較してみる。バーコードを印字する際に重要な点は、縦野線の印字位置精度である。この点での仕様の比較は表 1 の通りである。縦野線を印字した場合には、0473-00 プリンタはドット・シリアル・プリンタであるために、各行ごとに印字ずれを起こす。この印字ずれを最小とするため、印字時の印字ヘッド動作方向を一方向に統一する、いわゆる片方向印字という印字方法をとるのだが、この場合印字は復路では行われなため印字速度が遅くなる (両方向印字の約 1.5 倍の時間を要す)。一方、0469-01 プリンタは、ドット・ライン・プリンタであるため、行間での印字ずれの発生が少ない。

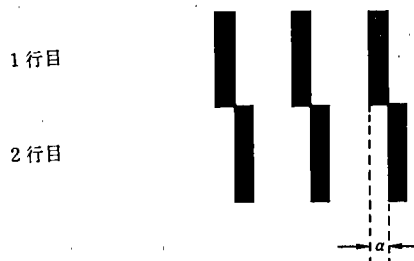
印字速度の比較は、印字方式が異なるため印字内容により変化し、0473-00 プリンタの片方向印字の場合と較べて、漢字 20 桁印字で 2.4 倍、漢字 30 桁印字で 3.2 倍も 0469-01 プリンタの方が速い。またタックフォームの印字は、0473-00、0469-01 両プリンタ共可能であるが、0473-00 プリンタは 0469-01 プリンタに較べて、ラベルの剝離性やラベルと台紙との段差等に対する条件がむずかしい。

さらに 0473-00 プリンタは、プリンタ内ファームウェアに新機能を追加できるだけの空領域がなく、バーコード印字機能をプリンタ内ファームウェアに持たせようとした場合は、0473-00 プリンタでは実現不可能である。この点については 0469-01 プリン

表 1 印字精度比較

Table 1 Horizontal print positioning

	0473-00	0469-01
縦野線印字位置 精度 (α)	0.2mm以下 (片方向印字時)	0.13mm以下



タは、プリンタ内ファームウェアに新機能を追加できる空領域がある。

以上の結果、バーコード・プリンタとしての開発対象は、0469-01 プリンタとなった。

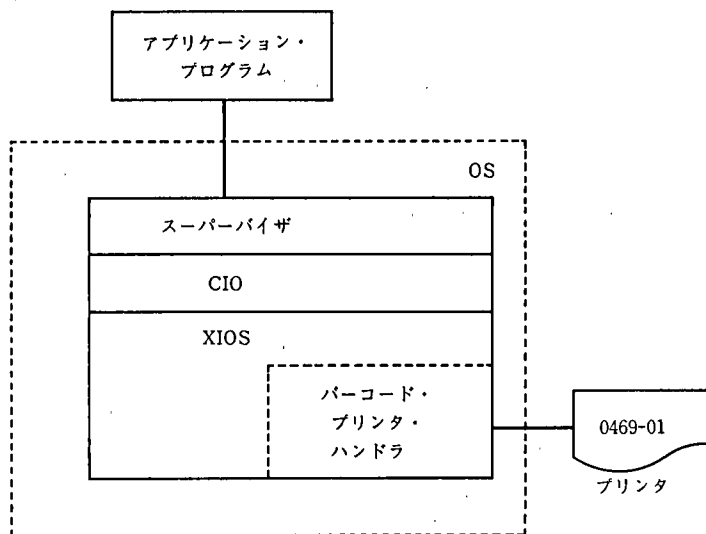
2.2 バーコード印字の制御方法

バーコード印字の制御方法として、以下の2種類の方法が考えられる。

- 1) DS7内プリンタ・ハンドラにてバーコードのイメージ・データを発生し、プリンタではイメージ・プリントを実行する。
- 2) DS7からバーコード印字用コマンドを受け取ると、プリンタ内ファームウェアは、コマンドを解析してバーコードの印字を実行する。

まず 1) について考察すると、DS7内のハンドラとOSとの関係は図1の通りで、ハンドラはOSの一部としてメモリに常駐し動作する。プリンタへ出力するデータは、すべてこのハンドラによってデータのチェックが行われる。ハンドラはバーコード印字用コマンドを検出すると、所定のバーコード・イメージ・データに変換してプリンタへ出力する。バーコード・データは、あらかじめバーとスペースの組み合わせを定義したテーブルをハンドラの中に持っておく。

プリンタ・ハンドラのとくに重要な機能は、バーコードの行間接続である。通常の印字では、印字行数は6 LPI (Line Per Inch) または4 LPI にて行われ、実際に1行中に印字できる縦方向ドット幅は24ドットである。6 LPIの場合、1行あたり30ドットに相当する幅があるためバーコードの行間接続を行うためには、1行を2回に分けて印字する必要がある。この際に1回目と2回目の印字での横方向の印字位置を合わせるために、バーコードを何桁目から印字したかをハンドラが知っていなければならない。このためには、ハンドラはプリンタに設定しているパラメータであるレフト(ライト) マージン、水平タブ位置、改行ピッチ等を常に管理しておく必要がある。



CIO : Character I/O (文字入出力)

XIOS : Extended I/O System (拡張入出力システム)

図1 プリンタ・ハンドラとOSの関係 (DS7)

Fig. 1 OS and printer handler (DS7)

一方 2),つまり 0469-01 プリンタ内ファームウェアにバーコード印字機能を付加する方法は、バーコード印字のコマンドをプリンタが受けると、プリンタは所定のバーコード印字パターンに展開して印字バッファに詰め込む。バーコードの行間接続についても、プリンタは実際に印字する装置であるため、何桁目からバーコードを印字するかを記憶しており、まったく問題なく印字できる。

以上の結果から、0469-01 プリンタ内のファームウェア改造による方法をとることとした。

3. バーコード印字パターンの決定

バーコード・プリンタの開発では、以上の結論により 0469-01 プリンタ内ファームウェアを作成することとなったが、このファームウェアの機能仕様においてバーコード印字パターンの決定が最も重要な点であった。

3.1 バーとスペースの印字幅の設定

印字ドット数と印字幅との関係は、印字ピッチ= a 、ピン径= b とした場合に、

$$\text{Max } B(i) = b + a(i-1) \quad (3-1)$$

$$\text{Min } B(i) = \sqrt{b^2 - a^2} + a(i-1) \quad (3-2)$$

$$\text{Max } S(j) = 2a - \sqrt{b^2 - a^2} + (j-1) \quad (3-3)$$

$$\text{Min } S(j) = 2a - b + a(j-1) \quad (3-4)$$

となる。

ここで、 $B(i)$ は i ドット分のバーの幅、 $S(j)$ は j ドット分のスペースの幅を表す。

0469-01 プリンタの印字ピン径と印字ピッチは、それぞれ 0.2 mm, 0.14 mm である。さらにインクのにじみを考慮した場合、ドット単位での最大径は 0.3 mm となる。実際には、上下のドットの交点ではインクのにじみが大きくなり、必ずしも計算値通りにはならないと思われるが、ここではドット径についてはインクのにじみがない場合 0.2 mm, にじみが最大の場合 0.3 mm として検討を進める。

3.2 バー印字幅比率

バーコードは、ワイド、ナローのバーとスペースの組み合わせで表す 2 値レベル・バーコードと、1:2:3:4 種類の幅のバーとスペースの組み合わせで表す 4 値レベル・バーコードがある。バーコード規格と 2 値, 4 値の関係を次に示す。

● 2 値レベル・バーコード

2 of 5	3本の細バーと2本の太バーで1キャラクタを構成する。0~9の数字キャラクタを表現できる。現在はあまり使用されていない。
NW-7	4本のバーとその間の3本のスペースで1キャラクタを構成する。0~9の数字、および特殊記号を表現できる。運輸システム、図書館システム等で主に使用されている。
CODE-39	5本のバーとその間の4本のスペースで1キャラクタを構成する。0~9の数字、A~Zのアルファベットおよび特殊記号を表現できる。工場の部品管理等で広く使用されている。

● 4 値レベル・バーコード

JAN	8桁または13桁の0~9の数字を表現できる。食料品等のPOSシステムを中心とした流通システムに用いられる。
-----	---

まず、2 値レベル・バーコードのバーとスペースの印字幅を決定する。

3.3 2値レベルのバー印字幅

2値レベル・バーコードのバー、およびスペースのナロー対ワイドの比率は、一般にバーコード・リーダ*の性能から

$$\text{ナロー} : \text{ワイド} = 1 : 2 \sim 1 : 3$$

の範囲内であることが望ましいとされている。

市販されているバーコード・リーダの設定しているスレッシュホールド**の値によって左右され、通常1:2の印字幅比率では誤読率が上昇してしまうので、1:2.5~1:3.0を目標値とした。

- 1) ナロー・バー……バーの印字幅は、ナロー・バーの印字幅によって各バーおよびスペースの印字幅を決定できる。バーコード全体の印字幅を極力短くしようとする場合、ナロー・バーの印字は1ドット幅で行えばよいことになるが、この場合のバーの幅が一番細い部分で0.14 mmとなる。

ドットプリンタで印字されたバーコードを読み取る場合、スキャナの分解能***は、0.2 mm程度であり、0.14 mmのバーは分解能以下の太さのため、バーかスペースかを正確に判別できない。ナロー・バーを2ドット幅にすると、バーの幅が一番細い部分で0.28 mmとなり、0.2 mmの分解能のスキャナで判別可能となる。このため、バーコード全体の印字幅を最短にするナロー・バーの印字幅は2ドットが適当となる。

- 2) ワイド・バー……ナロー:ワイドのドット数での比率を仮に1:3に設定した場合、ナロー・バー=2ドットでは、ワイド・バー=6ドットとなる。この時の実際の印字幅の比率は、式(3-1)および(3-2)より以下の通りとなる。

- ① インクのにじみのない場合

$$\text{Max } B(2) : \text{Max } B(6) = 1 : 2.7$$

$$\text{Min } B(2) : \text{Min } B(6) = 1 : 3.0$$

- ② インクのにじみが最大の場合

$$\text{Max } B(2) : \text{Max } B(6) = 1 : 2.3$$

$$\text{Min } B(2) : \text{Min } B(6) = 1 : 2.4$$

インクのにじみが最大の場合、この比率は1:2.5よりも小さくなるが、ワイド・バーを仮に7ドットに設定すると1:3の比率を超えてしまうので、やはりワイド・バー=6ドットが妥当である。

- 3) スペース……ナロー・バーとワイド・バーをそれぞれ2ドット、6ドットとした場合、スペースの幅をこのバーの幅に近い値に設定しようとする、式(3-3)および(3-4)よりナロー・スペース、ワイド・スペースはそれぞれ3ドット、7ドットとなる。

この時の実際のスペースの比率は以下の通りとなる。

*バーコード・リーダ:バーコード・リーダは、スキャナ部、デコーダ部から構成される。スキャナ部からバーとスペースの情報を入力し、デコーダ部ではそれを解析し、情報交換用コードに変換する。

**スレッシュホールド:スキャナ部から入力された情報がナロー・ワイドのどちらに属するかを判断するための境界線。

***分解能:バーコードのデータをスキャナ部がサンプリングする際の1入力単位の直径値。

① インクのにじみのない場合

$$\text{Max } S(3) : \text{Max } S(7) = 1 : 2.3$$

$$\text{Min } S(3) : \text{Min } S(7) = 1 : 2.6$$

② インクのにじみが最大の場合

$$\text{Max } S(3) : \text{Max } S(7) = 1 : 2.9$$

$$\text{Min } S(3) : \text{Min } S(7) = 1 : 3.1$$

インクのにじみが最大の場合、ナロー・スペース：ワイド・スペースの比率が1：3を少し超える部分が出てくるが、ほとんどの場合1：3の比率を超えることがないこと、および他のドット構成がとり得ないことより、ナロー・スペース、ワイド・スペースはそれぞれ3ドット、7ドットに決定する。

実際のバーコードの読み取りテストでの正読率も、このドット構成であればかなり高い値(99.5%)が得られた。したがって、ナロー、ワイドのバーおよびスペースの構成は、以下の通りとなった。

ナロー・バー = 2ドット

ワイド・バー = 6ドット

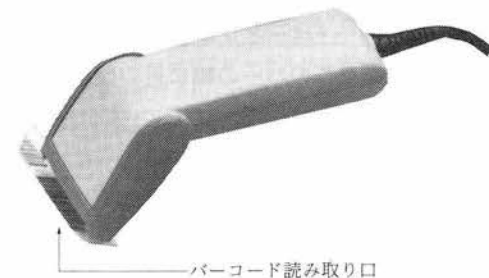
ナロー・スペース = 3ドット

ワイド・スペース = 7ドット

3.4 4値レベルのバー印字幅

JANコードは、先に述べた通り1：2：3：4の4種類の幅のバーおよびスペースで構成し、最小幅を1単位として、これを1モジュールと呼ぶ。1モジュールの幅は、0.33 mmと規定されている。

一般的にこのJANコードの読み取りは、ハンドスキャン・タイプのバーコード・リーダー(図2)が使用されている。ハンドスキャン・タイプの場合、読み取り可能なモジュールの幅は0.8倍(0.26 mm)から1.6倍(0.52 mm)が一般的である。この0.26 mmというのはバーコード・リーダーの分解能からくる制限で、0.52 mmというのはバーコード・リーダーの読み取り口の幅から決まる制限である。したがって0469-01プリンタにて印字するモジュールの幅も、0.26 mm～0.52 mmの範囲内とする必要がある。



バーコードに読み取り口を近接、静止させることで自動的に読み取る。

図2 ハンドスキャン・バーコード・リーダー

Fig.2 Hand scan bar code reader

表2 モジュール幅2ドットでの印字幅および印字幅比較
Table 2 Print width and print width ratio (2 dots/module)

		ドット数	2	4	6	8
印 字 幅	印 字 幅 が 最 大 の 場 合 に じ み	Max $B(i)$ (バー最大幅)	0.34	0.62	0.91	1.19
		Min $B(i)$ (バー最小幅)	0.28	0.57	0.85	1.13
		Max $S(j)$ (スペース最大幅)	0.28	0.56	0.85	1.13
		Min $S(j)$ (スペース最小幅)	0.22	0.51	0.79	1.07
	印 字 幅 が 最 大 の 場 合 に じ み	Max $B(i)$	0.44	0.72	1.01	1.29
		Min $B(i)$	0.41	0.69	0.97	1.25
		Max $S(j)$	0.16	0.44	0.72	1.01
		Min $S(j)$	0.12	0.41	0.69	0.97
印 字 幅 比 率	印 字 幅 が 最 大 の 場 合 に じ み	Max $B(i)$	1	1.8	2.7	3.5
		Min $B(i)$	1	2.0	3.0	4.0
		Max $S(j)$	1	2.0	3.0	4.0
		Min $S(j)$	1	2.3	3.5	4.8
	印 字 幅 が 最 大 の 場 合 に じ み	Max $B(i)$	1	1.6	2.3	2.9
		Min $B(i)$	1	1.7	2.4	3.1
		Max $S(j)$	1	2.8	4.6	6.3
		Min $S(j)$	1	3.3	5.6	7.9

この範囲内のドット数は、バーは2または3ドット、スペースは3ドットである。バーおよびスペースのモジュール幅を2ドット、3ドットにした場合の印字幅並びに各モジュールの2倍、3倍、4倍のドット数の印字幅と印字幅比率を表2,3に示す。

この表からもわかる通り、モジュールの幅が大きくなるほど、規定の印字幅比率に近づいてゆく。モジュールの幅は2ドットよりも3ドット、3ドットよりも4ドットの方が良いということになる。しかしながら、モジュールの幅を0.26mm~0.52mmの範囲内にするためには、バーおよびスペースのモジュールの幅は3ドットを超える大きさは取りえない。

バーおよびスペースのモジュール幅を3ドットにした場合の印字幅比率をみてみると(表3)、インクのにじみが最大の場合作規定の印字幅比率を逸脱しているが、平均的にみるとほぼ基準を満たしていると考えられる。

これより、バーとスペースのモジュール幅を3ドットとする最終的な判断は、バーコード・リーダでの読み取りテストの結果で決めることとした。テスト・プログラムにて印字し、読み取りテストを行った結果として99.4%の正読率を得ることができた。

この読み取りテストで満足のいく正読率を得たため、JANコードのモジュール幅は3ドットと決定した。

3.5 バーコード印字パターン

バーとスペースの印字幅の設定が完了したため、次にはバーとスペースを組み合わせ

表3 モジュール幅3ドットでの印字幅および印字幅比較
Table 3 Print width and print width ratio (3 dots/module)

		ドット数	3	6	9	12
印 字	印 字 幅	Max $B(i)$ (バー最大幅)	0.48	0.91	1.33	1.75
		Min $B(i)$ (バー最小幅)	0.42	0.85	1.27	1.69
		Max $S(j)$ (スペース最大幅)	0.42	0.85	1.27	1.69
		Min $S(j)$ (スペース最小幅)	0.36	0.79	1.21	1.63
幅	が 最大 の 場 合	Max $B(i)$	0.58	1.01	1.43	1.85
		Min $B(i)$	0.55	0.97	1.40	1.82
		Max $S(j)$	0.30	0.72	1.15	1.57
		Min $S(j)$	0.26	0.69	1.11	1.53
印 字 幅 比 率	印 字 幅 比 率	Max $B(i)$	1	1.9	2.8	3.6
		Min $B(i)$	1	2.0	3.0	4.0
		Max $S(j)$	1	2.0	3.0	4.0
		Min $S(j)$	1	2.1	3.3	4.5
	が 最大 の 場 合	Max $B(i)$	1	1.7	2.5	3.2
		Min $B(i)$	1	1.8	2.5	3.3
		Max $S(j)$	1	2.4	3.8	5.2
		Min $S(j)$	1	2.6	4.2	5.8

せ、バーコード印字パターンを構成することとなる。

印字パターンは、各バーコード規格にパターンの構成が定義されているので、これをそのままプリンタ内ファームウェアの中でテーブルとして持てばよい。ここでは、例としてNW-7の印字パターンの構成を表4に、またその印字例を図3に示す。

4. コントロール・コマンドの決定

次にバーコード印字機能を持ったプリンタに、DS 7がどのような制御方法でバーコード印字させるか検討する必要がある。制御方法は、プリンタ・コントロール・コマンドによりプリンタを起動する方法をとった。

まず、バーコード印字の際の必要条件を挙げてみる。

- 1) バーコードの印字高さが可変であること。
 - 2) 同一行内で、バーコードと文字の混在印字が可能であること(図4)。
 - 3) バーコードの下のヒューマン・リーダブル・エリアへの文字がバーコードと同じピッチで、かつバーコードと混在印字可能なこと(図5)。
- 1) のバーコード印字高さについては、以下の2種類の方法が考えられる。

- ① バーコード印字の際にバーコード印字高さを、あらかじめコマンドの中で指定する。
 - ② 行ごとにバーコード印字指定を繰り返し、複数行の印字で高さを調整する。
- 同一行内でバーコードと文字の混在印字を可能とするためには、①のバーコード印

表4 NW-7パターン構成
Table 4 NW-7 code configuration

キャラクタ	B1	S1	B2	S2	B3	S3	B4	キャラクタ	B1	S1	B2	S2	B3	S3	B4
0	0	0	0	0	0	1	1	-	0	0	0	1	1	0	0
1	0	0	0	0	1	1	0	\$	0	0	1	1	0	0	0
2	0	0	0	1	0	0	1	.	1	0	1	0	1	0	0
3	1	1	0	0	0	0	0	/	1	0	1	0	0	0	1
4	0	0	1	0	0	1	0	:	1	0	0	0	1	0	1
5	1	0	0	0	0	1	0	+	0	0	1	0	1	0	1
6	0	1	0	0	0	0	1	a	0	0	1	1	0	1	0
7	0	1	0	0	1	0	0	b	0	1	0	1	0	0	1
8	0	1	1	0	0	0	0	c	0	0	0	1	0	1	1
9	1	0	0	1	0	0	0	d	0	0	0	1	1	1	0

B1, B2, B3, B4がバー・エレメント
S1, S2, S3がスペース・エレメント
0 : NARROW
1 : WIDE

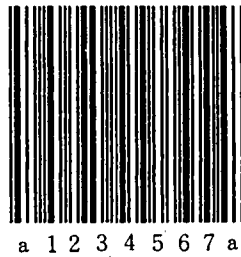


図3 バーコード印字例 (NW-7)
Fig.3 Bar code print sample (NW-7)

字高さを指定する方法はプリンタ内ファームウェアの処理が複雑になり、印字速度が落ちる可能性がある。また、混在印字する文字との位置関係を文字と同様には扱えなくなるのが考えられるため、適当とは言えない。

②の方法ではバーコードの行間接続が、どの改行ピッチ指定のもとでも可能であることが必要となる。これは、バーコードの1行あたりの印字高さを改行ピッチと同一とすることにより解決できる。このような改行ピッチに応じて印字高さを調整するのは、ドット・ライン・プリンタの特長でもある。したがって、バーコードの1行あたりの印字高さは改行ピッチと同一とし、この印字を複数行繰り返すことにより、印字高さを調整する印字方法を取るようになった。

このために必要なコマンドは、以下の3種類である。



図 4 バーコードと文字の混在印字の例
Fig. 4 Bar code and character strings



図 5 バーコード下のヒューマン・リーダブルエリアへの印字例
Fig. 5 Human readable interpretation of the bar code data

- 1) バーコード・データ・セット……バーコードの種類と印字させたいバーコード・データをセットする。プリンタは、メモリ上にセットされたバーコードのイメージを展開する。
- 2) バーコード印字……このコマンドが指定された桁位置からバーコード・イメージを印字バッファに展開する。このコマンドは印字開始命令ではないので、このコマンドの前後でキャラクタコードを受けると、バーコードと同一行への印字が行われることになる。このコマンドを繰り返して指定することにより、バーコードの縦の長さを調整することができる。
- 3) バーコード下の文字印字……バーコード下のヒューマン・リーダブル・エリアへバーコードの文字を印字する。このとき文字の印字ピッチは、バーコードと同一ピッチとする。

この一連の作業でバーコード印字が完了する。

5. おわりに

今回開発された 0469-01 プリンタのバーコード印字機能については、必ずしも最善の方法を取り得たとは言えない。もっと追及すれば性能・価格とも十分満足できる方法を見出す余地がある。たとえば、ドット・インパクト・タイプのプリンタでは、完全なバーコードは印字し得ないが、より性能の高いバーコード・リーダが開発されれば、より印字品質の悪いプリンタを利用しても、高い正読率を得られるようになる。

つまり、バーコード・プリンタを開発する際に、プリンタの印字精度をどこに設定するか判断材料の一つとして、デコーダの性能を含めることが必要である。また、インクリボンの新旧によるドット数の加減調整機能も、場合によっては必要となるであろう。

以上、バーコード・プリンタの開発事例を紹介したが、今後バーコードに関連する開発に従事する方々への参考になれば幸いである。

参考文献 [1] 森 宗正, 「各種バーコードのパーソナル・コンピュータによる印書実験」, 技報, 日本ユニバック(株), No. 15, 1987.

[2] Bar Code Symbology, INTERMEC, 1982.

執筆者紹介 北 崎 勲 (Isao Kitazaki)

昭和 57 年東海大学理学部卒業。同年日本ユニシス(株)
(当時日本ユニバック(株))入社。現在ワークステーション
開発部小型ペリフェラル開発課に所属し、DS 7 用 I/O の開
発に従事。



Ben Shneiderman 著
東 基 衛, 井 関 治 監 訳

ユーザー・インタフェースの設計
—使いやすい対話型システムへの指針—

Designing the User Interface

日経マグロウヒル社, B 5 判, xx+328 pp.,
1987 年, 5,800 円

本書は、ソフトウェアの設計におけるヒューマン・ファクタをはじめ、ユーザー・インタフェースの広い範囲を研究しているメリーランド大学コンピュータ・サイエンス学科の准教授である Ben Shneiderman による著書である。著者は、同大学内にある Center for Automation Research に属する Human-Computer Interaction Laboratory の所長でもある。

現在、人間は何等かの形でコンピュータを操作している。子供達はファミコン等によるコンピュータゲームに熱中し、コンピュータと関係ないと思われる婦人達でさえ、銀行の CD(キャッシュ・ディスク)を操作し、入金・支払・振込み等を行っている。

会社内では近年 OA 化が進み、キーボードアレルギーの人達も、否応無しに端末の操作を求められてきている。本書「ユーザー・インタフェースの設計」は、人間とコンピュータが対話をしながら作業を進めるシステムの設計者・管理者・評価者のために書かれたものである。種々の実験結果や論文により広範囲にまとめられているので、対話型システムを開発、運営している人達には、非常に参考になるであろう。

本書の構成は以下に示す、IV部、11章からなり各章は「はじめに」から始まり、「まとめ」、「研究課題」で終わっている。手取り早く内容を把握したい方は、この「はじめに」と「まとめ」、を読まれることをお勧めする。

第 I 部 ユーザー・インタフェース研究の動機と基礎

第 1 章 対話型ソフトウェアにおけるヒューマン・ファクタ

第 2 章 理論, 原則, 指針

第 II 部 対話形式

第 3 章 メニュー選択システム

第 4 章 コマンド言語

第 5 章 直接操作

第 III 部 ユーザー・インタフェース要素の考察

第 6 章 対話のためのハードウェア装置

第 7 章 応答時間と表示速度

第 8 章 ユーザーへの表現方法 (システム・メッセージ, スクリーン設計, 色, ウィンドウ)

第 9 章 マニュアル, オンライン・ヘルプ, チュートリアル

第 IV 部 評価と影響

第 10 章 対話型システムの設計, テスト, 評価

第 11 章 社会と個人への影響

このように各章のタイトルを一覧しただけで、対話型システムの問題点、および在り方が見えてくると思われる。

本書は、まえがきにも書かれている通り、第 I 部で人間の多様な運動能力や認知能力に対応するためにユーザー・インタフェースの動機付けを振り返り、システムを考える上での理論、原理、ガイドラインが述べられている。2.2.1 項「構文知識」には、【たとえば電子メール・システムを使うとき、段落を終えるのには RETURN キーを押す、文章を終えるには CTRL-D キーを押す、電子メール・システムを終えるのには Q キーを押す、セッションを終えるのには logout と入力する。コンピュータに関する知識のある者は、このような四つの終了方法は理解できるが、初心者やエンド・ユーザーには、根本的にはまったく異なる形式でありながら表面的に似通った命令にとまどってしまう】とある。コンピュータの世界では、常識が常識でないことがある。この章を読むことにより、人間自身を再認識するであろう。

第二部では、現在用いられている対話対法のメニュー選択、空欄記入方式、コマンド言語、直接操作などの基本的な対話手法を述べている。4.2 節「ユーザーの作業を助ける最適な機能」には、【過剰な機能の影響を調べる実験を IBM が行った。……使用されたのは平均 26 種、最大 34 種だった。また XEDIT には 141 個ものコマンドがあるが、もっとも経験のある利用者でさえ、その 4 分の 1 も使用していなかった。】とある。弊社の第 4 世代言語の一つである MAPPER の会話機能も、レベル更新ごとに機能が増え、今では 70 種以

上となっている。

第Ⅲ部は、対話のためのハードウェア装置をはじめ応答時間、表示速度、システム・メッセージ、画面設計、色、ウィンドウ、印刷されたマニュアル、オンライン・ヘルプとチュートリアルなどの重要な項目を扱っている。第6章では、キーボードから始まり、ライトペン、ジョイスティック、マウス、タッチスクリーン、グラフィックス、パッドやトラックボールなど各種入出力装置があげられ、その特徴が説明されている。さらに、音声認識や音声合成の利用、カラーディスプレイ装置についても述べられている。しかし、著者も述べている通り、「データ入力的主要な方法は、今なおキーボードである。」

第7章では、種々の実験結果により、応答時間、表示速度による作業効率について述べられており、システムの設計者は、一読する価値があると思う。端末使用者に、応答時間に対して聞くと、速ければ速いほど良いと、誰もが答えるだろう。しかし、著者によると、「対話のペースが速いと学習が不十分になり、十分理解せずに読み飛ばし、判断が軽率になり、入力ミスが多くなる恐れがある。」と述べている。

第8章では、対話システムが受け入れられるか否かを左右する重要な要因が述べられている。ただ、日本語（漢字）について、書かれていないのは残念である。

第Ⅳ部は、繰り返しの評価/設計手法・テスト・評価・管理手法および社会や個人に与える影響について説明している。

第10章に、ユーザー評価シートが記載されており、簡易言語や第4世代言語の評価に、一部適応できる。

最後に、「人間の多様性（能力・過去の経験・動機・性格・作業スタイル等）は、対話型システムの設計者に対する挑戦ともいえる。」この言葉を受けて、より良いシステムを作成するためにも、ぜひ一読して頂きたい。

(統合 OA システム部 大野 正恒)

★

Donald C. Gause 著
Gerald M. Weinberg 著
木村 泉 訳

ライト、ついてますか
—問題発見の人間学—

共立出版, B 5 判, vii+161 pp., 1987, 1800 円

著者の一人である Weinberg 氏と訳者の東京工業大学教授の木村泉氏の著・訳コンビは、bit 誌連載の「イーグル村通信」をはじめとする数々の著作を通して、情報処理分野に関係する人々に多くの示唆を与えている。原著の鮮やかな問題解析の切り口と滑らかでウィットに豊んだ訳は、多くの読者を魅了する。

本書は、副題「問題発見の人間学」が示すように、問題解決にまつわる人間的側面、とりわけ問題そのものへの接近（アプローチ）に関する態度を扱っている。まず「序文」は、次の4行ではあるが、それが本書の論旨と語り口を明快に表している。

序 文

問題 誰も序文なんか読まない。

解答 序文を第1章と呼んだらいい。

解答によって作り出された新問題 第1章はたいくつだ。

解答 第1章なんかやめて、第2章を第1章と呼んだらいい。

これに引き続いて「第1部 何が問題か?」、「第2部 問題は何なのか?」、「第3部 問題は本当のところ何か?」、「第4部 それは誰の問題か?」、「第5部 それはどこからきたか?」、「第6部 われわれはそれをほんとうに解きたいか?」と、楽しい(!)具体的事例を通して「問題発見」について解析を進めている。

まず、第1部で次のように切り出し、

問題とは、望まれた事柄と認識された事柄の相違である。

第6部で次のように結ぶ。

多くの場合に、問題を解決すること（ないし解消すること）は取るに足らない作業である——ひとたび問題は何であるのかわかってしまえば。

つまり、本書では「問題を発見すること」「問題の真の出所を見届けること」の重要性を説いている。

著者は、

計算機分野は、問題定義に関する教訓の大鉱脈である。(第2部)

として、「問題の理解」の重要性を指摘する。すなわち、

すべての解答は次の問題の出所(第3部)

である。言い換えれば、問題と解答は無限の連鎖である。したがって問題解決とは、

「解いた」問題がよりやっかいさの少ない問題で置き換えられること(第3部)

なのだ。甘い問題把握に基づいた解答は、やっかいな新たな問題を生む。善意が裏目に出た、というのはよくある話である。より良い問題解析は、「適切な」問題の理解に基づく。このための重要な規則を次のように述べている。

キミの問題理解をおじゃんにする原因を三つ考えられないうちは、キミはまだ問題を把握していない。(第3部)

多くの既婚者は、きつと万感をこめてこの言葉の重みを受けとめることでしょう。

つぎに課題となるのが問題を理解し解決する人は誰か、という主体に関する事柄である。一つの問題は、一つの(取り敢えずの)解決を生み、それが次の問題を生む。その人の問題は「解決され」、他の人に投げかけられる。その問題一解決の無限連鎖の中で、問題の大きさは増幅されたり減衰したりする。日常の経験が教えるところでは、多くの重大な問題は減衰している状態の問題を見逃したために生じている。問題の「理解」などより以前の「発見」で多くの重大な問題を防ぐことができる。主体的な問題意識のある人にチョット新しい視点を与えて「発見」を促せばよいのだ。ちょうど、トンネルの出口で「ライトついていますか?」という看板でドライバーのうっかりミスを防ぐように。

すなわち、

もし人々の頭の中のライトがついているなら、ちょっと思い出させてやる方がごちゃごちゃいうより有効なのだ(第4部)

というわけである。なぜならば、

問題の出所はもっともしばしばわれわれ自身の中にある(第5部)

からである。

近年、企業のシステム開発部門においてQCサークル活動が徐々に取り入れられてきているが、この活動の本旨は問題解決能力向上にある。QCサークル活動を通して、職を同じくする人達が「問題と思われる事柄」を論議し、自分自身の問題として把える。ほとんどの問題は、この時点で解決していることになる。「問題解決能力」とは、ウスウスとは改善の要ありと感じている事柄を白日の下に露らし公正に論議を進める力、と言っては言いすぎか?

本書では、いわゆるむずかしい問題について次のように述べている。

伝統的に「問題解決」と呼ばれてきたもの多くは、実はパズル解決である。パズルはむずかしいように設計されているものだ。(第5部)

したがって、多くの問題はそれが何であるかわかってしまえば、解決することは取るに足りないことなのだ。ところで心理学上での「頭の良さ」の定義は、「一度も経験したことのない問題に直面した時に、見通しをもって解決する力」というそうだ。学校のテストはパズルだ。「利口な学生は試験を、出題者の目で「読む」。もし教授が期待する答を与えなかったとしたら、その人は阿呆である。(中略)

明らかに試験問題は、答えの見当をつけるのに予定された試験時間より多くの時間がかかるはずがない。」(第5部)。

一方、実社会では問題は外界で生成されるのだ。期限内に解答に到達するという保障はない。ここでの「頭の良さ」とは、問題一解決の連鎖を的確に把握する能力と意志の力なのだ。そこで著者は問う、「あなたの頭の中のライト、ついていますか?」

「問題発見の人間学」の著者と読者に対しこれ以上ゴチャゴチャ言うことは失礼に当たるというものだろう。それぞれの読者のライトに当てて「深読み」をし、それぞれの立場で発見・啓示を得ることこそ重要であろう。「訳者口上」の一部を紹介して筆を置こう、「だまされたと思って開けてみてください、きつとお得ですよ」。

(ワークステーションシステム部 佐藤 博)

で広く使用されている第4世代言語である。佐々木政和は、UNIX マシン UNISYS 5000/70 上での第4世代言語 LINC の実現の中で、LINC の UNIX マシン上での実現方法について述べている。5000/70 システム上での LINC は、MCP 上の LINC とはまったく異なる構成をとっており、本稿では主な機能を中心に、LINC の UNIX マシン上での実現方法を紹介している。

昨今の大企業は統合ネットワークとして全国規模の営業所システムを展開しているが、各営業所では適切な意思決定が行えるような管理情報が得られなければならない。したがって通常営業所のエンドユーザは、ネットワークを利用したターミナルシステムをユーザ環境に立脚したシステムとして構築することになる。堀籠順之助は、統合ネットワーク指向インテリジェント端末の開発——伊藤ハム(株)における事例の中で、顧客要求を反映した多機能専用端末開発の全般的紹介と将来像について述べている。

南瀬末彦の花王(株)における人事システム専用端末は、花王(株)における新人事システム構築の一環として開発された、全従業員が直接操作できるビデオテックス (NAPLPS) 対応の専用端末について、機能および構造の特徴を紹介している。この VIDEOTEX 1100 で稼働する専用端末からは、個人向けの人事情報を社員自身が簡単に操作し、閲覧することができる。

各種バーコードの規格に合ったバーコードを効率よく印字するためには、プリンタの選定や制御方法、バーのドット幅等検討し実証を得るべき事項が多々ある。北崎勲の UNISYS DS7 用プリンタでのバーコード印字は、この作業内容、とくに各種バーコード印字パターンの最適化を中心に報告している。

☆

▶ 技報編集委員会

委員長 柳生孝昭

副委員長 米口 肇

委員 飯塚伊三雄, 岡井功雄, 佐野和義, 新野清嗣, 鈴木 勲, 内藤 聡, 永田利地, 西原良一, 野本雄一, 藤田康範, 前田英次郎, 村井啓一, 吉兼晴雄, 鷺尾 武, 朝倉文敏, 駒崎洋介

▶ 編集制作担当

技術情報サービス部 青柳幸久, 丹野敬子

● Editorial Board

T. Yagiu (Chairman)

H. Yoneguchi (Vice Chairman)

I. Iizuka, I. Okai, K. Sano, K. Shinno,

I. Suzuki, S. Naitou, T. Nagata,

R. Nishihara, Y. Nomoto, Y. Fujita,

E. Maeda, K. Murai, H. Yoshikane,

T. Washio, F. Asakura, Y. Komazaki

● Publications Staff

Y. Aoyagi, K. Tanno

(Technical Information Services Dept.)

ISSN 0289-6257

技 報

UNISYS TECHNOLOGY REVIEW

Vol. 8 No. 1 (No. 17)

発 行 日	昭和 63 年 5 月 31 日
編 集 人	柳 生 孝 昭
発 行 人	富 田 和 夫
発 行 所	日本ユニシス株式会社 東京都港区赤坂 2-17-51 〒 107 TEL (03) 585-4111 (大代表)
頒 布 価 格	1,500 円
印 刷 所	三美印刷株式会社

禁無断複製転載

© Nihon Unisys, Ltd. 1988

The Power of²



日本ユニバックとパロースが合併し、新しい会社が誕生。

1. (人の力)×(人のチカラ)=The Power of²
2. (企業のカ)×(企業のカ)=The Power of²
3. (日本のネットワーク)×(世界のネットワーク)=The Power of²
4. (アメリカの創造力)×(日本の創造力)=The Power of²

ふたつの力が手を結び、より大きな力を生みだそうという
スローガンを胸に、私たち日本ユニシスは、
コンピュータ業界の新しい力となって、はばたいていくつもりです。

日本ユニシス

UNISYS