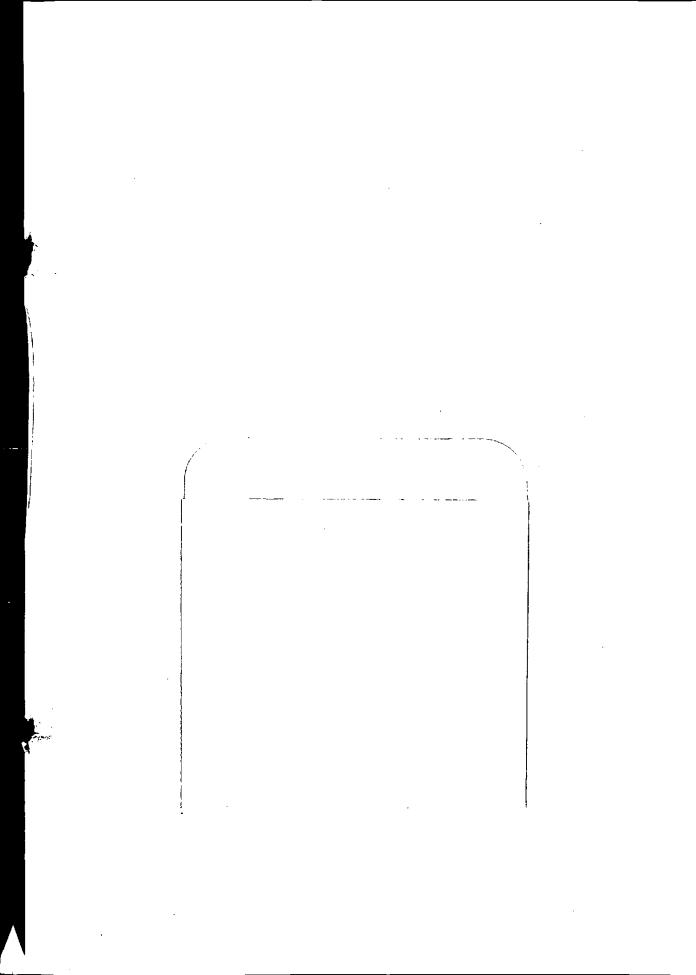
パーソナルコンピュータ利用技術の近未来 — 知識ベースとの融合は何をもたらすか——

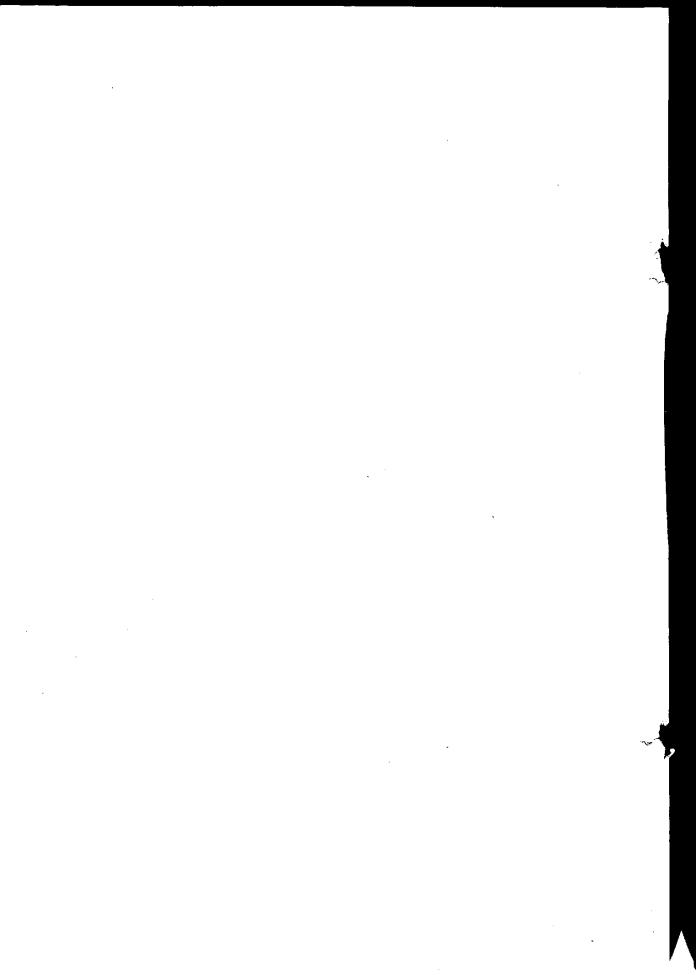
昭和60年3月

JIPDEC

財団法人 日本情報処理開発協会

この報告書は、日本自転車振興会から競輪収益の一部である機械工業振興資金の補助を受けて、昭和59年度に実施した「マイクロコンピュータの応用に関する調査研究」の一環としてとりまとめたものであります。





はじめに

パーソナルコンピュータは、コンピュータパワーを個人として利用できることおよび人間の知的機能の外延としてその機能を増幅するという意味で画期的なものであり、その利用技術の一層の研究が期待されている。

当協会マイクロコンピュータ振興センターでは、昭和56年度に「知的活動に利用する個人用マイクロコンピュータへのニーズと技術」について調査研究を行ったが、当時、最もニーズの高かったデータベースや通信機能は、今やパーソナルコンピュータの基本機能となっており、処理スピードやメモリ容量も当時と比べれば充分満足なものとなっている。

しかしながら、知的利用という面での現在の利用技術は、知的活動過程そのものを支援するものではなく、知的活動のためのインプットやアウトプットの支援にとどまっている。

このため、知的活動により広く、より深く立ち入った利用技術の研究が期待されている。

本年度は、パーソナルコンピュータの知的利用技術に関する最近の動向を示す ことを目的とし、

- ① 機能はどこまで進んでいるか
- ② どう利用されているか
- ③ 知識ベースとの融合は何をもたらすか

の3つの観点から調査研究を行った。

ここに、調査研究の実施にあたりご指導、ご協力いただいた関係各位に厚くお礼申し上げるとともに、本報告書が広くパーソナルコンピュータ利用技術の研究の場において活用され、わが国の産業の一層の発展に寄与することができれば幸いである。

昭和60年3月

マイクロコンピュータ応用技術調査委員会

(順不同,敬称略)

委員長 西 川 禕 一 京都大学工学部電気工学科教授

委員伊藤紘二 東京大学工学部

附属境界領域研究施設助教授

"安西"、古一郎 慶応義塾大学理工学部管理工学科専任講師

大野 侚郎 日本ビジネスオートメーション株式会社

取締役

〃 梶浦信孝 アイ電子測器株式会社 取締役

ロコンピュータ販売部部長

" 永 井 正 武 沖電気工業株式会社OA事業部システム推

進本部システム開発部課長

事務局 (財)日本情報処理開発協会マイクロコンピュータ振興センター

第1編 パーソナルコン	ビュータ,ワークステーションの機能はどこま	
で進んでいるか		1
第1章 現在のパーソ	ナルコンピュータ,ワークステーションの主な	
機能は何か …		1
1.1 高機能パーソ	ナルコンピュータ,ワークステーション誕生の	
背景		1
1.2 多機能パーソ	ナルコンピュータの持つべき機能	2
1.3 高機能パーソ	・ ナルコンピュータとワークステーションとの関	
係		2
1.4 ワークステー	ションとその主な機能	4
第2章 現在どのよう	なワークステーションが開発されているか	7
•	ンピュータとワークステーションの市場動向	7
	ワークステーションの動向	9
	ング・ワークステーションの動向(国内商品)	10
	ング・ワークステーションの動向(国内で入手	
	i品)	11
	ング・ワークステーションの動向(エキスパー	
_· -	lLISPマシン)	12
	ステ-ションの動向	14
	BMOPC77 & J	14
	[T/370概略 ······	15
)ワークステーション	16
)ために, 今何が必要か	18
)脈流を静観せよ	18
O. 1 July 1 HE 24.	- Althor C 10 180 - 24	

3.2 低価格・高機能化できる技	おによる対処 ······ 19	9
3.3 "机"では持ち歩けない …		0
3.4 さらに拡大の動きをみせる:	通信機能の増大とアプリケーショ	
ンウェアの強化	2	0
	要 2	1
3.6 第5世代へ向けての集約化		1
第Ⅱ編 パーソナルコンピュータ, ワ	- クステーションはどう利用され	
ているか		5
第1章 マイクロプロセッサの応用	とワークステーションへの発展 2	5
1.1 計測・制御へのマイクロプ	ロセッサ応用の発展経緯 :2	5
1.1.1 高性能汎用プロセッサ	2	7
1.1.2 高速制御用プロセッサ	2	8
1.1.3 オペレーティング・シ	ステム 2	8
1.1.4 プログラミング言語 …	3	1
1.1.5 データベース		2
1.1.6 図形処理/画像処理・		3
1.1.7 通信システム	3	3
1.1.8 マイコン応用事例 1	3	4
1.1.9. マイコン応用事例 2 …		8
1.2 ワークステーションへの応	用 4	. 1
1. 2. 1 OAシステム	4	4
1.2.2 CAD/CAMシステ	4	6
1.2.3 グラフィック・ディス	プレ装置4	8
	用されているか	
	ズの把握	
	5	
93 オフィスターミナルレーで	のワークステージョン 5	52

2.4 近い将来のOAシステムの構築例	54
2.5 IBMのオフィスシステム・オーバービュー	58
2.6 1980年代の情報処理についての一考察	59
第3章 教育システムでどう利用されているか	61
3.1 現 状	61
3.1.1 教育分野別利用状况	61
3.2 今後の課題	68
3.2.1 普及上の課題・留意点	68
3.2.2 具体例	71
第Ⅱ編 知識ベースとの融合は、ワークステーションに何をもたらすか	75
第1章 知識ベース構築技術はどこまで進んでいるか	75
1.1 現 状	75
1. 1. 1 実 例	75
1.1.2 人工知能の現状	76
1 1.3 知識工学の現状	78
1.1.4 知識ベースの構築	79
1.2 今後の課題	81
1.2.1 知識の次元	81
1.2.2 今後の課題	82
第2章 知識ベースは学習的マン・マシン・インタフェースを可能に	
する	84
2.1 生涯学習を支援する計算機利用システム	84
2.2 学習の対象としての知識の表現	87
2. 2. 1 C F U	89
2. 2. 2 K D U	97
2.3 知識の学習的検索と段階的提示	108
·	

•

.

第	3	章	知的]情報処理システムの問題理解システムへの応用	138
	3.	1	はじ	じめに	138
	3.	2	算数	の文章題の問題解決システム	139
,		3. 2	2. 1	何のためのシステムか	139
		3. 2	. 2	システムの概要	140
		3. 2	. 3	問題文章の意味解析	141
		3. 2	. 4	意味表現の問題表現への変換	142
		3. 2	2. 5	数式への変換と答の導出	145
		3. 2	. 6	システムの評価	145
		3. 2	2. 7	関連研究と今後の課題	147
	3.	3	初等	き物理学の問題解決システム	147
		3. 3	3. 1	何のためのシステムか	147
		3. 3	. 2	システムの概要	148
		3. 3	3. 3	定性的推論と「系」の導出	149
		3. 3		例 題	
		3. 3	5. 5	システムの評価	154
	3.	4	おわ	>りに	154
邦 [V	編	善	支術	トピックス	157
第	1	章	国风	産マイクロプロセッサVシリーズ,	157
第	£ 2	章	オフ	ブジェクトオリエンテッド言語 Small talk	171
第	3	章	知部	畿ベースシステム構築用ソフト「OPS-5」	178
第	4	章	Kn	nowledge Engineering Environment - K E E	186
第	5 5	章	知語	識ベースシステム構築用ソフト「BRAIN」	211

"Computer"という英語をそのまま翻訳すれば、それは"計算する機械あるいは道具"という日本語になる。だから今日、われわれは日常的に"コンピュータ"すなわち"計算機"という用語例になれている。計算あるいは数値計算は最も明確で実用的な知的操作の一つであり、一方計算機は確かに四則演算を繰返して動作する機械である。しかし、コンピュータはもともとより広い範囲の能力を持っている。すなわち、単なる計算機ではなくて、もっと豊かな情報の蓄積と情報の処理の能力を具えているのである。

ディジタル・コンピュータが日本で実際に製作されるようになり、実際に利用されるようになったのは1950年代の終り頃からであるといえるだろう。昭和でいえば30年代に入って間もなくの頃である。その頃からコンピュータに対して"人工頭脳"とか"思考する機械"とかいう呼び名も使われた。人類の技術史のうえで、それまでに様々な道具や機械や装置が発明され利用されてきたのだが、人間の思考活動、精神活動に直接関わるようなものはかつて存在しなかった。だからこそ、まったく新しい範ちゅうに属する機械に対して人々が多くの期待を抱き、時には過大ともいえる夢を描いたとしても無理からぬことであったといえよう。

一方では,人間の思考過程はつまるところすべて 2 値論理の操作に還元されるというように説かれた。それ故, 2 値論理の連鎖で動作するディジタル・コンピュータの機能を拡大していけば,やがて人間に匹敵する能力を持った"頭脳"が実現するはずだというわけである。しかし一方では, 人間の脳細胞の数は $10^{13} \sim 10^{14}$ のオーダだといわれている,それを当時の素子(初めは真空管であったがやがてトランジスタに置き換えられた)で実現するとすれば霞ヶ関ビルの何倍かの大きさになり,その消費電力だけでもとてつもない量になる。第一,それだけの素子数の信頼性を保証することはとても無理である。だから,技術的に人工頭脳を実現できるのはまだまだ先のことである,というようにも説かれた。

1950年代の終り頃といえば、ちょうど人工知能(Artificial Intell-

igence,AI)の研究が始まった頃なのであった。今日では人工頭脳という言葉はほとんど用いられず、専ら人工知能と言われている。人工の素子で脳神経回路網と似たような働きをするハードウェアを作るというよりは、人間の知的活動のモデルをソフトウェア的に実現するというニュアンスが強くなってきたために、用語にも変化が見られるのだとも考えられる。アメリカMITのP.H.Winstonは今日でいうAIの研究の歴史を6つの時期に区分している(P.H.Winston編: The AI Business,MIT Press 1984)。彼によれば1960年頃までは有史以前の時代である。有名なA.M.Turingがチューリング・マシンと呼ばれる仮想的な万能計算機械に関する論文を発表したのは1936年のことであるが、0、1の2進コードの記入された無限長のテープと、その記号を読み書きすることのできる出入力装置を具えた計算機械によって、人間の論理的知識とその操作をすべて代行することができるという趣旨の論文であった。ディジタル・コンピュータの本質について考察した先駆的な論文として、今日でもしばしば引用される論文である。ただし、彼の時代にはそのアイデアをハードウェアとして実現する技術は存在していなかった。

神経生理学の分野で、多数の神経細胞から成る回路網(ニューロネット)の動作を理論的に検討するために、細胞の機能をモデル化して議論した最初の人々はW.S.McCullochとW.Pittsである(Pittsは数学者)。そして彼らは、このモデルによって論理的な知的機能が説明され得ることを主張している。McCullochとPittsによる神経細胞のモデルは、パルス信号を入出力とするまさにディジタル的な論理操作と記憶の素子であった。サイバネティクスの概念を提唱し、フィードバック機構を中心として生物と機械における通信と制御機能の統一的説明を行おうとしたN.Wienerも、神経回路との類似性にヒントを得て、プログラム内蔵式の2進ディジタル電気信号による情報処理機構の基本仕様を作成した。

今日のコンピュータの基本方式にその名をとどめる von Neumannは、このような神経回路網モデルの研究やWienerのサイバネティクスの思想に影響を受けたものと思われる。彼は単に計算機械としてだけでなく、神経回路の情報処理

の仕組みをモデル化するアイデアとして、セル・オートマトン理論の枠組みを考え、そこで自己増殖作用の基礎的な過程を説明しようとした。そして、彼のアイデアはノイマン型といわれる手続き型の計算機方式の提唱につながるわけである。

要するに、この時代は神経回路網のモデル化を強く意識しながら、理論的にはディジタル・コンピュータの幕明けを確かなものにした時代であったということができる。ただ素子技術が未熟であったために、理論や概念上の提言とハードウェアとしての実物との間に大きな距たりがあった。逆に言うと、素子技術がより進み、低廉で信頼性の高い素子が大量に利用できるようになれば、コンピュータはやがて高度の人工頭脳となり得るであろうと考えられていたとも言えよう。

Winstonの言う次の時代は1950年代の後半から60年代の半ばにかけての約10年間であり、人工知能(AI)の夜明けの時代と呼ばれる。AIの歴史における一つの記念すべき出来事は、1956年6月に催されたダートマス大学夏期セミナーの集会である。J.McCarthy、M.L.Minsky、N.Rochester、C.Shannonの4人が呼びかけ、人間の知的活動が正確に記述できるかどうか、またそれをコンピュータでシミュレーションすることができるか、といった問題を議論したもので、参加者の中にはA.Samuel、A.Newell、H.Simonらの名もみえる。この会議が今日のAI研究の嚆矢となったといわれる。

この時代から素子技術もトランジスタからICの時代へと入り始め、ハードウェアの進歩にも支えられて、コンピュータを人工知能ツールへと発展させることにはかなり楽観的な空気が強かった。10年程度の歳月のうちに、ゲームや数学定理の証明、さらにある程度の芸術的価値を持った作曲などにおいても、人間の能力と比肩し得るようなマシンあるいはプログラムが実現するだろう、というような空気であった。研究者の数はあまり多くなかったが、いくつかの研究プロジェクトが進行した。例えば人間を相手にゲーム(チェッカーやチェス)をするプログラムの開発、コンピュータに自動翻訳をやらせようという試み、論理式の証明問題を人間の用いうるようなヒューリスティクス(通常、発見的方法と訳されているが、厳密な論理的推論でなく経験的知識や経験的法則、あるいはそれによって培われた勘のようなもの)を導入して解くプログラムLT(Logic The・

orist)の開発などである。LTはH.SimonとA.Newell によるものであるが、後向き推論(backward reasoning)を用いた点でも画期的なものであるということができるし、ヒューリスティクスの導入がAIの研究にとって本質的な部分を占めることを示した意義も大きいといえるであろう。LTの研究は、その後、GPS(General Problem Solver)へと発展した。

チェッカーやチェスのプログラム開発も多くの研究者が興味をもって進め、ヒューリスティクスの導入、ある程度の学習機能の付与などによって、特に規則の比較的単純なチェッカーについては1960年代の初め頃までにかなり高度なレベルに到達したといわれている。しかしチェスについてはあまり満足すべき結果は得られなかった。そして、自動翻訳の研究は最も期待はずれであった。単なる記号列の置換えだけではとても"意味"のある文章に翻訳することはできなかったのである。この時代の特徴は、人間の知的活動を神経回路モデルに還元して捉えるのではなく、コンピュータを独自の記号・情報処理機械と見なし、そのうえに人間の知的活動をシミュレートし得るようなモデルを乗せようとするものであったと言えよう。この傾向は今日まで引継がれ、ニューロネット・モデルとの訣別は大きなプラスの意義とともに、またマイナスの効果をもたらしているのではなかろうか。別の言葉でいえば、生理学的モデルの立場から、今日で言う認知心理的モデルの立場への転換である。

いずれにしても、予測された10年程の歳月の後、ある程度の進歩に対する満足感は当然伴なったにせよ、総体としては失望感が支配的であった。予測されたより遙かに低い幼稚なレベルのAIしか実現されず、ダートマス会議当時の楽観主義は見事に裏切られたのであった。別の言い方をすれば、これは恰好の反省の機会を与えてくれたわけである。例えば言語に含まれている意味とは何か、人間はどのような仕組みでそれを理解しているのか、あるいはより一般にわれわれの持つ知識とは何か、それをどのように獲得し、蓄積し、利用しているのか、また知識の一般化とはどのような過程を経て行われているのか、そもそも定量化し難い常識とか経験則とかはどんな形でわれわれの認識の中に蓄えられ、また増殖していくのか。言語翻訳のような知的にかなり高度と見なされるような行動のみな

らず,もっとありふれたごく日常的な行動の中でも,人間は表現し難い不思議な 働きを様々な形で営んでいる。こういった不思議に改めて驚き,挫折感の一方で は再びより強い興味を抱くようになる。

夜明けの時代の次に来る数年間,1960年代の後半はいわば暗黒の時代であり,AI研究の一時停滞した時期であった。もちろんこの時代に何も起こらなかったわけではない。例えばリスト処理用のコンピュータ言語として1960年にJ.Mc-Carthyにより開発されたLISPは,従来の言語と異なって,数値だけでなく記号をも容易に取扱えるようにした点で画期的であったが,大型コンピュータの共同利用法としてのTSS($Time\ Sharing\ System$)などに応用され,それにつれてAI研究者の間にも興味を持たれるようになった。

反省と停滞の数年間が過ぎた後、ルネサンスの時代が訪れる。60年代の終り頃というと、素子技術は既にICからLSIの時代を迎えている。超大型機一辺倒の傾向に代って、ミニコンピュータが普及し始めている。ちなみに、DEC社の12ビット機PDP8が1965年に発売され、16ビット機PDP11が登場したのは1970年のことである。ハードウェアの進歩と多様化が一段と激しくなっていった時代である。特に、ミニコンピュータの普及はそれまでの集中処理の思想を変え始め、分散処理の利点が盛んに議論されるようになる。価格の低廉化も一段と進んでいった。

A I の研究家達もあまりに壮大な一般的問題解決の夢を捨てて、各専門領域ごとの問題の特徴とその解決のノウハウを地道に追求していこうという態度に変わってきた。そこから今日言うところのエキスパート・システムの開発への道が開かれる。ゲームや初等的な数学問題を解くプログラムは、知的興味をそそるものではあり得ても、実用技術として見る立場からは多くの非難をこうむることになる。1960年代の終り頃に、スタンフォード大学のHPP(Heuristic Programming Project)においてE.A.Feigenbaum、B.G.Buchanan、J.Lederbergらによって、有機化合物の分子構造を固定するシステムDEN-DRALが開発された。これは今日でも歴史的に有名なエキスパート・システムであるが、彼らの思想は現実に有用性の大きなシステムを開発すること、対象分野

で専門家と同程度の高いレベルの能力を持たせること,システムの動作(推論など)について利用者が十分に理解できるものであること,というような極めて実際的インパクトを重視したものであった。その思想を実現するために,様々の専門的知見を組合わせて利用する発見的ルール(プロダクション・ルール)や候補の順序付けを行うルールを導入して探索(推論)を効率化することに成功したのである。

その後のエキスパート・システムとしては、血液感染症の診断と治療に関する知識利用を支援する目的で開発されたMYCINが有名であるが、その他大型のソフトウェア・システムにおけるトラブル・シューティングを行うためのものや、原子力プラントの故障原因の追求と対処法をプラント・オペレータに助言するためのもの、企業における生産・在庫管理を支援するためのものなど、実用的なシステムがいくつか開発されており、今日最も成功したAIの応用システムと考えられている。エキスパート・システムの構築技術は知識工学(Knowledge Engineering、Feigenbaumの命名による)とも呼ばれるが、これはAIと同義語ではない。むしろAIの応用技術の一分野と考えるべきものである。しかしエキスパート・システムや知識工学の概念と手法がAIの実用的価値を喧伝するのに果たした役割りは、やはり大きかったと言わざるを得ない。

エキスパート・システムの基礎をなす要素技術,あるいはAIの原理を与える 方法というものはいくつか存在するが,それらは次のように大別されよう。

- (1) 知識の表現法
- (2) 探索と推論の方法(ヒューリスティクスを含む)
- (3) 意味の取扱い法
- (4) 学習と自己組織化の方法

このうち,現用のエキスパート・システムに特に関係が深いのは(1)と(2)であるが,

(3)と(4)も将来の重要な課題と考えなければならない。

また、AIの主な対象としては、次のようなものがあげられる。

- 音声認識・理解
- (2) 画像認識•理解

- (3) 自然言語認識・理解
- (4) それらの総合的な認識・理解

さらに、AIの応用分野(課題)としては、エキスパート・システムの他に次のような分野が現在までに試みられ、ある程度の成功を見ている。

- (1) C A I
- (2) 機械翻訳
- (3) 簡単な文章理解
- (4) 自動プログラミング
- (5) 定理証明,問題解決
- (6) マン・マシン対話

先にも述べたように、エキスパート・システムあるいは知識工学はAIの応用の一分野であるが、現在のところ最も盛んに研究され、また実用が普及しつつある分野だといってよい。一方には、コンピュータに人間らしい作業をさせてみたい、それはどこまでできるのかといった知的興味がある。また一方には、複雑な専門的知識を必要とする判断作業に直面した未熟な専門家や素人が、その道のベテラン、大家と言われる人の高度な知識や判断能力を借用し利用したいという、切実な現実的要求がある。この2つの動機があいまって、知識工学技術が最近急速に普及しつつあるのだと思われる。

その要素技術としては、知識表現、知識獲得、知識ベースの構築に関するもの、知識ベースに作用して問題の解答や解決法を探索し、推論する機構(エンジン)に関するものがある。これらは主として、システムを設計し構築する立場からの技術であるということができる。また一方、利用者の立場から言えば、対象分野に関する特殊な知識が無くても使えるものであること(これは利用者のタイプと専門知識に関するレベルによる)、コンピュータのハードウェアとソフトウェアに関してもなるべく特別な予備知識を要しないものであることが望ましい。知識ベースをどのような形式で表現するかということ、どのような機構で推論を行わせるかということ、利用者とのインタフェースをどのような形にするかということは相互に関連した事項であり、一貫した思想と手法を使用することが必要であ

る。

エキスパート・システムにおいて最も多く使われている知識表現の形式は、いわゆるプロダクション・ルールと呼ばれるものである。つまり知識をIF……、THEN……,という形式で表す。IFの部分を条件部、THENの部分を行動部と言い、推論はシステムに組込まれたインターブリタ部によって実行される。知識には確かな事実に関するものと、発見的(経験的)判断に基づくものとがあるが、エキスパート・システムでは後者を積極的に導入するところに意味がある。知識の表現形式を大きく宣言的表現と手続き的表現に分けた場合、プロダクション・ルールによるものは基本的には後者に属するが、モジュール性と柔軟性にも富んでいるという点で、両者の中間的性格を具えたものであると言える。プロダクション・ルールを用いた場合、推論としては条件照合・実行型の推論が行われる。条件に適合したルールを探索して、逐次推論の連鎖を作成していくのであるが、条件に適合したルールが一つでないときには順位付けなど競合を解消するためのルールを設けておく必要がある。

推論の向きにについては、前向き推論(forward tracking、ボトムアップ推論ともいう)と後向き推論(backward tracking、トップダウン推論ともいう)があり、条件(前提)から出発してゴールへ向かう連鎖を作るか、逆にゴールから出発してゴールを成立させる条件を探る連鎖を生成するかの違いがあり、両者が併用される場合もある。初期の有名なシステムのうち、DENDRALでは前者を、MYCINでは後者を用いている。

その他の代表的な知識表現一推論機構の組合わせには,命題論理・述語論理ー 形式推論,意味ネットワークー連想推論などがある。

さて,知識工学はまだ歴史も浅く,それだけにまだまだ問題点も多い。思いつくままにそれらを列挙してみよう。

(1) 知識の整合性と知識の獲得

知識ペースはエキスパート・システムの知識の源泉であり、またその全てである。従って、知識ペースに誤った知識、矛盾したルールが入っていれば、当然のこととして導かれた結論は誤ったもの、あるいは不確定なものとなる。経

験的な知識がやむを得ず持っている潜在的な誤謬の可能性を何らかの形で陽に 表現しておくこと、多数のルールを構造的に整理すること、矛盾を発見し整合 性のあるルール・ベースとすること、などは特に大型のシステムの場合に必 要な技術である。

さらに厄介なのは、新たな知識の獲得の問題である。特定分野の専門家の持つ知識をコンピュータ中のルールに変換する作業は、その分野の専門家とシステム構築者という異分野の人間の共同作業であるが、これには決まりきった手順は無く、手工業的な過程を経なければならない。特定の実世界を抽象的な記号の世界へ変換する過程とも言えるこの作業を支援するツール、手作業を半自動化するツールが今後より積極的に開発されるべきである。一応出来上ったシステムが、ちょうど未完成な知識体系を持った人間が試行錯誤を繰り返したり、より高度の知識体系を持った専門家(先生)に評価を受けたり教えられたりしながら学習を実行して、自らを改良していくような機能(学習機能)を持つことが理想であるが、これは今後の最も重要な研究課題であろう。

(2) ユーザ・インタフェースとユーザ・モデル

エキスパート・システムはそれ自身で完結した働きをするシステムというよりは、人間の判断や決定を支援するためのシステムと考えるべきである。つまり、ユーザに対するコンサルテーションの機能が本質的なものとして重視されなければならない。医用診断、機械の設計、管理の意思決定などにおいてそうであり、特に教育・訓練用のシステムにおいては、マン・マシン協調作業の円滑さが利用効果を支配する。そのためにはユーザに余分の負担をかけずしかも柔軟性のあるユーザ・インタフェースにすること、例えば画像、音声、自然言語(に近い言語)を有機的に組合わせて利用することや、ビットマップ・ディスプレー、マウスのような高解像度画像入出力技術を利用できるようにすることなどが大切である。

それと共に、利用者の現在状態や記憶、推論の特徴などを表すユーザ・モデルが存在することが望ましい。特にCAI(Computer Assisted Instruction)の場合は学生モデル(student model)を構築して、シス

テムが学生の知識の有無、知識の正否、適用法の適否などを知ったうえで、適切な教育戦略を決定することが成功の必須の条件と考えられる。このように程度の高い(sophisticated)教育を行い得るシステムはICAI(Intelligent CAI)システムと呼ばれている。

(3) ツールの整備

コンピュータを応用した各種のシステムの開発及び利用において常にそうであるようにエキスパート・システムがより高度のものとなり、より広い範囲の普及を遂げるためには、ハードウェアとソフトウェアの両面においてコンピュータそのものがより低廉で高速化されること、周辺機器がより安く豊富に整備されること、システム開発環境もより整備されることが必要である。

さて、以上のようなAI、わけても知識工学、知識ベース技術の歴史的展開とコンピュータ技術との関わりを眺めたうえで、われわれが本年度の委員会で調査研究の対象としたのは、近来ますます高機能化しつつあるパーソナルコンピュータとAI、知識工学との結び付きの可能性を見極めることである。冒頭にも述べたようにコンピュータは単なる計算の道具ではなく、より幅広い情報処理のツールであり、それに知的活動の機能を付与する試みがコンピュータ史の大きな部分を占めている。

われわれは既に3年前、昭和56年度の調査研究において「マイクロコンピュータとその利用技術の将来」について考察を加えたことがある。当時は、16ビット・マイクロコンピュータが本格的に市場に参入してきた時期であり、マイクロコンピュータあるいはパーソナルコンピュータが高機能化するにつれて、ユーザはどのような知的活動に個人用コンピュータを利用しているのか、あるいは利用することを希望しているのかについて調査し、そのニーズとまた技術的シーズについて分析と検討を加えることは、たいへん興味ある作業であった。

その時の調査によれば、知的活動において利用したい項目として挙げられたのは

- (1) 個人用データベース
- (2) 文書の作成,文献の整理・検索

- (3) 各種処理(科学・技術研究用のシミュレーション,経済関係のシミュレーション,発想・思考の援助のためのシミュレーション,アイデアの検証,CAD,外国文献の翻訳など)
- (4) 管理(OA, LA的なもの)
- (5) 通信関連(通信用ステーションなど)...

などであった。個人向きの用途とはいえ、その中には業務用と家庭用の双方が含まれている。中でも圧倒的に多かったのはデータベースに用いたいとの回答であった。文書、文献シミュレーション、通信などの項目もなるほどと思わせるものがあった。また、家庭用の中には子供の教育用や趣味のためのCAIが含まれていたのも興味深い。

次に、それらの知的活動を支援するのに必要な、システムの機能として挙げられた上位項目は次のようなものであった。

- (1) 高度な入出力装置
- (2) 大容量メモリ
- ⑶ 個人用データベース及び文書作成用ソフトウェア
- (4) 通信機能
- (5) 自然言語処理能力

中でも(1)の高度な機能を持つ入出力装置に対する要求は極めて強いものがあり、特に入力では日本語(含音声),出力としては高解像度,高品質のCRTディスプレイであった。その他にも、手軽な図形・画像入力装置、手書き文字入力装置、高印字品質のドットプリンタ、VTRやTVとのビデオ入出力インタフェースなども挙げられていた。

以上のような希望や意見を見直してみると、3年前にユーザは既にいろいろと 鋭い指摘を与えていることがわかる。つまり、その後の高機能パーソナルコンピュータの発展の経緯や、それがワークステーションという概念に進んでいったことと照らし合わせると、ユーザの求めるところと技術の進展の方向とが大筋においてよく一致しているといえるのである。

本報告書の第 I 編では、パーソナルコンピュータ及びワークステーションの機

能はどこまで進んでいるかという設問に答える形式でまとめられている。高機能パーソナルコンピュータの誕生の背景は何であったか、現在乃至未来の高機能パーソナルコンピュータの具えるべき機能はどのようなものか、さらにワークステーション(ビジネス用とエンジニアリング用の両者を含む)という概念がどのようにして生まれ、それが持つ機能と果たすべき役割りはどのようなものかについて、第1章で考察と展望を試みた。コンピュータという概念がより多様な知的活動を支援する場としての電子机、すなわちワークステーションの概念へと発展していく筋道が明らかにされている。

第2章では、現在の市場にどのようなワークステーションが登場しているかを 国内外の代表例について具体的に紹介し、特にエキスパート・システム用LISP マシンや IBMウークステーションの動向と次世代戦略に注目した。さらに第3 章では、将来の更なる進化のためにどのような観点から技術的問題に対処すべき かを論じてみた。

第Ⅱ編は、高機能パーソナルコンピュータとワークステーションが現在の局面でどのように利用されているかを述べたものである。代表的な応用分野を3つに分け、第1章では計測・制御という伝統的分野への応用について検討した。第1章では具体的な応用事例を紹介し、CAD/CAM システムについても触れた。複雑な制御システムでは数式モデルによる制御方式の他に、あるいはそれを補完し、あるいはその機能に柔軟性を持たせるために、エキスパートのノウハウを組み込むことが有用な場合が多い。エキスパート・システムやファジィ・システムを応用したコントローラも、ぼつぼつ実用化が始まっている。第3章では、新しい応用分野として最近注目を集めつつある教育システム(CAI)を取上げている。現時点での教育分野別の利用状況をデータに即してまとめた後、将来の本格的な普及をはかるうえでの課題と留意点を挙げている。

第Ⅱ編では、知識ベースとの融合はワークステーションに何をもたらすかについて的を絞り、堀下げて論じてみた。エキスパート・システムやその中核を成す知識ベースについてはこの序論でも述べたわけであるが、第1章ではAIの現状、

知識工学の現状、知識ベース構築技術の現状をより具体的に紹介し、知識とは何かを改めて検討しながら今後の課題にも言及している。第2章は学習・教育、特に社会的にもニーズの高まっている生涯学習の観点から、必要な知識を学習的に検索し提示させるための知識ベースシステムの構築法と利用法について、最近の研究成果に基づいて具体的に述べたものである。さらに第3章では、人間の学習や思考のメカニズムについて人間のサイドから、つまり認知心理学のサイドから考察を加える一つの具体的な試みについて述べている。それは人間がコンピュータと会話しながら学習する形でのシミュレーションを通じて実行されるが、例えば人間が問題を提出し、コンピュータがそれを解くというような独特の過程を通じて、人間の学習モデルに迫ろうとするわけで、将来のAIや知識工学の発展のために基本的に何が必要かについて示唆するところが多いであろう。

第Ⅳ編は本調査研究に関連した最近のいくつかの技術トピックスを紹介したものである。具体的に参考になる点が多いであろう。

以上の各編各章は委員会での討論を踏まえて各委員により分担執筆されたものである。中には委員個人の見方が反映されている部分もあろうかと思われるが、お恕し頂きたい。なお、第IV編については委員外からのご寄稿をも賜った。ご協力に対して深謝の意を表する次第である。



第 I 編 パーソナルコンピュータ, ワークステーション の機能はどこまで進んでいるか

	•

- 第 I 編 パーソナルコンピュータ, ワークステーションの機能は どこまで進んでいるか
- 第1章 現在のパーソナルコンピュータ, ワークステーションの 主な機能は何か
- 1.1 高機能パーソナルコンピュータ、ワークステーション誕生の背景 近年、パーソナルコンピュータ(以下PCと略す)とワークステーション (以下WSと略す)は発展の一途をたどり、上位機種の性能ではこれまでの汎 用中型機並みになろうとしている。ビジネス用、技術用、スタンドアロン型、 ネットワーク型など製品の形態・用途は実に多様である。

今後はさらに素人向け/熟練者向け/専門家向けなどのように、クラス別に 改善されれば除々ではあるが名ばかりだけでなくなり、汎用中型機にも負けず に追い抜いてしまうであろう。

このような躍進を成し遂げられるその大きな原動力は、何よりも図1-1にも 示すように、それは、①技術的立場に起因する基本的な諸変化と、②ユーザの

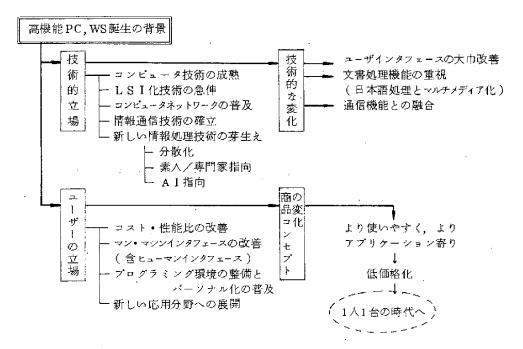


図1-1 高機能パーソナルコンピュータとワークステーション誕生の背景

立場に起因する最終的には"1人1台の時代"を迎えたいという切な念願との 相乗効果で呈じた成果であるといえる。

1.2 多機能パーソナルコンピュータの持つべき機能

PCが高性能化してくると図1-2に示すように、ハードウェアはある水準以上のものが装備されるため、次の諸機能を複合してもつことができるようになる。

- ① コンピュータ機能
- ② 表集計計算機能
- ③ ワープロ機能(当然、日本語ワープロ機能である)
- ④ 通信機能
- ⑤ データベース機能

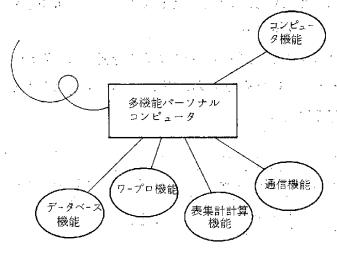


図 1 - 2 多機能パソコン(という巨象のような怪獣!)の 持つべき機能

1.3 高機能パーソナルコンピュータとワークステーションとの関係

高機能PCは、従来のPCに比べると一段と多機能化され、用途に応じて使い分けられる汎用型PCに属するものである。具体的には、次の機能が挙げられる。

- ① コンピュータ機能(データ処理機能)
- ② ワープロ機能(日本語文書処理処機能)
- ③ 通信機能(ローカル・ネットワーク機能)
- ④ 画像入出力機能
- ⑤ 図形,イメージ入出力機能
- ⑥ 高度なマンマシン・インタフェース機能(手による効率よい入力と 視覚に訴える表示・出力が中心だが、場合によって高分解能ビット マップ・ディスプレイ、マルチウィンド、マウスなどのポインティ ング・デバイス、イメージ入出力など)
 - ⑦ 高性能 P C の O S (ハードウェア機能と同様に、ソフトウェアの中核の O S として次の機能が要求されている。すなわち、
 - マルチプロセス処理
 - 入出力の仮想化(統一されたファイル概念)
 - 『・スァイル管理機能
 - ローカル・ネットワーク機能
 - 日本語入出力機能
 - ・・マルチウィンド機能
- 画像入出力機能……"
- ・図形・イメージ入出力機能
 - コマンド・インタブリタをおきなると、 1992/2007
 - ・言語プロセッサ(C, Pascal ; BASIC, FORTRAN ,

- COBOL, Ada, Lisp, PROLOG)

 - (水漬書プログラム) in the comparing an income
 - 機密保護するからいます。
 - 、・その他ユーティリティ等がある。)....

こうなると、図1-3に示すように高機能PCとWSとは非常に似かよった ものになってしまう傾向を示す。それもそのはずで、図示のように高機能PC はもともとWSの核となすべきものであるからだ。

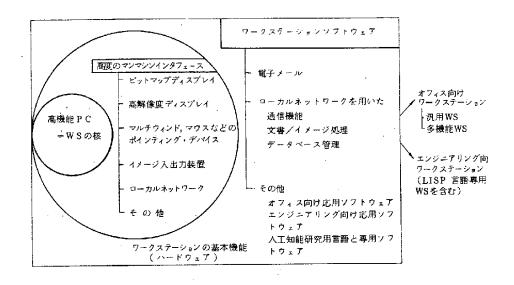


図1-3 高機能パーソナルコンピュータと.

ワークステーションとの関係

現在のWSの原型は、Xerox 社のPARCで1973年に開発されたALTOであるといわれている。元々、"Dyna-Book"と称する大型ノート大の専用コンピュータ開発の発想がその前身である。現在では、さらにAI(人工知能)化思想をも取り入れるに至っている。

しかし、WSは実は古くて新しいことばである。WSの研究は、もともと"人間の知性"についての研究との密な関係があって、1962年10月に発表されたSRI(Stanford Research InstituteのDouglas C. Engelbartの論文"Argmenting Human Intellect: A Conceptual Framework"にそのことが述べられている。

それによると、WSも道具の一つとして人間の知性の向上のためにあるものであると記述されている。ここでいう知性とは、複雑な問題に対処する力、その問題を理解する力、問題解決の方法を見い出す力などであるとしている。

だが、Engelbart の指摘によると、この考え方は、そもそも科学者である Vannevar Bush(1945)の提起が最初であるという。Bushのアイデアによれば、WSは"机"であり、memex という名前の装置であると述べている。また、Bush は今後の頭脳労働者の仕事量に十分対処できる道具としてこれを提案し、これは機械化された個人ファイルとライブラリのようなものでなければならないと強調している。

このように、WSのイメージは既に早くから定義されている。ただ、必要な 機能としてこれまでの技術では実現できなく、普及できなかっただけのことで あった。

従って、WSの機能も今後の技術の成長と共に完成されていくものであるが、 その原型はすでに存在しているといえる。

表1-1は、WSに必要とされる機能および重要度について調査した最近の一例である。この表では、確かに米国の一側面をのぞいたものであるが、おそらく日本の場合もこのほかにさらに日本語文書処理機能を追加すれば、とりあえず納得できるとなろう。

表1-1 ワークステーションに必要とされる機能の重要度

	1		
,) } 1 1 1 1 1 1 1 1 1	小	• 中	大
1. ドキュメント処理と配布		- 19 A A	
キーボード/ワードプロセッシング		,	0
ドキュメント・フォーマットの格納 ドキュメントのローカルで格納			
			. 0
(オフィス内での他の部門からのアクセス) ; ;			
語句のスペル用辞書		0	
効果的ルーチン		0	· ·
多量ウインドウ		<u> </u>	L
2. オフィス・アプリケーション	V 5 2 G		· ' .
オフィス・データの処理		0	1.5 pt 157
対務表の分別	Y	<u> </u>	
データ/金融分析		<u> </u>	ļ
電子メール		0	
音声メール・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・	· O.		1
ワード処理			
3. ソフトウェア			<u> </u>
デバッガー	0		
トール・トランスレーター(大学)・大学・デートである。	0 /		ř
C.P/M	0_		
MP/M	0	4	1 17
Unix very service of the service of			
4. グラフィック			
カラー		0 .	
イメージ処理			
CAD/CAM	0	,	
ビジネス・グラフィックス			0
5. データベース	L		1
インハウス・データベース・アクセス			
外部データベース・アクセス	0		
6. コミュニケーション	·		
X 2 5	. 0		
3270との互換性	 -		10
プロードバンドLANとのインタフェース	İ		<u> </u>
ベースパンドLANとのインタフェース		1	Ŏ
PABX E O 1 × 2 × 2 × 2 × 2 × 2 × 2 × 2 × 2 × 2 ×		0	
SNAとの互換性	L	1	0
7. インプット/アウトプット	J	4	
インフット/ リファンター インテリジェント・コピア/プリンター	T	Ιο	
カラー・ブリンター		+	
	0	+	
プロッター			0
文書作成プリンター	1		
音声入出力	0		
タッチ・スクリーン		 	+
カーソルの移動(マウス)		1	9
高速プリンター	<u> </u>		0

(出典:コンピュートピア1985年1月号 クリエイティブ・ストラテジィ社調べ)

第2章 現在どのようなワークステーションが開発されているか

2.1 パーソナルコンピュータとワークステーションの市場動向

最近、にわかに「高機能パーソナルコンピュータ」や「ワークステーション」 という言葉が使われている。このブームはここ当分続きそうである。

パーソナルコンピュータ(PC)についての市場動向や生産台数の推移についてはよく知られているが、ワークステーション(WS)になると意外とそのデータが乏しく、今一つはっきりしないところが現状のようである。

そこで、比較的に先駆者である米国を例に、その市場動向を眺めてみることにする。表 2-1に示すように、WSは1983年には60億ドルであったものが、1985年に2倍、1986年には3倍と急速にその市場を拡大しつつある。図2-1に示すように、米国の特殊性から特に金融および保険、続いてソフトウェア開発などの専門サービス関係が圧倒的に多く利用していることがよくわかる。

表2-1 アメリカにおけるワークステーション市場の動向(単位:10億ドル)

		, -		
	1983	1985	1987	1989
汎用ワークステーション	3.948	8.4 7 9	1 5.3 0 0	2 5.1 4 3
多機能ドキュメント処理	2.257	3.623	5:0 1 7	6.210
エンジニアリング・ワークステーション	0.07:0	0.259	. 0.435	0.514
計 :	6.1 7.5	1 2 3 6 1	2 0:7 5 2:	3 1.8 6 7
· · · · · · · · · · · · · · · · · · ·				

(出典: コンピュートピア 1985年1月号 クリエイティブ・ストラテジィ社調べ)

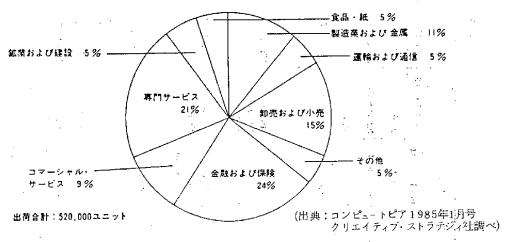


図2-1 アメリカにおける産業別の汎用ワークステーションの出荷状況(1982)

表2-2 主な国内オフィスワークステーション一覧

メーカ名	沖電気	E	Ą	高 🗄	E 通	日. 文	東芝	9 77 -	削工者	松下電器
機種名	オフィスターミナル OKITAC-200	N5200-05	N6300-55	FACOM9450-II	FACOM2750	H1TAC T-560/20	TOSBAC DP/2	GW200	SEIKO3300 "アンサー"	Operate 7000
CPU	8086 - 2	8086	8086	MN 1613	オリジナル	8086	オリジナル	80286 (8MHz)	8088 - 2 + 8085A - 2	MN 1613
生記憶容量	512 ~ 768 KB	MAX640KB	MAX768KB	384 ~ 512 KB	192KB	256 K B	128 ~ 512KB	1.5~2.5MB	256 ~ 512KB	512 KB
表示能力(ディスプレイ)	14 インチ・カラー 720 × 480 1,120 × 70	12インナ 640 × 495	1,120 × 720	12 インチ・カラー /グリーン 640 × 480	モノクロ 960×675 カラー960×700	15インチ・モノクロ 1,120 × 720	14 インチ・モノタロ 640 × 500 ANK 2,000 文字 漢字 1,000 文字	2.048 × 1,536	12 インチ・オラー /クリーン 640 × 500	
o s	MS-DOS 専用OS	MS-DOS CP/M86 PTOS	NTOS	CP/M86 APCS-III	CP/M-86	MS-DOS ホストサポート VOS3, VOS1/SE	専用OS	MP/M-86 CP/M-86	CP/M CP/M86 専用OS	CASII
音額	GW-BASIC LEVEL-II COBOL 他	BASIC COBOL	FORTRAN	BAS1C COBOL	COBOL FTL	拡張 BASIC OF IS∕POL	COBOL FRIEND DATA CUP	C BASIC COBOL	BASIC COBOL FORTRAN SBOL	BASIC COBOL 簡易言語 アセンブラ
通信 インタフェース	RS232C		シリアル インタフェース	RS232C		SYN同期 HDLC	JIS~->>> BSC SDLC HDLC ACOS-6	イーサネット	RS232C 通信回線 IF	HDLC BSC
	105~	69. 8	91 ~	98~	111 ~		258 ~	710 ~	82.5~	120 ~
主な用途	・ O A システム 用オフィス統 合システム ・オンライン ターミナル	・OAシステム	用分散処理ターミ	・ネットワーク! データ処理 パーソナル処!		・マルチWS・ビデオWS・	・日本語データ 処理が可能な O A システム 用トータル W S	Gシリーズ用 WS ・汎用WS	事務処理経営管理資料作成ワープロターミナル	• OA ,LAN 指 向事務処理

確かに日本の市場になると、現在はまだ比較的高価であるXerox 社のStar、Apple 社のLisa などを連想し、その後も後継機が続々登場しているが、どちらかというと高嶺の花であって、いま一つ普及していないのが現況である。しかし、国内市場についてもWSの開発状況を把握し、生産台数の推移をつかめるようにする必要がある時期がきていると思われる。おそらく、ごく近い将来においてWSと高機能PCはその適用分野および技術的な面において除々に接近しつつあるため、いずれは一つに集約されると思われるが、後になっても正確なデータが追跡できるように、いまから正しい推移を残しておきたいものである。

2.2 オフィス向けワークステーションの動向

手近なデータ処理はパソコン、文書作成もパソコンまたはワープロ、データベース照会はホスト・オンラインのターミナルなどといったような専用機器が続々と登場し、OA活用に実践的に展開されているが、オフィス向けWSも着実にビジネスWSとして成長し、OAにおける一つのブームとして注目されつつある。

表2-2には主な国内で商品化されたオフィスWSを順不同に一覧表にまとめてみた。表中には、主にビジネス指向のWSをメインに収録したが、どれも基本的には高機能PCの登場によって形成される新たな潮流と融合しながらも汎用かつ多機能WSとしての道をたどっているようである。

しかし、原型であるSTAR(ゼロックス社)をはじめ、マルチステーション5550(日本IBM)、Lisa(アップル)などについては意識的に表中から省略したが、まさに無視できない存在となる数々の巨大な群像であるといえる。

2.3 エンジニアリング・ワークステーションの動向(国内商品)

次に、国内商品を中心にエンジニアリングWSについて調べてみることにする。表 2-3が、主な国内エンジニアリングWSを一覧表にまとめたものである。これからも続々と開発されると予想されるが、現在のところは主にCAD/CAM用や図形処理用のものがファーストステップとして開発されている。

表2-3 主な国内エンジニアリング・ワークステーション一覧表。

				· · · · · · · · · · · · · · · · · · ·
メーカ名	東 芝	ソード	精工舍	ソニーテクトロニクス
機種名	UX-300F	M685	SE1KO9100	4115J/M4115J
CPU	ナリジナル	68000×2 (12.5/6.25MH ₂)	8086 + 8087 8087 - 2	8086 + 8087
主記憶容量	512 ~ 1 MB	4 KB キャッシュ 512 K~3MB	384∼896 KB	
	15インチ縦型 漢字ディスプレイ	14インチ・カラー 16 色同時表示	12インチ・カラーン グリーン	
	800×640 ANK 80字×40行	$1,024 \times 780$	640×480	1,280×1,240
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1	漢字 40字×40行			,
os taka	∵OS∕UX ⊶	UNOS (UNXV7 とコン パチブル)	i RM X 86	CP/M 86
言語	FORTRAN : COBOL C BS Tセンプラ	FORTRAN COBOL C PASCAL	BASIC	DTI(ダイレクト ターミナル・イン
通信インタフェース	RS-232C GP-IB	RS-232C	RS-232C 高速 RS232C LAN	RS-232C DMA
価格(万円) :	430 ~	475 ~	118~	495 ~
主な用途・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・	・OA,FA,CA 分野に対応	・CAD/CAM ・電子メール ・リレーショナル データ・ベース	・ CAD ・ 科学技術用WS	・グラフィックWS ・高速図形処理

今後は、LSI設計指向のものが次のステップとして開発されていくであろう。年々急激に増加している技術者たちの計算作業、設計作業負荷の軽減をはかるため、パーソナル指向の設計支援ツールとなる新しい時代のエンジニアリングWSが望まれている。

次期エンジェアリングWSの開発は、いろんな意味においても、少なくとも 先端技術のレベルを維持するためにも重要な道具の一つとして、そろそろ注目 を浴びてもよいのではないかと考える。

2.4 エンジニアリング・ワークステーションの動向(国内で入手可能な輸入商品)

エンジニアリングWSは,

高度なマンマシンインタフェースをサポートする高 性能ディスプレイによる対話処理



人間の思考に頼るルーチン化しにくい仕事を助ける



エンジニアリングの生産性向上

といった図式の目的を達成することが可能であるということで、海外では早 くも商品化されている。

だが、技術的には直ちにこの製品の低価格化、高速化、高解像度化、インタフェースの向上、日本語処理の強化、分散データベースの実現などを一挙に実現することが困難なため、国内メーカの開発状況はどちらかといえば多少遅れぎみである。特に、ソフトウェア工学分野の技術適用は、かなり遅れているようである。WS関係アプリケーションソフトウェアについても大いに見習う内容がある。

現在,国内で入手可能な輸入商品であるエンジニアリングWSにかぎっても,

その製品は多様である。従来のCAD/CAM分野の製品に加えて、LSI設計用WSが脚光を浴びている。次に、簡単であるがその主な例を挙げておくことにする。

- CAE 2 0 0 0 (CAE Systems / 日本システムテクノロジーズ)
- LOGICIAN (Daisy Systems/セイコー電子)
- Tegastation (GE Calma /伊藤忠データシステム)
- λ755/λ745(Metheus /大日電子)
- IDEA1000 (Mentor Graphics / メンターグラフィックス・ジャパン)
- I C A P (Phoenix Data Systems / 技術情報研究所)
- SL-2000 (Silvar-Lisco / 伊藤忠データシステム)
- SCAL Dstar (Valid Logic Systems/テクノダイヤ)

2.5 エンジニアリング・ワークステーションの動向

(エキスパートシステム用LISPマシン)

エンジニアリングWSのもう一つの開発の動向として、いわゆる専門家向けのエキスパートシステムの商品化が進んでいる。

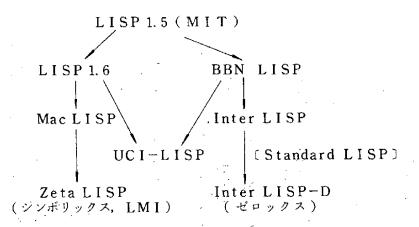
この種のシステムの特徴は、表 2 - 4 に示すように、LISPマシンが主流である。これらのシステムは、人工知能の研究開発の中で今最も脚光を浴びているのがこのようなエキスパートシステムである。

そのほかに、国内では日本DEC社がSYSTEM-20をLISP用ミニョンとして、沖電気がPROLOG用コンピュータORBITを、日立がUTILISPによる操作ガイダンスをもったWSを、それぞれ市販しようとしている。

実際現在のエキスパートシステムは研究所をやっと抜け出しかけている段階にあり、解決を急がねばならない問題が散在している。ほとんどのエキスパートシステムが利用しているプログラム言語はLISPであるが、このLISPには数多くの方言があり、導入してもなかなか使えない。大雑把にみても、次のように分岐している。

表 2 - 4 エンジニアリングワークステーション(エキスパートシステム・LISPマシン)一覧

メーカ名	メーカ名 富士通		富士ゼロックス	LMI(伯東)		シンボリックス(ニチメン 東洋情報システム)
機 種 名	FACOM 2	S S - 3 E	1 1 0 0	Sevies II	Lambda	LM-2	3600
CPU		68000 (12.5MH ₂)		32ビットマシン	UNIXプロセッサ	32 ピットマシン	カスタマイズ
主記憶容量 仮想空間	8 M バイト 16 M バイト	1 ~ 8MB 8KB	1.15 ~ 1.5MB 8MB	256 K B 16 M B	(6800,10MHz) 1 ~ 20MB 16M/4GB	1 ~ 4MB 16MB	1.125~36MB 1 ギガバイト
表示能力 (ディスプレイ)		14 インチ・カラー 1,024 × 768	17 インチ 1,024 × 808		モノクロ・カラー 1,024 × 800	800 × 900	カラー(オプション) 1,100 × 900
ディスク(フォーマット容量)			23 H B	20 ~ 600 MB		80MB	67 MB
D S		UNIX V		Zeta Lisp	Zeta Lisp-PL US UNIX	Zeta Lisp	Zeta Lisp
言語	UTILISP	BASIC FORTRAN C LISP PROLOG	Interlisp-D Smalltalk-80	Zeta Lisp	FORTRAN C PASCAL LISP PROLOG	Zeta Lisp	FORTRAN PASCAL Common Lisp PROLOG
通信インタフェース		RS 232 C LAN	イーサネット LAN .		RS 232 C バラレル(8/16 ビット) イーサネット		RS 232 C イーサネット LAN
価格(万円)		790 ~			1,800~6,500		2,400~3,500
主な用途など	ホストコンピュー タFACOM-Mシ リーズ、S-3000 シリーズとセット にして使用する。 86.3月発売予定	業務,CAD/CAM	よるポインティン		人工知能 自然言語 故障診断 CAD/CAM	仮想記憶 1ページ=1KB	人工知能 記号処理技術 ソフトウェア工学



利用者にとって、使用をさらに難しくしているのは方言だけでなく、LISP を本格的に使用できるコンピュータが少ないこともある。

もし、LISPの標準化も進んでいれば、多くのエキスパートや設計技術者にとっての作業環境が大きく改善されるだけでなく、LSI化によるLISPマシンの実現にも一層の拍車がかけられるにちがいない。

2.6 | BMワークステーションの動向

2.6.1 現在の | ВМの Р С ファミリ

まず、現在のIBMのPCファクシミリを図2-2に示す。IBMの動向を見ずにしては、大きな戦略は立てられない可能性があるからである。

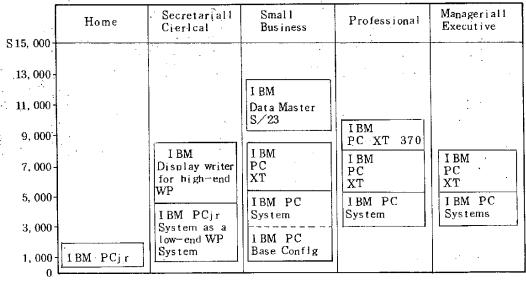


図2-2 IBM PCファミリ(1983年頃からの)

2.6.2 PC XT/370 概略

PC XT/370は、IBMのWSの考え方を推察する上で、またこれからのWSの方向を考える上で、まずは取り上げなければならない製品である。

PC XT/370は、PC XTに3枚の専用ボードを加えたもので、370命令セットのほぼ全命令が実行でき、メインフレームOS VM/SP(Virtual Machine/System Product)が使用できる。XT/370 用のVMは、VM/PC(VM/Personal Computer)と呼ばれ、これはXT/370が単一ユーザ4300と等価であることを示している。

3枚の専用ボードは、それぞれ 1 個の I BM特別仕様のマイクロプロセッサを持ち、2 個は 6 8 0 0 0 0 I BM用修正版、他は 8 0 8 7 の修正版である。 I BMのみの専用のものであるといわれている。

1個の68000は、150個の370命令セットのうちの固定小数点命令の72命令を実行し、他の68000は浮動小数点を除く60個の命令をエミュレートする。8087は、370の浮動小数点命令を実行する。

専用ボード内には、370マクロ用メインメモリとして512K RAM が用意されている。370チップは、XT8088ベースのマザーボードの RAM 256 Kはアクセスできない。また、ボード内には3277エミュレーションが含まれており、XT/370は3274クラスタコントローラに直接結ぶことができ、ホスト計算機と通信し、ファイル転送ができる。

図2-3には、 PC XT/370ソフトの概念図を示す。 X T/370は I BMホストと結ばれているとき、3個のウィンドで同時に V M アプリケーションできる専用のマルチウィンドインタフェースを持っている。この時、 MS-DOSは4番目のウィンドで動作する。

しかし、PC XT/370は次に示す制限があるため、4300および 308XのすべてのVM/SPアプリケーションを実行することができない。

(1) VM/PCは、単一仮想計算機

VM/SPは、マルチ仮想計算機能を持っており、VM/SP下でMVS, DOSを運用することができる。マルチ仮想計算機機能を必要とする

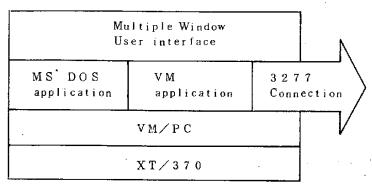


図2-3 PC XT/370概念図

PROFS (Professional Office System), データベース用SQ L (Structured Query Language)などのアプリケーションプログ ラムは単一仮想計算機XT/370上では動かない。

(2) 実メモリ、仮想記憶の制限

VM/PCの仮想記憶は、4MBで実メモリは512KBである。512KBの内100KBはOSに使われ、ユーザアプリケーション用は400KBである。従って、これらの制限を越えるアプリケーションは動かない。

(3) パフォーマンス・・・

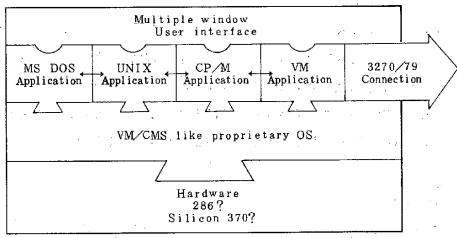
実メモリとページング数の制限から、パフォーマンスが非常に悪い(レスポンスタイムなど)場合がある。

2.6.3 次世代のワークステーション

PC ATの出現によって、次世代のWSの輪郭をのぞけるようになり、 図 2 - 4 に示す PC 3 の概念図が明確となった。 プロセッサは 8 0 2 8 6 が採用されていることもわかった。

専用OSとして、XT/370同様VMが採用され、そのサブシステムとしてMSDOS、UNIX、CCP/Mが同時に複数のウィンドで動作し、これらのアプリケーション間でのデータアクセスが可能である。また、PCXT/370の制限もかなり除かれている。

その結果として、図2-5に示すように、現在のIBMのPCファミリへ



DESIGN PRINCIPLE OF PROPOSED NEXT GENERATION IBM PC OPERATING SYSTEM

図2-4 PC3 概念図

S 15,000	Home	Secretarial l Cierical	Small Business	Professional	Managericall Executive	Scientific Engineering
13,000			,			Low-end 4300 workstation
11,000						
9,000 -			IBM PC XT multiuser version (S/23 replacement	IBM PC	"intelligent" IBM PC	"Modular" IBM PC XT11 capable of accepting
7,000 -			with intel286) IBM PC XT··	XT 11 (Silicon 370)	XT 11 (Al software)	MC 68020 NS 16032
5,000 -		IBM PC in a Displaywrite configuration	single user version IBM PC fully config- ured System	<u> </u>		intel 432 intel 386
3,000 -		IBM PC jr Sys- tem as a low-end WP SYSTEM	ured System	Transportab- le IBM PC	"Smart" Ter-	٠.
1,000 - 0	IBM PC jr	MASISIEM	Base System	Note Booksi- zed IBM PC	minal Phone	

図 2-5 1985 IBM PCファミリ

第3章 更なる進化のために、今何が必要か

3.1 先ず、歴史の脈流を静観せよ

高機能PCもWSも決して一日にして突如に降って湧いてきたものではないことは、既述のBush の"机"であるmemex の懐胎に始まり、ALTO試作機 (1973)を原型に誕生したことは周知のとおりである。

現在、はやりのマウスでさえもすでに $1962\sim69$ 年頃に、Engelbartによって試作されていたことも事実である。(図3-1参照)

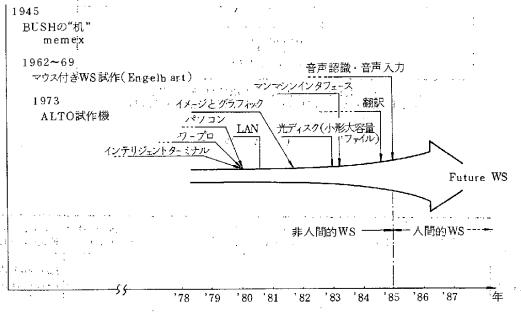


図3-1 ワークステーションの歴史

WSは、その後の数々の新技術の導入によって、特に高機能PCの出現、LAN技術、ディスク技術、高解像度ディスプレイ技術などによって、多種のWS商品化が可能に至っている。しかし、どの製品もどちらかというとマシン偏重型であって、いわゆる非人間的WS指向のものが主流となっている。誰にでも簡単に使えるようには洗練されていない。

そういう意味で、Future WSとなる今後のものは、もっとヒューマンインタフェース指向を重んじたマンマシンインタフェースに気を配り、いわゆる人間的WSにめざしてもらいたいものである。

3.2 低価格・高機能化できる技術による対処

高機能にして低価格化されうる商品は、ユーザとしての大きな魅力がある。 高機能PCやWSに対しても例外にはならない。

図3-2には、米国の場合のデータによるものであるが、WSの価格動向を

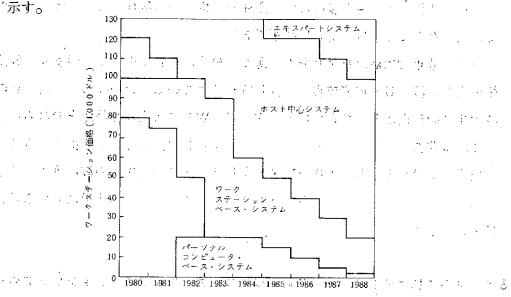


図3-2 米国におけるワークステーションなどの価格動向

(出典:日経エレクトロニクス '84 7/30号; Daratech Inc 調べ)

図示のように、パーソナルコンピュータ・ベースシステムも、ホスト中心システムも、ワークステーション・ベースシステムやエキスパード・システムも、いずれも確かに含まれるアプリケーションによって変わるが、ハードウェアの価格は低下し続けている。結果的には、いつかはユーザはいろいろな能力を持つコンピューティング・リソースを混在して利用することができるようになると考えられる。

従って、ハードウェア技術としては、たとえばゲートアレイやPLA素子などによるVLSI技術を積極的に導入しても、ファームウェア化を駆使しても挑戦しなければならないことがたくさんある。

3.3 "机"では持ち歩けない

WSは"電子の机"として大いに成長している。書類を電子化し、オフィスの道具としてアイコン(絵文字)化し、画面上でディスクワークが行えるようになりつつある。

だが、"机"では持ち歩けない。会社の"机"から離れてしまえば、仕事ができなくなるようでは、それは問題である。

一方、近年では商品をより軽く、薄く、短く、そして小さく、つまり"軽薄短小化"の波が日本の産業界にも押し寄せてきている。いわゆる"一人一台の時代化"の傾向がやってきている。従って、WSにも次期商品開発のねらいとして、是非あって欲しいものは、ポータブルWSの出現であると推察できる。

出来るだけ小型化した多機能WSの早期商品化こそ、次代の波に乗るべく姿であると予測できる。それだけに、日本のエレクトロニクス産業に対する期待は大きいと言える。

さらに拡大の動きをみせているのが通信機能の増強である。とくに、異なる レベルのコンピュータとの協調が進むにつれて、プロトコル(通信手順)の標 、準化の積極導入が予期されている。是非手遅れにならないうちに、いまからは すでに着手しなければならない時期がきていると痛感している。

WSを端末として稼動させる際に使われる通信媒体・通信速度などのフィジカル・インタフェースとホストコンピュータのネットワークアーキテクチャや伝送制御手順などの論理的インタフェース、さらにその上で行われるオンライン機能にできるだけ制約がないことが望ましく、いかなるオンライン処理にも耐えられる柔軟性をもったオンライン機能が必要である。

通信媒体として、公衆回線・特定回線・DDX網での接続、さらにLANへの乗り入れにも配慮が必要である。論理的インタフェースとしてIBM社のSNAを始め、ホストコンピュータの持っているネットワークアーキテクチャの

もとに標準端末のエミュレーションやトランザクション処理のできるプログラムインタフェースを用意して、あらゆるオンライン形態に応じられるソフトウェアおよびハードウェア構造が必要である。

. 次に、アプリケーションに関するもので、国内では特に目立って立ち遅れているので、急ピッチな開発計画が必要である。

独自のソフトウェア開発も必要だが、OAにかぎってもその応用分野が急速に広がりつつある。従って、効率のよい開発を行うためにも標準化手法の導入が必要である。官民一体による開発体制も、産学共同開発の組織化もときには 偉力を発揮するものがある。そして、一日も早く生産性の高い開発環境が準備されることが先決である。

3.5 UNIXを超えるOSが必要

高機能PCやWSのOSとしてUNIXが世の中の標準となりつつあるが、ネットワークへの接続およびより複合化したアプリケーションの構築のためには、少なくともUNIXにおける①リアルタイム性および②プロセス間通信の強化などが必要で、その点で弱味をみせている。

そういう意味で、UNIXを超えるOSが必要である。

事実、IBMのUNIXの導入は、新規ユーザの確保およびUNIX PC /WSに対する攻撃が目的であると考えられるが、IBMの主眼は、UNIX でも、MS-DOSでもなくIBM専用OS(VM)である。IBMは、VM 下でサブンステムとして各種OSを動かすことが本来の目的であると容易に推 察できなければならない。

3.6 第5世代へ向けての集約化

今後、WSに期待される機能は増々ふえる一方であろうが、何よりもマンマンン・インタフェースの改善を最重点化すべきである。

エキスパートシステムもやがて成長し、知識ベースをもった思考支援システムとして成長していくだろうが、アーキテクチャ的には第5世代指向のものに

集約して欲しいものがある。そして、人間のように知識をもち、推論し、判断するような夢のコンピュータの利用方法があるといわれる人工知能の分野の高まるニーズにこたえて欲しい。文字どおりある分野のエキスパート(専門家)と同じように考えられるシステムに一歩でも近づける努力を払ってもらいたい。最近、巷間でソフトウェアの危機が話題となっている。それは、マイクロプロセッサの急成長、パソコンの出現、データ通信の量の激増などにより、開発しなければならないソフトウェアの量は10年で5倍というハイペースで伸びているのに対して、プログラマは年率3~5%の増加、ソフトウェアの生産性向上は2~3%であるといわれている。これではソフトウェアの作成は、まもなく現在のような人海戦術ではとても対処できなくなることが予想される。

そこで、このソフトウェアの危機を救うのがこのような人工知能を利用した 自動プログラミングが使用できるのではないかといわれている。この自動プログラミングの原型を、現在のコンピュータを使って行っている意志決定支援システムの中にみることができるが、同様の機能を次期に出現するポータブルタイプのエキスパートシステム上には、除々でも実現できることを切に期待したいものである。

参考文献

- (i) 「"ワークステーション"新時代」コンピュートピア誌 '85年1月号 P 138~
- (2) 「特集・拡がるワークステーションシステム」データ通信誌 '84年3月号 Vol.16, M 3
- (3) 「実用段階を迎えたワークステーション」日経エレクトロニクス誌 '83年5.23号 P 147~172
- (4) 水田浩「エキスパート・システム」ビジネスコミュニケーション誌*84年12月 P 64~71
- (5) 桑原啓治「商品化が進み、研究も活発なLISPマシン」日経エレクトロニ クス・ブックス「人工知能」 '83年4月 P175~187

第Ⅱ編 パーソナルコンピュータ・ワークステーション はどう利用されているか

		3

第Ⅱ編 パーソナルコンピュータ・ワークステーション はどう利用されているか

第1章 マイクロプロセッサの応用とワークステーションへの発展

1.1 計測・制御へのマイクロ・プロセッサ応用の発展経緯

1970年代に始まったマイクロ・プロセッサ技術は、現在までに大きく成長し、既に3世代になると考えられる。

実用的なマイクロ・プロセッサの第1世代は8ビットで始まった。この時は、制御への応用では、マイクロ・プロセッサはプログラマブルな制御素子として扱われ、ハードウェア・ロジックの置き換えが主であった。ハードウェアとソフトウェアの分業は行われず、設計・製作者は、マイクロ・プロセッサとその周辺機器のハードウェアに精通した上で、制御用ソフトウェアを作った。こうして作られたハードウェア、ソフトウェア共にそれぞれの専門システムであり、汎用性はない。マイクロ・プロセッサのこのような利用形態は、制御用プロセッサの大きな分野として現在に至っている。

第2世代は、汎用OSの出現に始まった。この世代の発展に大きく寄与したのは、外部記憶としてのフロッピィディスクの発達である。汎用OSとファイルの利用は、第1世代の組込形の応用以外に、より高度の計測、制御システムへ、マイクロ・プロセッサを利用できるようになった。

OSの出現により、ソフトウェアの作成はハードウェアから独立し始めた。 ハードウェアとソフトウェアの両者に精通した人間は、システムの作製者には 求められても、大部分の応用プログラムは、ソフトウェアの専門家によって作 られるようになった。

ソフトウェア・ツールとしてのプログラム言語も、アセンブラからFORT RAN等の高級言語へと進歩し、データ処理能力も高まった。

こうしてOSのサポートを受けたマイコン・システムは、各装置のメイン・ プロセッサとして、全体の管理、監視、制御等に利用されるようになった。

16ビット・システムで代表される現在の第3世代において、こうしたマイ

コン・システムの高性能化はますます進んだ。16ビット・システムでは、マイクロ・プロセッサの処理能力の向上と同時に、汎用OSの種類も増え、システムのメイン・プロセッサとして使用される度合がより多くなった。

・制御用システムにおいても、各種のリアルタイムOSが現われ、リアルタイム・マルチタスクによるプロセス制御への応用が始まった。

16ビット・ンステムでは、プロセッサのメモリ空間が1MBと大きく、かつウインチェスタ・ディスクのような、高速、大容量の外部記憶の利用が可能になったことにより、画像処理、データ・ペースのような、従来ミニコン以上のシステムで行われていた処理が可能となった。

この世代の特徴として、更にマイクロ・プロセッサの多様化があげられる。 マイクロ・プロセッサの応用は、高速ロジックの置き換えから、大規模制御 ンステムのメイン・プロセッサとしての利用まで広範にわたり、それぞれの規 模に応じて各種の機能を持ったものが現われた。

(1) 周辺回路の取込み

高速ロジックの置き換えのような用途を目的とし、主に組込形で用いられ プロセッサに周辺回路を取込んだものが多く現われた。主な周辺機能として は、ROM、TIMER、DMA、ADC、DAC等がある。

(2) 低消費電力化

CMOSを中心とした技術により、組込みシステム用として用いられる、 プロセッサとして8/16ビットの汎用プロゼッサの低消費電力化が進んだ。 これによって装置の発熱を押え、設置環境の悪い所でも完全密閉システムで 24時間運転を長期間続けることが容易になった。

(3) 高機能化

プロセッサのクロック周波数を上げ、処理能力を向上させるのみでなく、 命令セット等の高機能化によって処理能力の向上が行われている。主な機能 としては、算術演算機能、ストリング処理機能、タスク・スイッチング機能、 メモリ管理機能等がある。これらのものは、プロセッサ内部に組込まれたり、 コ・プロセッサとして提供されたりする。 (4) 特殊プロセッサ

通信プロセッサ、画像処理プロセッサ等に代表されるような、特殊目的の 高速、高機能プロセッサである。

· 使一生 化化二十分 医纤维 一条 生活

各次公司的 医复杂病 医牙囊 医复数形

計測,制御システムとしては、上記のような多様なマイクロ・プロセッサを それぞれの目的に応じて各機器に使用し、それらを通信手段で統合して全体を 構成するような、分散処理システムが今後の主流になるであろう。

宝ごのような計測に制御システムの発達から、『次世代のマイグロ・プロセッサ として、『次のような方向が求められる。『『言語』 「「「「「「」」 「「」」

- 1. 1. 1 、高性能汎用プロセッサの おおい カラン もっと シャップ
- ・ 1.16/32ビットの高性能汎用プロセッサが、大規模ジステムのメイン・ プロセッサとして求められる。このようなシステムで実行されるプログラム は次のようなものが考えられる。
 - (1) 図形処理

- (2) データ.・バモス機能 ペンコー きーコーケーション 2g から2m3人
- (3) 通信機能 システムの分散化はより強まり、それらの間の通信機能は一層重要となる。通信の手段も接続されるステーションの性質によって各種のものが要求される。通信は、システム内で直接に機器制御を行っているフロント・エンドのプロセッサ・システムから、システムのメイン・プロセッサへ、

更に、メイン・プロセッサから事業所全体の大型システムへという縦の通信と、複数の制御システム間での横の通信との両者に広がる。

1.1.2 高速制御用プロセッサ

汎用プロセッサにメイン・プロセッサとしての高機能化が進められると、 それらを機器の直接制御に使用することは、コスト、実装サイズ、消費電力、 プログラム作製の容易さ等の点で不都合が生じてくる。このために、小規模 システムのための高速の制御用プロセッサが求められる。これは現在の8/ 16ビット・プロセッサの高速化、高信頼化、低消費電力化の方向である。

以上,主にマイクロ・プロセッサ自身を中心として,計測,制御システムへの応用の発達をみてきたが、次に,これを利用したソフトウェアをみてみよう。

1.1.3 オペレーティング・システム

8 ビットのフロッピィディスク・ベースのOS, CP/Mが, マイクロコンピュータ・システムの汎用OSの流通的な始まりであった。

8 ビット・システムにおいては、メモリ容量や外部記憶容量、処理速度等の制限のため、シングル・ユーザ、シングル・タスクに限られた。

このシステムは汎用システムとして、計算処理などのプログラムの実行を 行うもので、特に制御用のリアルタイム機能をそなえたものではない。

8ビット用のリアルタイム・システムとしては、iRMX、MTOS等いくつかのシステムが発表された。

これらはいずれも組込みシステム向きであり、それぞれの専用のアブリケーションと組合わせてシステムを構成する。

16ビット・システムになって、メモリ容量の増加(IMB)、ウィンチェスタ・ディスクの利用により、外部記憶の大容量化、高速化、プロセッサの処理速度の向上等により、マルチタスク・マルチユーザのOSが実用化さ

れた。

更にOSは、それぞれの目的によって特長付けられた。即ち、一般の事務計算処理等を目的するもの、プログラムの効率的な開発を目的とするもの、リアルタイム・マルチタスクによる計測・制御を目的とするもの、シリコン化により組込みシステムへの利用を目的とするもの等がある。

この傾向は、マイクロプロセッサのそれぞれの応用分野での利用効率を高めることになった反面、相互のシステム間でのデータやプログラムの互換性をなくし、プログラム開発やデータ処理の能率化の防げともなってきた。

このような反省から、最近では各目的向きのOSのアーキテクチァを統一し、そのようなシステムをマイクロプロセッサの種類についても統一的に用意しようとする動きや、ファイルの互換性を特長とするシステム等が表われた。¹⁾²⁾

現在マイクロ・コンピュータ用OSとして販売されているOSの代表的な ものを表1-1に示す。

表 1-1 マイクロコンピュータ用 0 S一覧(1)

OS名	メーカー名	特	徵
CP/M-86	Digital Research	汎用OSの一つである。	来,最も広く使用されている分を分離しているために,移
CCP/M-86	Digital Research	マルチタスクで動作できる.	たマルチウィンドによる仮想
MS-DOS	Microsoft		- DOS)として広まつた。 が,パイプ,リダイレクショ ド処理を取入れている。

¹⁾ 坂村 健;マイクロブロセッサ用標準リアルタイムオペレーティングシステム 原案:TRON,情報処理学会第28回全国大会編文集(1984)

²⁾ Little, J; "A Module Approach to Microcomputer Operating Systems", COMPUTER DESION, July 1984

表 1 - 1 マイクロコンピュータ用OS一覧(2)

- 0, S名	メーカ名	特徵
S 1	Multi Solutions	マルチユーザ,マルチタスクの汎用OSである。 OSはモジュール化されており,再構成が容易である。 CP/M,MS-DOS,UNIX等と互換ファイルが 使用できる。
	Softech Microsystems	Pascal 教育用システムとしてカリフォルニア大学で作られたものから発展した。 中間言語を使用しているので、異なるCPUへの移植性が良い。
	Microware Systems	UNIXに似たコマンド体系を持った68000用のマルチタスクOSである。 実行プログラムを全てメモリ・モンュールという形で扱い。モジュール単一でROM化も可能である。
UNIX APPENDANCE	AT&T	Bell 研究所でPDPおよびVAX用に開発されたTSS型のOSである。プログラム開発等のソフトウェア開発システムとしてのツールを多く持つている。'16/32ビットのCPUに対するOSとして注目されている。
Pick	Pick Systems	データベース管理を中心とする事務用OSである。ミニコン用システムから出発したが、現在では各種のマイコンシステムに移植されている。
ai+CROS86		8086システム用のリアルタイム・マルチタスクOS である。CP/M互換のファイルを使用し、CP/Mの BDOSコールもエミュレートできる。
MTOS	Industrial Programming	組込用のリアルタイム・システムとして, 8086, 68000用がある。 ROM化して使用できる。
i RMX	Intel 社	8086/80286用リアルタイム・システムである
ELX-86M	エミックシステム	8086用のリアルタイム・マルチタスクOSである。 CP/M: 86上で開発し、ターゲットシステムに移植 する。
-V R T X	Hunter and Ready	リアルタイム・シリコンOS 2KB EPROM×2で提供される。 8086, 68000, Z8000用がある。

1. 1. 4 プログラミング言語

アセンブラ、FORTRAN、COBOL、BASIC等のなじみ深い言語に加えて、最近は各種の問題向き言語が使用され、その主なものを表1-2に示す。

表1-2 プログラミング言語一覧

: 言語名	特	274	
		敬	マイコン用製品
	科学計算用言語として最も古 る。 最新の規格はANSIのFO も採用している。	,	ai-FORTRAN86 MS-FORTRAN FORTRAN-86 SSS-FORTRAN
	初心者用の言語として, マイ いる。インタープリア言語で, 持っている。	コマンドに編集機能等を	CBASIC-86 BASIC Q9 BASIC 86
	事務用言語として大型機から る。 A N S I ニ 7 4 で規格され,		LEVELI-COBOL CIS COBOL RM/COBOL COBOL 86
С	AT&TのUNIXと共に普 述も可能である。 構造型言語で制御用プログラ		ai-C86 OPTIMIZING C86 LATTICE C C86
Pascal	構造型言語として広く使用さ UCSD p-Systemの主		PASCAL/MT+86
Ada	アメリカ国防省の標準言語と ミニコンからマイコンまでの している。マイコン用では,	制御システム作成を目的と	JANUS/ADA
PL/M PL/I	構造型言語として8086用 効率が良い。 PL/M, PL/I共に同様	•	PL/M-86 PLM86 PL/I-86
LISP	記号処理言語で、今後、人工 の一つとなるであろう。	知能等の分野で主要な言語	TLC-LISP
PROLOG	人工知能研究などで注目され マイコン用はインタープリタ		PROLOG micro-PROLOG PROLOG-J

制御、計測用言語として専用に設計されたのはAdaであり、その大きな特徴の一つにタスク分割と、その並行処理サポートがある。このような考えは、C、Pascal等にも追加機能として加えられたセットも発表されている。この機能は、OSによるサポートと共にそれを容易に実現するための言語上での記述法が、制御、計測プログラムの重要な問題の一つとなる。

一方、システムの知能化のための言語として、**LISP、PROLOG**等の発達が著しい。これらは今後、システムの処理能力が向上し、知能化が進むと共にますます発展するであろう。

1.1.5 データベース

データベースは、従来、大型機における機能であったが、最近リレーショナル・データベースを中心としてマイコン用データベースが各種発表されている。

データベースの利用は、メイン・システムにおける生産、在庫管理のための記録、エキスパート・システムの知識ベース、計測・制御システムにおけるデータ管理等に今後一層多用されるであろう。

表1-3は、現在使用できるデータベースの主なものを示す。

表1-3 データベース一覧 (1)

データベース名	メー ヵ 名	システム名	言語インタフェース	備考
dBASEII	Ashton-Tate	CP/M-86 MS-DOS	なし	リレーショナル 専用言語によるプログ ラミングができる。
CONDOR20	Condor Computer	CP/M-86 MS-DOS	なし	リレーショナル
DATA ACE	Computer Software Design Inc.	CP/M-86 MS-DOS UNIX	なし	リレーショナル
INGRES	Relational Technology Inc.	UNIX	Cobol Fortran Pascal.C.Basic	リレーショナル

表 1-3 データベース一覧(2)

データベース名	メーカ名	システム名	言語インタフェース	備考
DATAFLEX	Data Access Corp.	CP/M-86 MS-DOS	PASCAL/MT+	リレーショナル
PERSONAN PEARL	PEARLSOFT	CP/M-86 MS-DOS		リレーショナル 事務用統合ソフトウェア • パッケージ
RAPPORT	Logica		COBOL FORTRAN	汎用データベース FORTRAN で記述され 各種システムに移植可
PDBS PDBSII	日本ビジネス オートメーション	UCSD p = System	UCSD pascal	個人用汎用データベース
Oracle	Oracle Corp.	UNIX	COBOL FORTRAN C. Pascal	リレーショナル
Micro RIM	Micro RIM Inc.	MS-DOS	COBOL FORTRAN BASIC Pascal	リレーショナル
UNIFY	Unify Corp.	UNIX	C. RMCOBOL	リレーショナル

(出典:一部日経エレトロニクス '83, 8-1より記載)

1.1.6 図形処理/画像処理

16ビット・システムのメモリ容量の大きさ、処理速度の向上は、大量の データ処理を必要とする図形/画像処理への応用を可能にした。

この分野は、ビットマップ・ディスプレイの発達と共に、CAD/CAM 等の専用ワークステーションを生み出した。

このような応用システムについては、次節で改めて述べる。

1.1.7 通信システム

通信システムの分散処理形式のシステムを発達させ、それぞれの目的向き に専門家されたシステムの組合わせにより、並行処理能力、即時処理能力、 コスト・パフォーマンス等の向上を計る。 通信システムは、大きく分けて公衆/専用通信回線を用いる広域通信システムと、構内のような局所化された地域内でのローカルエリア・ネットワーク(LAN)に分けられる。

計測・制御への応用の面から見ると, LANの応用は, 今後のシステム構成の最重要な要素の一つとなる。

現在発表されているLANシステムは、表1-4に示す。

計測・制御用としては、今後、画像/図形データ、音声データ、数値データ等を同時に送るため、一層高速のシステムが要求される。

現在発表されているLANシステムは、いずれも各社の独自プロトコルによるもので、異なるシステム間の接続はできない。

ISOによるOSIモデルの規格化や、国内で提案されているパソコン通信プロトコルなど、統一化の動きと、それに対応するシステムの開発が期待される。

1.1.8 マイコン応用事例 1

計測・制御システムにマイクロプロセッサを利用する目的の一つには、システムの知識化がある。計測・制御する対象の複雑化、精密化に伴い、その運転者にも高度な知識が要求される。しかし、これを運転者の熟練のみに頼ることは、人件費の高騰をまねくのみでなく、不注意による誤り、運転者の違いによる品質の差を生じることにもなる。

このようなシステムを、一般の未熟練労働者にも容易に扱え、かつ一定の 品質を保障するためには、コンピュータ・システムによる助けが必要となる。 ここで示す例は、自動車の車両検査システムである。検査対象となる自動 車は、近年各種の規制やユーザの要求により、複雑な機器を多く搭載し、そ の検査、調査作業は複雑化した。更に、最近の技術の進歩の早さは、検査、 調整作業手順の改訂が多く、作業者の訓練に多くの費用と時間を必要とする。

³⁾ バーソナル・コンピュータ通信に関する調査研究会『パソコン・ネットワーク規格書♪

表1-4 ローカルエリアネットワーク

			I.			
モデル名	メーカ名	アクセス方式	伝 送 路	伝送速度	形態	備考
ETHERNET	Хегох	C S M A / C D	同軸ケーブル	10 Mbps	バス	Intel
OKINET - 2000	沖電気工業	CSMA/CD ・ トークン/	同軸ケーブル .:光ファイバー	10 Mbps 32 Mbps	バ ス リング.	
C - Ne t	コアデジタル	R S . 232 C	ツイスト・ペア, 光ケーブル	300~9600	スター	
サンヨー・インフォメーション ネットワーク・システム	三洋電気	CSMA/CD	 同軸ケーブル 光ファイバー	10 Mbps	パス	
S-NET	ソード	C S M A / C D	ツイスト・ベア	1 Mbps	バス	
C&C NET	日本電気	CSMA, CSMA/CD トークン タイム・ストロット	ツイスト・ベア 同軸ケーブル 光ファイバー	1~32 Mbps	バス, リング	
BIPLAN	National Semiconductor	ポーリング	「同軸ケーブル 光ファイバー	3.5 Mbps	2 9 —	
OMNINET	Corvus System Corp.	トークン	ツイスト・ベア 光ファイバー	1 Mbps	, , , , , , , , , , , , , , , , , , ,	
ai - NET	アイ電子測器	CSMA/CD トータン	同軸ケーブル ツイスト・ベア, 光ファイバー		バ ス リング	:.
Net/One	Ungermann Bass	C S M A / C D, トークン	同軸ケーブル	10 Mbps	パス	
Cy - NET	サイメックス	CSMA/CD/ACK	同軸ケーブル	1 Mbps	バス	
ARCNET	Datapoint	トークン	同軸ケーブル	2.5 Mbps	バ ス 	
Appolo Domain	Appolo Computer	トークン	同軸ケーブル	12 Mbps	リング	
RINNET	リコー	CSMA/CD	同軸ケーブル	10 Mbps	バス	ETHERNET

この車両検査システムは、このような問題を解決する一つの方法として、マイクロコンピュータ・システムに必要な検査、調整手順を、データベースとして搭載し、作業者との会話形式で、順次作業を進めていくものである。検査は約70メートルのコース上で、最大5台の車両が同時に扱われる。検査項目は次のようなものである。

- ① サイド・スリップ・テスタ
- ② ブレーキ・テスタ
- ③ スピード・メータ・テスタ
- ④ ホーン・テスタ
- ⑤ ハンドリング・テスタ
- ⑥ 排出ガス・アナライザ
- ⑦ ヘッド・ライト・テスタ
- ⑧ 車台装置テスタ

検査システムとしては、これらのテスタの外に、車両の移動と、その位置 検出システム、作業者への表示と、データ入力装置、結果を印刷するプリン タ等で構成されている。

このシステムでは、マイクロコンピュータ・システムは原則として、検査 とその結果による作業者への指示を行うのみで、調整作業自体は人間によっ て行われる。

マイクロコンピュータ・システムは,直接,計測・制御を行っている8ビットのサブ・システムと,そこから通信回線で接続されている16ビットのホスト・システムで構成されている。作業現場は気候の変化による温湿度の差が大きく,排気ガスや外部から浸入する塵埃も非常に多い。このため,サブ・システムは完全密閉形で,外部と絶縁した熱交換システムを持っている。サブ・システムは,検査データの収集と作業指示の表示を,ホスト・システムとの通信による指示を受けて行う。

ホスト・システムは、作業現場から離れた好環境の事務所内に設置されているので、特別の設置対策は取られていない。

ホスト・システムは、サブ・システムから送られたデータに基づき、データの解析、判断を行う。そのために、ホスト・システムには20MBのウィンチェスタ・ディスクを用意し、各メーカの車種に応じた検査、調整手順をデータベースとして記録している。

検査手順は次のように行われる。

(1) 車両の導入と条件設定

車両は外部より運転者によってコースの所定位置まで移動される。車両 位置は光電管および磁気センサを利用した位置検出システムで測られ、必 要な指示をパネル・ディスプレイで運転者に与える。

検査の開始時に与えるべき項目は、次のものがある。

. . .

- ・ ・ ・ ・ ・ ・ ① メーカ名 ・ ・

 - ③ 車両番号
 - (4) シャーシ番号
- リー・コート ⑤ エンジン番号
 - ⑥ 検査手順
 - ⑦ 両検査項目

これらの入力は、専用キーボードにより直接ホスト・システムに送られる。これにより、ホスト・システムは必要な検査手順をデータベースより 選択し、サブ・システムに指示を与える。

(2) 検査データ入力と制御出力

検査データ入力は、全体で約200点になる。アナログ・データは、サブ・システム内のADCインタフェースでディジタル化し、接点等のデジタル入力は、各検査項目毎にまとめてデジタル入力インタフェースで受け取る。分析装置との間はシリアルの通信インタフェースで接続されている。制御出力は、作業指示ディスプレイ、油圧システム等で、約40点のデジタル出力である。

(3) 検査手順

検査は、車両が所定位置に設置された時から始まる。一般に検査には、 運転者による何らかの車両の操作が必要であるので、各検査項目毎に車両 の前上方に、大型の表示装置が設けてある。運転者は、この表示に従って 車両の操作を行う。

大型の表示装置は、操作指示を示すのみで、検査の結果調整が必要となった場合の調整手順は、別のCRTディスプレイに表示される。このCRTディスプレイは、ホスト・システムに接続されていて、検査結果から判断した調整手順をデータベースから求め、調整マニュアルとして作業者に示す。

全項目の検査、調整が終了したら、その結果が所定の書式に従って印字され、コースの出口で車両の運転者に渡される。

1.1.9 マイコン応用事例 2

前例は、マイクロコンピュータ・システムの利用によって、高度な作業に 未熟練労働者を使用する道を開くものである。一方では、作業者の熟練度に は関係なく、大型荷物の扱いのように、数人の補助作業者を必要とするよう な場合や、作業者の相互連結や調整のために人手を要する場合がある。

このような一例として、ストック・ヤードにおける入出庫管理がある。

このシステムは、鋼材のストック・ヤードにおいて、入出庫を行うクレーン群を制御するものである。

ストック・ヤード内には18台のクレーンがあり、5レーンに分割されている。各レーンには、それぞれ3台のクレーンが配置され、残りの3台は入出庫路に配置されている。

今までの入出庫作業は、クレーンの運行管理、作業指示、運転等の全てが 人手によって行われていた。

本システムは、これらの作業の省力化を図り、かつ効率を上げるために、 マイクロコンピュータ・システムにより、管理・制御を行うものである。 システムは、クレーン群の作業管理とストック・ヤードの在庫管理を行うメイン・ステーションと、各クレーン毎の制御を行うローカル・ステーションで構成されている。

メイン・ステーションは、ストック・ヤード内の一室に設置された固定機で、ローカル・ステーションは、各クレーンに一台づつそれぞれの運転室に 設置された移動機である。これらの間は無線モデルを用いた通信によって結 ばれている。更にメイン・ステーションは、専用線によって事業所全体の計 算機システムとも結ばれ、出荷指示と在庫情報を授受する。

メイン・ステーションの主な構成と機能は次のとおりである。

- ① 16ビット・マイクロプロセッサ・システム トーニュー・
- - ③ 外部記憶 ウィンチェスタ・ディスク 51 0 MB は 5 点
 - ④ 通信インタフェースデータ・いイウェイ・インタフェース ** A**無線式通信システム ***

メイン・ステーションは、常に現在の在庫状況をデータベースとして保管 し、更にクレーン群の作業内容と現在位置を常時モニタしている。

ホスト・システム(事業所全体システム)より、出荷指示を受けると、データベースを検索し必要な製品の在庫情報を求める。。

この情報と、その時のクレーン群の運行状態を合わせて、メイン・ステーションは最も適当と思われるクレーンに対して運行指示を発行する。

出荷の情報は、作業完了時にローカル・ステーションより送られ、それを ホスト・システムに通知する。

以上の作業をメイン・プロセッサはオペレータなしの無人システムで行う。 ローカル・ステーションの主な構成と機能は次のとおりである。

- 16ビット・マイクロプロセッサ・システム
- ② メモリ容量 ROM 64KB

RAM 48KB

- ③ プラズマ・ディスプレイ
- ④ 無線式通信システム
- ⑤ クレーン位置検出システム
- ⑥ クレーン荷重計量システム
- ⑦ クレーン制御インタフェース
- ⑧ データ入力用専用キーボード

ローカル・ステーションは、メイン・ステーションからの運転指示を受けると、クレーンの運転者に対して現在位置と目的地までの地図をディスプレイ上に表示し、運転開始を通知する。この表示は、クレーンの現在位置と目的地までの距離に応じて順次縮尺を変えて、運転者にとって自分の位置の認識とそれに基づく運転制御が効率良く行われる様にする。

クレーンの運転は、全てこのディスプレイと、吊上げフックに設けられた センサを利用し、従来のような信号手をなくし、運転者の単独運転とする。

現在は、上記のように各クレーンには運転者がおり、ローカル・ステーションは、位置の検出とクレーン相互の衝突防止などは行うが、緊急停止等の 。安全管理上の最少限の制御以外は、直接クレーンを制御しない。

これは、本システムが全て人手による管理・制御システムから、始めて無人化への移行システムであり、重量物の運搬に対する運転・制御のノウハウのシステム化等の課程上にあるためで、長終的には、完全無人化を目ざしている。

以上のように、本システムは、専用・組込みシステムといえるローカル・ステーションとシステムのメイン・プロセッサとして高度の判断を行うメイン・ステーションを通信手段によって結んだ分散型の制御システムの一つのモデルといえるものである。

1.2 ワークステーションへの応用

前節では、工業用の計側・制御を中心として、マイクロプロセッサの発達と その応用について記した。

本節は、マイクロプロセッサを用いた製品として最近著しい発展をしている ワークステーションについて述べる。

一口にワークステーションといっても、それぞれの用途によって性能、価格に大きな差がある。単なる文章作成、簡易計算等を行うことを目的とするもので、いわゆる一般のマイコン、又はパソコンといわれるものに、それぞれのソフトウェア・パッケージを使用しているものから、一方では、専用のカスタム・プロセッサを使用し、高速のグラフィック・プロセッサ、通信インタフェース、専用マンマシン・インタフェースをそなえたCADシステムのようなものまである。

ワークステーションを、その中心のCPUの形式で分類すると次の3種類となる。

- (1) 専用のカスタムのCPUを使用したもの。主として、マイクロプログラミングによるプロセッサを使用している。XeroxのAltoやStarに代表される。
- (2) 汎用のマイクロ・プロセッサを使用するもの。最近の16/32ビット・マイクロプロセッサを使用したもので、SUNやDOMAINに代表される。
- (3) カスタムVLSIのCPUを使用したもの。YP9000にみられるように、自社製のLSIによるCPUを用いたもので、今後人工知能の発展に従い、LISPやPROLOG専用マシンのような形でなくなると思われる。本節では、この内第2の形式によるものに限って、そこにおけるマイクロプロセッサの応用を考える。

ワークステーションに要求される機能は,

- ① 高性能プロセッサ
- ② 大容量メモリ,大容量外部記憶

- ③ 高速, 高分解能グラフィック・ディスプレイ
- ④ ネットワーク・インタフェース(LAN)
- ⑤ マンマシン・インタフェース

等である。

表1-5に見られるように、現在のワークステーション専用システムとして発表されているものには、68000系のプロセッサを使用したものが多い。これは、68000のアーキテクチャがVAX等の32ビット・ミニコンと類似し、プログラムの移植が容意であることや、アドレス空間が線形で16MBと広く、大量のデータ処理が可能であること等によると思われる。

今後のCPUとしては、68020. NS16032, 80286がある。いずれも高性能プロセッサとして、高速処理、高度な命令セット、大容量のメモリ空間を持ち、各種の周辺LSIを用意している。

ワークステーションに不可欠の機能の一つが、高速、高分解能のグラフィック・ディスプレイである。CAD/CAMのような図形処理を中心とするもののみでなく、OAシステムにおいても自由な文字やビジネス・グラフの表現を求めて、ビットマップ・ディスプレイが中心となっている。マンマシン・インタフェースとして、マウスを使用し、アイコンによる図形化された手続きや、マルチウインドウによる表示等、ディスプレイ機能の占める比重が大きくなるにつれ、グラフィック・プロセッサを独立させるシステムが多くなる。

グラフィック・プロセッサとしては、ビットスライス・プロセッサ、汎用マイクロプロセッサ、専用グラフィック・プロセッサの3種類の構成があるが、 今後は高性能化をめざし、専用プロセッサの開発が多くなると思われる。

ローカルエリア・ネットワーク(LAN)機能は、前節でも述べたように、 現在ではEthernet と Token - passing の2方式が中心である。いずれも 専用のLSIを使用することができる。LANをシステムの重要な構成要素と して取入れ、特長のあるシステムを作っているものにアポロDOMAINがある。

次にOAシステムとCAD/CAMシステムについて、いくつかの例を見てみよう。

表1-5 ワークステーション一覧

								•
機種名	メーカ名	CPU	記憶容	量(MB) ディスク	ンステム	ディスプレイ	通信機能	備考
Lisa	Apple Computer	68000	1	10	Lisa OS	12"ピット・マップ	RS-232C .	事務用
NCR9005	NCR	8086	0. 128 ~ 0. 64	10 ~ 80	MS-DOS CP/M-86	12"ビット・マップ	OMNINET	事 務 用
P-RODUCER-100	Prime Computer	8086	0. 256 ~ 0. 512	5.6~16	CTOS	キャラクタ	ETHERNET	事務用
OKITAC-200	冲弧 员工業	8086	0.64 ~ 0.768	10 ~ 21	専用OS	14"ピット・マップ	OKINET-2000	事 務 用
O A - 9 0 D X	シャープ	68000	0.77 ~ 1.3	10 ~ 40	UNIX	12"ビット・マップ	ÉTHERNET	事務用,
SE1K03000	相工合	8088	0. 256 ~ 0. 512	10 ~ 40	SBOL CP/M-86	12"ビット・マップ	ST-NET	事務用
N 5 2 0 0	日本電気	8086	0. 256 ~ 0. 64	40	PTOS, CP/M-86 MS-DOS	12"ピット・マップ		事 格 用
GW-200	1) =	80186	2 ~ 6	20 ~ 40	MP/M-86	17"ピット・マップ	RINNET	事 務 用
DOMAIN	Appollo Computer	68000	0.5 ~ 3.5	34 ~ 158	AEGIS	17"ビット・マップ	.DOMAIN	况 用
SUN-1	Sun Microsystems	68000	1 ~ 4	84	UNIX	17"ビット・マップ	ETHERNET	况 用
Gateway	Forward Technology	68000	0.5~2	80	UNIX	15"ピット・マップ	ÉTHERNET	汎 用
CONCEPT	Corwus Systems	68000	0. 25 ~ 0:5	5. 7 ~ 19. 7	ccos	15"ピット・マップ	OMNINET	ж н
λ750	Metheus	68000	1 ~ 4	30 ~ 60	UNIX	19"ピット・マップ	ETHERNET	エンジニアリング用
LOGICIAN	Daisy Systems	8086	0.75 ~ 1.5	40	UNIX LIKE	17"ビット・マップ	ETHERNET	エンジニアリング用
SCALD	Valid Logic - Systems	68000	1 ~ 6	33 ~ 280	UNIX	20*ビット・マップ	ETHERNET	エンジニアリング用
				70 m			-	

1. 2.1 **OA**システム

ここでOAシステムとは、次に述べるCAD/CAMシステムに対して、 文章処理を中心として考えられたシステムのことをいう。

一般のマイコン、パソコンといわれるものに、ワードプロセッサ・ソフトウェアを付加したものや専用のワードプロセッサ・ンステムも、OA用ワークステーションの一つといえるが、ここではワークステーション専用システムとして開発された例である、GM-200とDOMANについてみてみよう。

これらのシステムに共通の特長として、1にビットマップ・ディスプレイがある。

これは、OAシステムといえども、文章の自由なレイアウトや、文章中でのグラフや図の利用が重要で、図形処理機能が重要とされることを示している。図形処理は、一般に大容量のメモリ空間を必要とし、プロセッサも大きな空間を扱う必要がある。ここに示した例では、80286も68000も16MBまでのメモリ空間を扱うことができる。

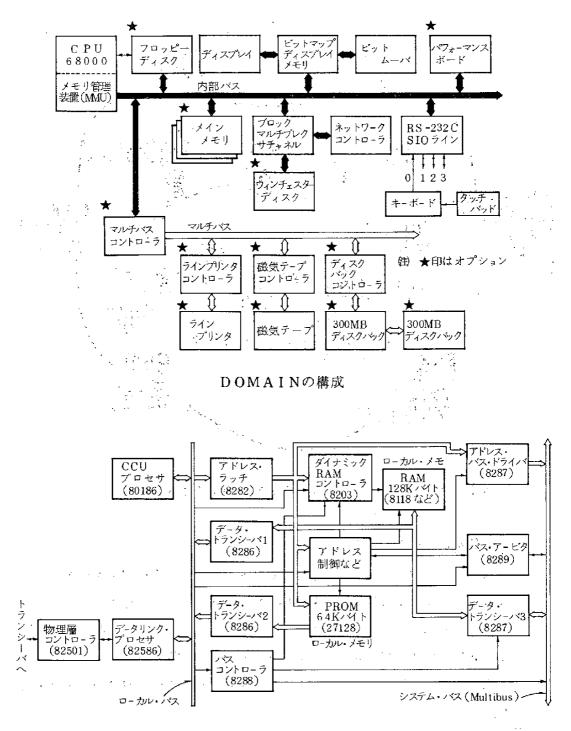
システムに実装されているメモリは、それぞれの構成によっても異なるが、 最小 0.5 M B から、最大 5 M B 程度である。

GW-200は、80186の2プロセッサ構成となっており、主プロセッサとディスプレイ・プロセッサを分離し、ディスプレイ機能の高度化が、応用プログラムの処理能力を防げないようにしている。

DN300は、アポロDOMAINを構成するワークステーションの一つである。DOMAINはシステムがLANを中心とした分散処理システムであり、DN300の他に、いくつかの表示ノード、サーバ・ノードで構成される。 *)

このためにDN300は、単一ステーションとしては比較的小型にまとめられている。

^{*)} ノードとは単一のステーションのことを云う。



GW-200の構成

図1-1 OA用ワーク・ステーションの構成

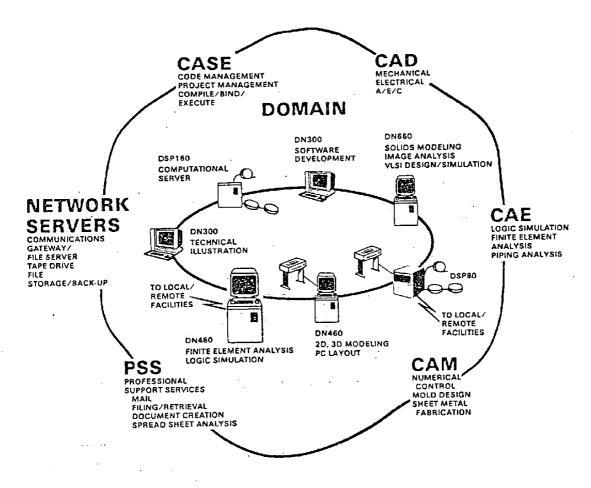


図1-2 ネットワークを中心としたDOMAINの構成

1.2.2 CAD/CAMシステム

前出のDOMAINは、CAD/CAMシステムへの応用も多いが、ここ *) では λ 7 5 0 と I BM-PCを使用したシステムについて述べる。

主プロセッサは、UNIXシステムで実行され、 $1\sim 4MB$ の主記憶を仮想記憶方式で使用している。

^{*)} ブルース・グラッドストーン「IBMのパーソナル・コンピュータを使った簡易CAD用 ワークステーション」 日経エレクトロニクス '83 5-23

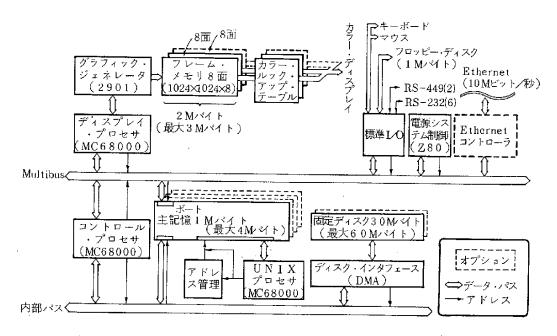


図1-3 3つの68000で構成される 1750 システム (日経エレクトロニクス '847-30より転載)

エンジニアリング・ワークステーションの特長の一つに、高分解能カラー・ディスプレイがあるが、 λ 750も1024×1024×8 のカラー・ディスプレイを持ち、専用プロセッサとして68000を使用している。

一方、IBM-PCは、世界的に広く使用されている汎用パソコンである。 DASHIは、これにグラフィック・ボードとメモリ 及び通信インタフェースの専用ボードを付加している。

DASHIは、専用のエンジニアリング・ワークステーションに比べて、 ディスプレの分解能、処理速度、記憶容量などでは小規模のものであるが、 約\$13000.-という低価格で、個人向けのシステムを実現している。

今後、汎用マイコンの高性能化は、このように、専用オブションを追加するのみで、専用ワークステーションを作り上げる傾向を増すと思われる。

1.2.3 グラフィック・ディスプレ装置

最後に、ワークステーションとは異なるが、専用システムへのマイクロ・ プロセッサの応用の一つとして、CR3000グラフィック装置を示す。 これは、CADシステムの端末として、図形イメージ処理機能を重視した 装置である。

高速図形処理は、パイプライン・バスで接続された専用プロセッサで行い、 それの制御用として68000を用いている。

システム全体としては、主プロセッサとチュネル・プロセッサ、グラフィック・プロセッサの3つが68000で構成され、それぞれの周辺機器には280を使用したペリフェラル・プロセッサが用意されている。

このシステムのように、各レベルの汎用プロセッサと専用プロセッサの組 合わせによって、システムを作りあげる方向が多くなるであろう。

_ 18 ---

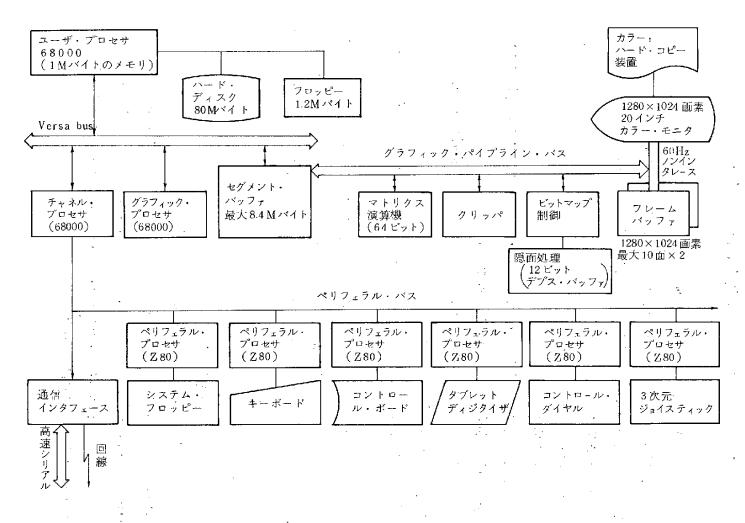


図1-4 CAD用グラフィック・ターミナルCR3000の構成

(日経エレクトロニクス'84 6-18より転載)

第2章 オフィスシステムでどう利用されているか

2.1 生きたOA化の役割とニーズの把握

PCおよびWSによって処理可能なオフィス情報を大別すれば、一般に①計数情報と②非計数情報に分けられる。非計数情報は、さらに

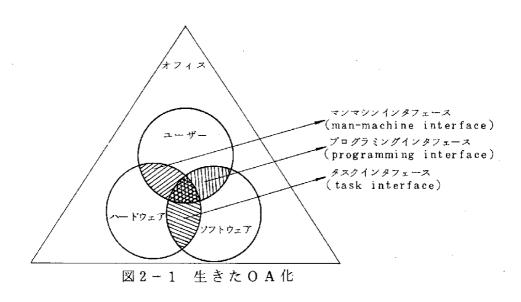
- ① 文書情報
- ② 図形,イメージ情報
- ③ 音声情報

に分類される。計数情報も非計数情報も、 OAの実現によってそれぞれの情報の伝達・処理・制御の技術が確立して可能となった。

やがて、PCもWSも人間とオフィスシステムの接点に位置づけられ、オフィスに働く人達を直接支援するオフィスマシンとなるので、何よりも主役である人間との親和性を基本に、これからのOAシステムを構築しなければならない。

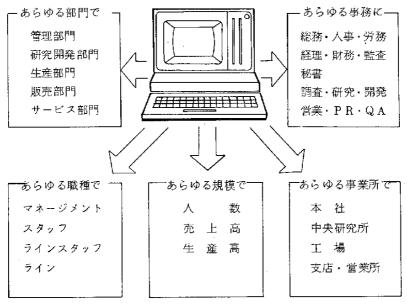
オフィスオートメーションは、単に事務の機械化やその延長ではない。 OA 化によって、オフィスでの機械的ルーチンワークを機械にまかせ、単純作業から働く人を解放し、生産性を向上させながら、経営の各管理レベルにも役立つ価値ある情報を提供する、という一連の融合されたシステム構築を目指したものでなければならない。

つまり、図2-1に示すように、ハードウェアとソフトウェア間の"タスクインタフェース"(task interface)をとることはもとより、ユーザである人間とハードウェア間には、"マンマシンインタフェース"(man - machine interface)、そしてソフトウェアとの間には"プログラミングインタフェース"(programming interface)がとられ、調和された整合性をはかられていなければならない。



2.2 パソコンの導入事例

いまのパソコンでも、図 2-2に示すように、各職場・対象ごとにいろんな形で導入され活用が進んでおり、積極的に使用することができる。たとえば表 2-1に掲げるように、そのアプリケーション事例はどんどん広がっている。



各職場・対象ごとに、いろんな形で導入・活用がすすんでいる。 図 2-2 パソコンの導入事例

表2-1 パソコンのアプリケーション事例(一部)

	パソコンのアブリ	リケーション事例	
提案実施状況管理	パソコン使用予約管理	デザイン管理	受注•売上統計
CRT画面プリント	コスト計算	商標登錄管理	在庫管理
部品伝票管理	超過勤務管理	実験センター室記録	¥\$管理
設計図面管理・	部品明細表作成 /	技術資料検索	価格管理
図書購入計画管理	原価集計書	社内規定配布先管理	予算•実績管理
開発プログラム管理	バグ・レポート・リスト	工数計算プログラム	支店•客先別売管理
部品管理	業務分類	経費管理	顧客管理
市場情報管理	納期管理	規格ドキュメント・ファイル	独算管理
技術情報管理	資料INDEX	設計日程表管理	駐車場管理
不良部品リスト	個人スケジュール管理	形式登録台帳	社内講習会名薄管理 :
薬品のテーマ別使用量	旅費・交通費実績管理	PAT権利評価	学会加入者名薄
モード解析	残業管理	外部団体加盟登録	外部セミナー登録管理
IDTパラメータ計算 .	JOBスケジュール管理	カタログ管理	帳票管理
事務用品管理	汎用忘備録	広告管理	住宅貸付金利計算

特に、高機能PCにおいても、すでにMulti-Window としてビジオンあるいはMS-WINなどが導入されている。UNIX、Concurrent CP/Mなどにみられるマルチ・プロセス型OSの搭載、LANなどのネットワークへの接続など、大幅に機能拡張がなされつつある。その意味において、PCもWSと同様に、その適用分野および技術的な面においてそれぞれ接近しつつある、その一側面を見せている。

2.3 オフィスターミナルとしてのワークステーション

オフィス情報をもっと詳細に分類し、その発展過程を吟味してみると図2-3のピラミッドのように、まさに遂次つみ重ねてゆかねばならないものがあることが良く分かる。上へ行けば行く程高度な技術が必要とされる。

しかし、今日のワークステーションはいまだ成長の過程にあるとはいえ、オ

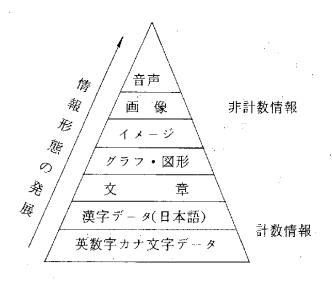


図2-3 オフィス情報ピラミッドにおける 情報形態の発展

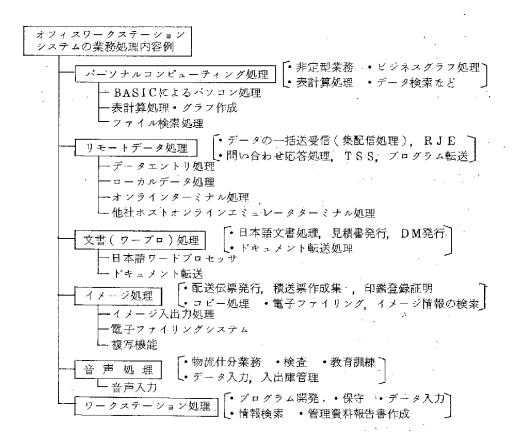


図2-4 ワークステーションの適用アプリケーション例

フィスターミナルとしてはすでに十分その能力を発揮できるようにはいたっている。

図2-4に示すように、

- (1) パーソナルコンピュティング処理
- (2) リモートデータ処理
- (3) 文書(ワープロ)処理
- (4) イメージ処理
- (5) 音声処理

など、WSのユーザはターミナルの意識をもたずに、いつでも自由にホストのシステム機能が利用できる。また、複数のWS同志から、同時に種々の業務処理を行わせることもできるという利点があるほかに、業務のプログラム開発という面でもオンライン制御といった複雑な処理が不要なので、開発が容易になっている。

だが残念なことは、国内におけるWSの普及は何よりもまだその絶対台数がまだ少なく、正しく評価できる時期にはいたっていない。現段階は、WSの普及よりも、PCの普及の方がよりいきおいよく激増している様子である。

2.4 近い将来のOAシステムの構築例

PCもWSも、OAシステムを実現するツールの一部にすぎない。

実際、オフィス業務を支援するOA機器はたくさんある。たとえば、次のも のがあげられる。

- 日本語ワードプロセッサ
- オフィスコンピュータ
- データ処理システム
- プリンタ・インテリジェントターミナル
- オフィスデータベース
- 各種専用オフィス機器
- 多機能電話機

- ・ファクシミリ
- 音声, 画像, 応用システム
- インテリジェントPBX
- LANシステム
- 通信総合ネットワークシステム

この他に、ビル管理システムなどのオフィス環境システムなどもすべてOAシステムの基盤となって構成することができる。道具(OA機器)だけでOAをリードするわけではないが、人間の英知を集約し、これらの道具でオフィスを生かす日がやってきている。

既述のように、オフィスにおけるデータ処理、文書処理および通信技術の各分野では、今後ともOA化のためにその機能の向上と多様化の方向に進み、さらにそれらの総合化の気運が高まり、次第に進展していくものと思われる。

このように、高度化・統合化の進むなかオフィスで活用されるOAシステムは、ユーザフレンドリィに立脚した使いやすさとエフェクティブであることが、何にも増して最も重要なことである。いわば、機器の大衆化、システムの大衆化を促進することがこれからOAシステムにシビアに要求される。

図 2-5は、誰でもが O A システムの恩恵が受けられるように大衆化の実現をはかった、近い将来の O A システム構築例の一つである。図示のように、(1) L A N (ローカルエリア網)形 O A システムと、(2) P B X 形 O A システムとが一体となってのアーキテクチャで構成される。

そして、一般企業においてOA機器の将来動向としては、図2-6に示すように、まず個人レベルの機器からスタートするが、やがて部、集団レベルの機器→事業・全社レベルの機器へと次第に普及していく傾向を示している。

つまり、オフィスオートメーションは、単機能レベルからLANのレベル、 LANのレベルからINSレベルへと発展していく姿が近い将来の像として熟 しつつある。

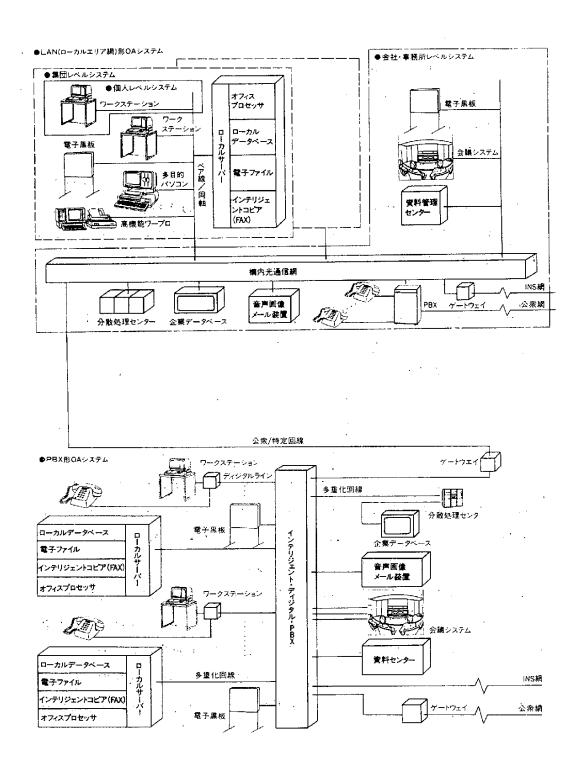


図2-5 近い将来のOAシステム構築例

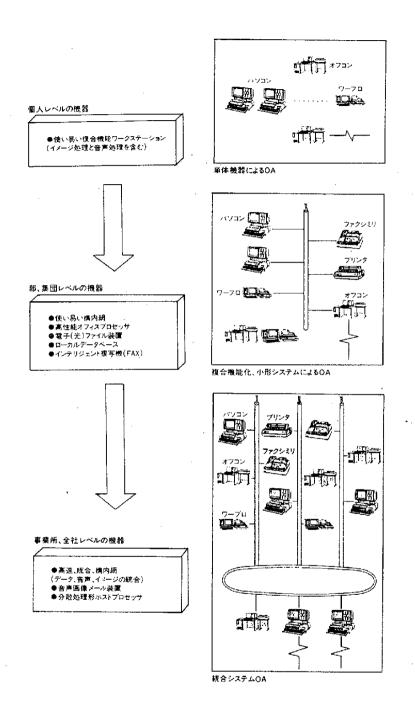


図2-6 OA機器の将来動向

2.5 IBMのオフィスシステム・オーバービュー

IBMは、数年前まで将来の製品や方針についてはとかく秘密主義に徹し、 ,84年末のように、新製品発表とともに文書による「オフィスシステム・オー バービュー」となる同社のOA戦略を詳述したことは、いたって珍しいことで ある。

同文書は、図2-7に示すように、現行の製品ベース上にいかにしてオフィスシステムが構築されるか、そのオフィス戦略となる鍵製品群を詳しく述べているといわれている。

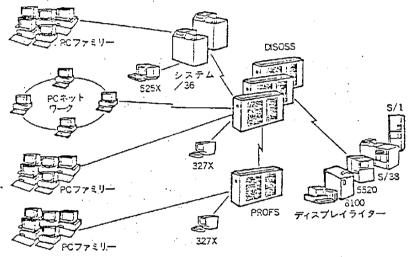


図 2-7 IBM オフィス・システム

要約すると、IBMのオフィス・アーキテクチャによって統合される製品群は幅広いカスタマーに高い機能を提供でき、その鍵となる製品はパソコンと汎用ワークステーション、PCネットワークと3274クラスター、小型ホスト機システム/36とシステム33、大型ホスト機システム370などであると述べている。

IBM社によれば、同社オフィスシステムの目的は次の三つの基本的な機能をユーザに提供することである。

- (1) ユーザが情報の作成・印刷・処理・解析を行えるプロダクティビィ・アプ リケーション
- (2) システム・ライブラリーとデータ・ベース内の情報へのアクセス

(3) 他のユーザとの情報交換

と述べている。

また、こうした機能は、システム内または組織全体に拡張可能なシステムのネットワーク内で提供され、組織内の多様なユーザをサポートし、長期的にはテキスト・データ、イメージ、音声、グラフィックなどあらゆる形で情報を処理できるようになるだろう。ことIBM社のオフィス戦略を業界の注目を集めている。

2.6 1980年代の情報処理についての一考察

オフィスシステムにおいて、現段階のエキスパートシステムはまだ実用化に 踏み出したばかりとはいえ、大いに期待したいものがある。

それに、1970年代の情報処理形態の変化を比較してみるだけでもわかるように、図2-8に示すとおりに、かつての「コンピュータ主導の時代」から「人間主導の時代」へと大きく変遷し、まさにPCとWSが主役しかも一台一人となる新しい時代の幕開けを迎えていることがよくわかる。

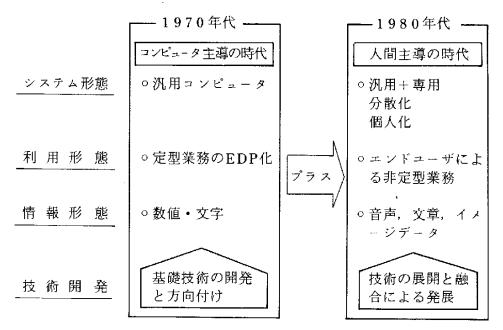


図2-8 1980年代の情報処理形態

参考文献

- (1) 「特集・拡がるワークステーションシステム」データ通信誌'84年3月 vol 16, 私3
- (2) 「沖電気OAシステム」(カタログ)
- (3) 矢矧晴一郎「スーパーパソコン革命」ダイヤモンド社 56年初版

第3章 教育システムでどう利用されているか

3.1 現 状

パーソナルコンピュータ(以下パソコンと略称する)をはじめとし、VTR、ビデオディスク、ファクシミリ、ビデオテックス、CATVなどのメディアを利用した新しい教育システムが、最近は様々な分野でみられるようになった。4月から始まる放送大学、CATVによる在宅学習、CAI、CMI、ファクシミリによる通信教育などがその例である。特に、ハードウェアのコストが増々下がる傾向と共に、家庭、塾その他の教育機関へのメディアの普及は進み、新しい学習形態が出現している。これらは、従来の学校教育を中心とした画ー的な集団教育に比べ、より個別的で、より視覚的であるという特色をもっている。

一方、教育ニーズは年々多様化する傾向にある。週休2日制の普及、出生児数の減少、家電製品の普及による主婦の自由時間の増大、高齢化などがその要因である。たとえば、学校教育においては高校生の進学が多様化し、塾などでも一律の受験指導だけでなく、コースの新設などによる展開をはかり、需要の多様化に対応しようとしている。カルチャーセンタその他の社会教育機関においても人々の志向に変化がみられたり、生涯教育的なあり方が求められてきている。

こうした教育ニーズの変化、多様化に対応していく方策としても、パソコンを中心に様々なメディアを利用していこうという動きがみられ、徐々にではあるが教育効果の上で、また教育市場の拡大の上での有効性が見い出されようとしている。

3.1.1 教育分野別利用状況

(1) 学校教育

現在一番普及しているのはパソコンであり、高等学校を中心に導入が進んでいる。文部省社会教育局発表によると、昭和58年3月現在の保有率

は, 高等学校は 5 6.3 % に達したのに対し, 中学校は 3 0.9 %, 小学校は 0.5 8 %となっている。

高等学校のうち、職業教育に関する学校(商業高校,工業高校など)では、情報処理教育が行われている所が多いが、普通科のみ置く学校では殆んど行われていない。

バソコンの利用の仕方は、CMI、情報処理教育、CAIに大別される。 このうち、CMI的利用をしている所が最も多く、中学校では70%、高 等学校では63.4%である。これに対し、小学校ではCMI的利用が多く 69.4%となっている。

パソコンの普及が進む一方、メーカによる教育ソフトの開発が盛んになり、様々なソフトが出まわるようになってきた。そのため、CMI的利用から徐々にCAI的利用へと移行する傾向にある。しかし、既存のソフトはまだまだ充実していない。生徒に密着したものを作るため、現場の先生がBASICなどでプログラムを作るなどして対応しているが、これは大変に時間もかかるし、先生にとっては負担が大きすぎる問題がある。

パソコン以外の導入例としては、ビデオテープ、ビデオディスク、ビデオテックスがあるが、これらはまだ実験段階であり、事例も極めて少ない。

昭和60年1月,文部省社会教育審議会では、パソコンの教育利用のあり方についての中間報告をまとめた。その中で、パソコンの導入は教育の基本課題と結びついた意義をもつものであること、利用方法として生徒が自分の進度に合わせて学ぶうえで、また教師が指導計画を作成したり学校の経営管理を効率化するうえで有益であることを説明している。そしてパソコンを導入するときはどんな注意が必要か、効率的に利用するにはどんな条件整備が必要かを指針としてまとめている。今後については、教職員らの研修の充実、教育用に使いやすいハードウェア、ソフトウェアの開発が大事であると指摘している。

また,現場の学校教育関係者や教育工学の立場からも,教育でのメディア利用のあり方をめぐって事例研究活動が盛んに行われるようになってき

た。

(2) 補習,進学教育

この分野では、メディアの利用形態において様々なものが出ている。営業展開の上からも学習効果の上からも、それぞれのメディアを組み合わせ、 それぞれの利点を集めることによって効果をあげようとする傾向がみられる。

また、大手塾では全国的にネットワーク化したり、ターゲットの拡大をはかろうとする動きがあり、そのメディア利用の中で、教材、出版、電鉄、電機、流通など異業種からの参入が相次いでおり、塾業界との業務提携もふえている。これらは、それぞれがもっていた教育ノウハウ、情報、スペース、販路などの条件をつないで新たな市場を開拓しようというものである。

次に、学習塾と予備校での利用状況について述べる。

学習塾では、パソコンによるCMIが普及している。CAI的利用は考慮はしているが実現に至らないというケースが多い。また、利用されている場合も、従来の方法の中で補完的な位置にある。しかし、数多くはないものの、パソコン塾、ファクシミリ塾といったメディアを積極的にとり入れた学習形態も出現してきている。

パソコン塾は、対面教育ではできない特性をもっているので、現在は少ないが今後増えていくことが予想される。たとえば、数研塾では、まもなくパソコン学習塾のチェーン展開をする予定であると発表している。

また、在宅型パソコン塾とでもいえるVAN-CAIシステムが、日本教育システム、富士通、横河北辰電機により実用化されようとしている。これは、通信回線を利用して家庭のパソコンに教材を提供するものである。 学習塾の強味として、教材データベースの蓄積があり、在宅学習を補完するための教室形式の授業が展開しやすいことが特色である。

ファクシミリ塾は、家庭にファクシミリを設置し、電話を介して家庭で 個別対応の指導をするもので、通信添削と家庭教師の長所をもちあわせて いる。塾の弱点とされていた通塾時間や教室の確保といった問題を解消している反面、コミュニケーションを大量にとれないという弱点ももっている。大阪のエミールファクシミリ個人指導システムがその代表である。

その他、最近になってCATV塾も出現した。これは日本ソフトノミックスによるもので、CATVとファクシミリを併用している。問題・答案・質問をファクシミリで送り、画面を通じて講師が説明する。在宅学習と対面教育の長所をもっている。また、河合塾でも実験的に開始を行っている。

次に予備校では、地方進出と手広く受験情報を確保する必要性から、大手ではオンラインシステムによる全国ネットワーク化をはかっている。たとえば、代々木ゼミナールではPASSオンラインシステムがあり、受験校ともつないでいる。何合塾ではPDSPというシステムにより、入試情報、在籍生情報サービスを行っている。

(3) 企業教育機関

OA化、FA化が進むなかで、事務改善、システム管理、情報処理などの知識をもつ人材が必要になってきている。そのため、従来の企業教育専門機関では、これらの専門講座、養成コースを増設するなどして対応している。また、最近ではコンピュータメーカ以外の業種からの参入で、「ニューメディア塾」「中高年向け専門のコンピュータ教室」などといった新しい教室が相次いで開設されている。これらに参加するのは企業人が多いが、ターゲットを企業の中高年、一般の学生、OL、主婦などとしている点で新しい。

教育形態としては、レンタルによるビデオテープの利用があるが、あまり普及していず、生産者側にとってはコスト高の状態である。また、個々の企業内で、ビデオより教育プログラムを作成利用するケースもあるが、編集作業はしろうとには難しいという問題がある。

(4) ライセンス取得機関

自動車教習所、各種学校、専門学校、通信教育がこの分野に入る。この

うち、専門学校が単なる資格取得を目的とするのでなく、就職へのステップとしてのものに変わってきており、中でも情報処理教育の需要が多い。 各種学校や通信教育は、放送大学やカルチャーセンタの出現により需要が 低下してきている。

(5) カルチャーセンタ

大手で、オンラインによるネットワーク化が進んでいる。また、キャプテンに実験的に参加するケースもあり(産経学園)、活性化の手段としてメディア利用が検討されている。朝日カルチャーセンタでは、ネットワークにより教室に通えない地域の人達のニーズを獲得するとともに、あわせてカセットテープ、ビデオソフト、書籍などの教材販売をすることで相乗効果をあげようとしている。

(6) スポーツ教育機関

スポーツのHow-to ものとして、スキー、ゴルフ、テニス、サッカーのビデオディスクがあるが、一般向きであり、教育用としては売れていない。

(7) 社会教育施設

パーソナルコンピュータの保有率は、図書館 3.5%, 視聴覚センタ・ライブラリ1%である。(昭和58年1月現在) 利用形態としては、図書や教材の管理に利用するケースが最も多く、次いで、マイコン教室などの社会教育活動に利用されるケースとなっている。

3.1.2 商品別実態

現在、教育のなかで利用されているメディアには、カセットテープレコーダ、VTR、ビデオディスク、パソコンなどのパッケージ系と、ファクシミリ、CATV、ビデオテックスなどの通信系・放送系メディアとがある。 これらのメディアにのる教材、システムには、どのような製品があり、どのように普及しているかについて、次に製品別に簡単に述べる。

(1) カセットテープ:英会話を中心とする語学ものが多く,その他には受験

用教材がある。販売は書店ルートによるものが多く,活字媒体と音声媒体 の結合が進んできている。

- (2) ビデオテープ:教科書会社, 訪販会社が中心となって, カセットテープ からビデオテープへの移行を進めている。教科書会社では, 光村図書出版 と東京書籍をはじめ多くの所が参入している。教材は理科や社会といった 教科用である。訪販会社による英会話ビデオ教材には, ジャパン・オブ・ザワールド「ザ・パスポート・フォー・ユー」他, 数多くのものがある。
- (3) ビデオディスク:徐々に普及している。パイオニア「高校受験教材」, タイムライフ「イングリッシュコミュニケーションシステム」, テスコ教 育システム「ビジュアル・イングリッシュ・プログラムシリーズ」がある。 出版業からの参入が目立つ。
- (4) **CAI**システム:学習塾を中心にここ1~2年で浸透し始めている。また、一般家庭にも**CAI**が認知され始めており、本格的な普及時期を迎えている。

種類としては、CAI専用機、市販パソコン用教育ソフト、オーサリングシステム、ビデオディスクーパソコン連動システム、ひらがな・かたかなLOGOがある。

これらのうち、商品化が最も進んでいるのはテープレコーダを使ったパソコンベースのCAI専用機である。湯浅教育システム「ヤルキー」、日本ユニバック情報システム「LEARN UP システム」「コンピューデント」などがある。

市販パソコン用教育ソフトでは、MSX、FM-7、PCシリーズなどを対象機器とするものが出始めている。特に最近、出版や教材会社と大手家電メーカとの連携による参入が目立っている。これは、教育ノウハウ、教材データベースを豊富にもっているという利点を生かして、活字以外のメディアにも進出し、トータルなCAIづくりをしていこうというものである。任天堂と福武書店によるファミリーコンピュータ仕様のもの、東京書籍とソニーによる「ニューホライズン」(MSXパソコン)、旺文社と

東芝による「中学TOPゼミ」(パソピア700)がまもなく発売される 予定である。

オーサリングシステムは、プログラミング言語の知識を必要とせずに、文字、図形などを作成できるシステムである。販売形態としては、専用機に組み込まれたものと、市販パソコン用に作られたソフトとがある。日本ユニバック情報システム「LEARN UP」(専用機)、日本電機「Lesson Writer」(市販パソコン用ソフト)などがある。その他、アスキーでも、PC-9801、IBM5550に対応するものが開発中である。

ビデオディスクーパソコン連動システムは、ビデオディスクの情報収容能力、ランダムアクセス機能、静止画再生などの機能と、パソコンのインターアクティブ機能とを組み合わせたシステムである。学習者の理解度のチェックや個人の能力に合った学習ができるのが特徴である。現在商品化されているのは、アイペックの英語教材用システムだけであるが、同様のシステムは、シャープやビクターによっても開発中である。

ひらがな・かたかなLOGOは、米国MITで開発されたLOGO(子供にも可能な学習用コンピュータ言語)を、ユニー、ワコムが日本語向けに開発したものである。

- (5) **CMI**システム:従来は学校や塾により、オーダーメイドで作られていたものであるが、最近ではパッケージ化された商品も相次いで発売されている。種類としては、成績管理システム、時間割編成プログラム、学習塾用成績・経営管理システム、マークカード・シートリーダがある。
- (6) **CATV**:現在は、まだ一方向のみであるが、双方向によるシステムをセコムが開発中である。また、日本ソフトノミックスでは、**CATV**とファクシミリを組み合わせた形で対話性をもたせ、塾を展開しようとしている。
- (7) VAN・ビデオテックスCAIシステム:正智塾のシステムが実用化に 入ったのに続いて、学習ソフト関連企業とVAN業者の提携により開発が

進められている。

(8) その他:予備校や通信教育の大手により、受験のための模試の成績処理、 入試情報サービスのためのオンラインシステムが構築されつつある。

3.2 今後の課題

・現状では、いい商品が少ないのが一番の問題である。数多くある中には粗悪なものもあり、そのため、最近はCAI商品の質を目指して評価基準を設けるなどの動きがでてきている。今後は、生産者側にとってはどれだけ人々に密着したものが作れるか、質の高いものを作れるかが問われていくだろう。一方、ユーザ側にとっては増々いろいろな商品が出まわる中で、何をどのように使うかの目を養うことが必要になっていく。

3.2.1 普及上の課題・留意点

(1) 生產者側

教育する側,教育を受ける側ともに人間である。しかも,教育という場合には,そこに何らかの意図なり作用があるわけで,コミュニケーションとして,微妙なものがある。この微妙さに対応する上で,機能上要求されることはかなり多くなっていくだろう。

まず、マン・マシンインターフェースがもっと人間の活動(五感・言語活動・動作など)に密着したものにならなければならない。これについては、米国のネグロボンティ教授らの研究が現在先行している。

対話性は、教育上重要である。たとえばCATVは双方向可能になれば、 教育においてかなりの利用が期待できるだろう。

また、ランダムアクセスも大きな意味をもつ。ビデオディスクで、この 点が実現され、徐々に普及しているが、個高で、再生、消去ができない弱 点が残っている。

AI(人工知能)機能は、教育のうえで実現されれば、 適用範囲がかなり広が

ると思われる。教育の場において、相手自身の理解、教育ニーズの判断、ニーズ を満たすための方法をみつけて実践する、そして評価するといった一連 の活動は、豊富な経験、的確さ、すばやい対応が要求される。

しかもこれらは全く個対個の関係の中での活動であり、経過の中で変化していく。こうした相互の個別性、流動性に対応していくには、AI的機能が必要になっていくと考えられる。

それと関連し、データベースの構築も必要になっていくだろう。教育の上では、相手に合わせていくことが大事である。相手のニーズの変化に追いついていくには、ユーザと共に情報を構築していく機能が必要である。 次に、ユーザにとって身近な問題について述べる。

まず、マシンやソフトの互換性がないために、使う範囲が限られたり、 また、そのインターフェースをとる負担がユーザにかかっていくことが多い。コストが下がる傾向にあるとはいえ、個々のユニットを組み合わせれ ば高くつく。この点の改善が望まれる。

また、実際に使う時の問題として、マニュアルがしろうとにとり、わかりにくい。マン・マシンインターフェースと同様に、教育現場(先生、生徒)向けに工夫が必要である。

(2) ユーザ側

学校教育で、教師の指導のもとにおいて使われる場合と、家庭学習のように全く個別な形で使われる場合とに分けて述べる。

学校教育における導入と利用については、文部省社会教育審議会の報告の中で指摘されているとおりであるが、これに加えて個人的な観点をつけ加えて説明する。

道具は上手に使えば効果が大きいが、使う側に、使えるだけの技量、スケールが要求される面ももっている。使いこなせば良さが出てくる。このことはパソコンなどの利用についてもいえることで、教育の場において、パソコンの特性がよく分かった上で使いこなせる人材が必要になっていく。

また、生徒に適したいい教材をみつけなければならないだろう。しかし、 良い教材があっても、教育の基本的なあり方は変わるものではないと考え る。あくまで人間が育てることで人間になっていくはずである。

たとえば、パソコンなどの場合、どんなソフト、システムを選び、どんな状況においてどういうやり方で適用するかは教師の考えによる。また、教育活動においては、教師の個人的な経験や直観といったものが大きな役割をもつ。これらの活動、コミュニケーションにおいて、どこまでマシンに負うかは相手と教師との個々の状況、関係において判断されるものであり、特に、教師がどこまで使いこなせるかにかかわってくる。

つまり、最終的には教材をも含めての使う側の人間性が問われることには 変わりがない。

パソコンなどの導入によるCAIの事例研究は、これから大事になっていくだろう。教育効果はすぐに形となってあらわれるものではなく、長い目でいろいろな見地から研究されていくことが必要である。 パソコンの場合、 視覚的なインパクトの強さで生徒の興味をとらえることは多いだろうが、それが本当の興味、 意欲となっていくかは、 また別の問題である。そして、 その評価や判断は、 教師がその生徒をどう見るかにかかわってくるなど、 かなり個人的な要素をもつ。こういった事例は、 表に出して教師以外の様々な見地からの研究により評価し、 積み重ねていくことが必要だとみなされる。

それから、試行錯誤の中で留意しなければならないのは、それらの弱点について知ることと、それによる悪影響を別なところで補うことである。相手が小さい場合にはこの点がとくに強調されなければならない。一般的には、パソコン等のマシン利用の弊害として、健康面でのことや、社会性が育たないなど、精神発達上での問題が言われているが、こういった課題は、使い方の上での工夫と、学校生活や家庭生活、地域性などトータルな環境をとらえた中で補うことにより、随分変わりうるはずだ。また、学校におけるパソコンなどの利用の仕方は、家庭でのパソコンでの学習の仕

方にも反映する面もあり、その指導そのものが学校に求められるようになっていくだろう。

次に個別学習について述べる。使い方、学習の仕方は、マシンやシステムに依存する部分が大きいだけに、どうしても学習の仕方が狭くなり、固定的になりやすい。目的に応じて上手に使うようにしたい。また、市販のソフトの中には、質の良くないものもあるため、選ぶときに注意が必要である。

3.2.2 具体例

ソフトウェアハウスが某ゲームメーカとの共同により, 幼児低学年向言語による CAI システムのプロトタイプシステムの開発に取り組んだ例を紹介する。今回は, プロトタイプシステムの段階だが, 教育用システムの開発には, さまざまのむずかしい課題があることが実感された。

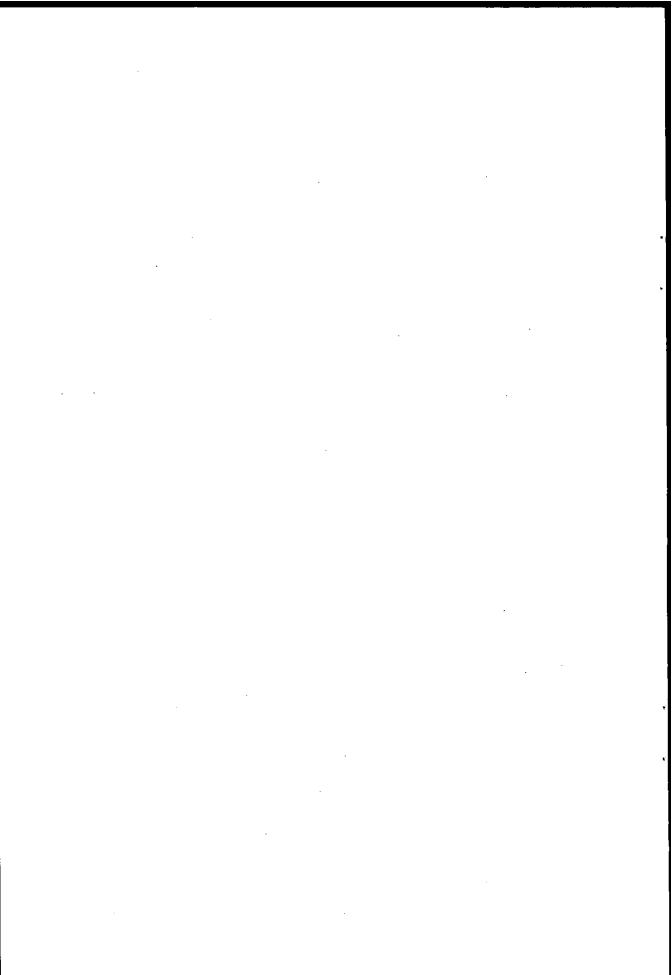
企業側の立場でみれば、アイテム自体は既に世の中に製品としてあり、種類も多くなっている。その中で、いかに新しい商品価値をつけ、他との差別・化をはかるかが、販売の決め手になる。

また、ターゲットが一般の人や小さい子供までをも含む場合には、使いや すさや機能の上でかなりの完成度が要求される。これには思った以上の時間 と費用がかかる。

このような説明を満たす商品は、単なるアイデアだけではなく、地道な蓄 積の上にでてくる。

そのためには、実現をはかる体制づくりが基本となる。技術ばかりでなく、 教育ノウハウや教材の知識両面からのタイアップが必要である。また、長期 的な開発体制の見通しも当初から必要である。成果の良し悪しは、使う側に 評価されていくわけであり、この点からも長期の見通しが必要である。

その他,実現のための体制とあわせて,販路体制などについても考慮が必要で,実現する上でと,市場展開していく上でとの両方のむずかしさがある。



第 ■ 編 知識ベースとの融合は、ワークステーション に何をもたらすか

"-			
			٣
			,

第 **■**編 知識ベースとの融合は、ワークステーション に何をもたらすか

第1章 知識ベース構築技術はどこまで進んでいるか

1.1 現 状

1.1.1 実 例

米国における若干の代表的な知識ベースシステムを例示する。

(1) Mycin

英語風のIF-THEN型の探索的ルール約400を組み込んで、伝染性血液病の診断・予防をする。質問応答を的確に行った点に特徴があった。

(2) Hearsay - I

複数の独立したエキスパートシステムを組み込んで、黒板モデルと呼ばれるデータベースの上で、約1.000語からなる話し言葉を理解する。

(3) R 1

約1,000のIF-THENルールを組み込んで、VAXコンピュータの機器構成を営業の新人にアドバイスしてみせ、有用な実用的システムとなった。

(4) Internist

内科の疾病と症候に関する100,000 の判断を組み込んで、通常の内科 医以上の知識の深さと問題解決能力をもつ最初の知識ベースシステムとなった。

パーソナルコンピュータ関連では次の例を挙げておく。

(5) Sales Edge / Human Edge / Negotiation Edge

心理学のパーソナリティ理論に基づいて、顧客と営業マンの対応のモデルを作成した。営業マン自身には86の質問を自問自答させ、顧客には50の質問によってそれぞれのパーソナリティのプロフィールを仮定し、商談促進のためのアドバイスを行う。アップルないしIBMのパーソナルコンピュータ上で稼動し、各プログラムの価格は約200ドルから300ドル

いままでのところ、10,000コピーを販売している。

1.1.2 人工知能の現状

図1-1のような枠組みのもとで現状を素描する。

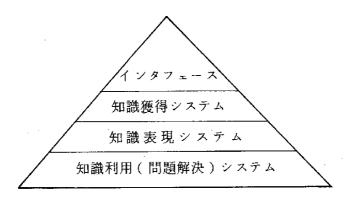


図1-1 知識ベースシステムの構成

知識利用システムは、知識システムの名称を用いる以上どんなシステムに も必要な問題解決手法を含む。現在の典型的なものに、

- . ① 記号プログラム
 - ② 述語論理
 - ③ 探 索
 - ④ ヒューリスティックス
 - ⑤ モデリング

などがある。これらの手法については後述する。

知識表現システムとして用いられている代表的なものに以下がある。

- ① プロダクションシステム
- ② 述語論理
 - ③ フレームシステム

周知のようにこれらは、対象とする問題領域と問題解決手法と密接な関連 があるけれども、特定の対象に何を用いたらよいかの適切な指針が現在ある とはいいがたい。例えば、ブロダクションシステムひとつをとっても、

- (1) マルコフ過程のようなプログラミング手法に帰するもの。
- (2) 経験則(rule-of-thumb) もどきのもの。
- (3) 脳の短期記憶モデルに相当する認知モデルに帰するもの。

のように勝手な使い方がある。また、タイプの異なる知識表現システムを統合する試みもあるが、実用的な知識ベースシステムを構成するには至っていない。

これらの知識利用システムや知識表現システムに対する典型的な批判のひとつを紹介しておく。

ルールベースのシステムが実用化されたとすれば、それは決定的に領域固有的(domain specific)であり、機能主義的である。機能主義は「御用聞き」イズムである。本来の専門家(エキスパート)は、領域を超えてAをBと見なしたり、その全体を「もっともらしさ」、「意義の深さ」、「重要性」、「見事さ」などでモニターする。「アナロジー」も上手だ。これらに反省のないシステムは果して人工知能といえるか。

なお、知識表現として、既に文章化された過去の知識の蓄積の知識ベース 化の課題がある。近い将来の自然言語解析能力の向上により、特定テキスト (例、法令文)の内部知識表現形式が問題となる。

知識獲得システムへのひとつの視点として、認知心理学との対比が第5世 代プロジェクトで試行されている。つまり、それを知識の同化、調節、均衡 化のプロセスとみなす。

知識の同化は、エキスパートがもつ知識を知識ベースへ無矛盾かつ系統的にとり入れるプロセスである。

知識の調節は、知識ベースを無矛盾かつ系統的に更新するプロセスである。 知識の均衡化には2種類ある。ひとつは、エキスバートの知識と知識ベースのそれとの過不足を補うプロセスである。もうひとつは、2つの既存知識群の整合化を図るプロセスである。

知識ベースのインタフェースはいくつかの側面をもっている。すなわち, 知識ベースとエキスパート,知識工学者,最終利用者,ソフトウェアも含め たコンピュータシステムである。このそれぞれに適切な言語が必要となる。 エキスパートには問題領域の言葉づかい、知識工学者にはLispやProlog、 最終利用者には自然言語、コンピュータシステムには各種システム記述言語 が必要である。

1.1.3 知識工学の現状

知識工学というより狭い分野で主要な3つの課題がある。第1は、もっともらしくどう「知的」機能を組み込むかである。第2は、問題の複雑さに応じて適切な問題解決手法をどう導入するかにある。第3は、これらをどう工学的に、しかも適正に扱うかの課題である。

もっともらしさは問題解決手法の効率とマンマシンインタフェースの自然 さに大きく影響される。効率向上の手法には以下がある。

- (1) 核心となる知識は知識ベースにこれを必ずとり込む。
- (2) 探索方法を工夫する。
- (3) 既存の知識や探索を再利用する。
- (4) 知識ベースの知識をできる限り、当該の問題解決に利用する。
- (5) 複数の問題解決を用意する。
 - 一般に問題解法の複雑さは次の事情に起因するからである。
 - (a) 知識ベースの知識に誤りやあいまいさがある。
 - (b) 知識が動的に変化する。
 - (c) 探索空間が大きい。
 - (d) 問題解決法の効率が悪い。

第2の適切な問題解決法に対する、常識的な方略は図1-2のようにまとめることができる。

第3の工学的課題は,エキスパートシステムの開発実用化の過程で除々に 実現されるほかはない。

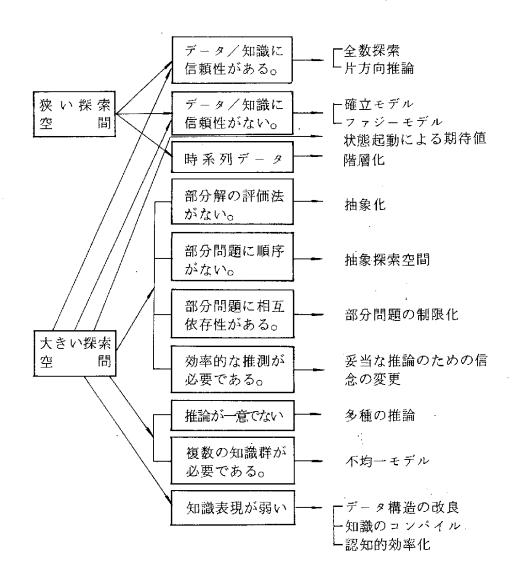
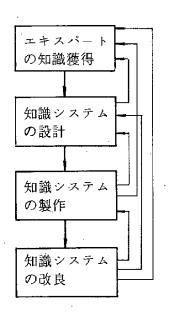


図1-2 対象問題と知識ベースの利用

1.1.4 知識ベースの構築

知識ベースの構築は図1-3のようなステップで行われる。実際のプロジェクトにもとづいて、各ステップの留意点を挙げれば次のようである。

まず、エキスパートの知識体系の問題がある。その知識体系が、広いスコープと記述的な性格さらに不確実さをもっていると、問題領域の限定や中核概念の抽出に非常に手間がかかる。場合によっては、エキスパート自身の知



問題領域 中核概念

モデリング 知識表現

知識ベース 推論エンジン プロトタイプ

改良された概念/ モデリング /知識ベース/……

図1-3 知識ベースの構築

識体系の修正や改良を伴うことがある。

プロの棋士の棋力と,自身の認識を体系化して提示する能力とは等価ではない。

知識システムの設計については、通常ルールベースのツールやLispや Prologのような言語が先にあって、それらを仮定して知識ベースを構築しようとする誤りをよく犯す。問題領域ごとに適切なモデリングや知識表現があるわけだから、個別領域ごとにそれらをまず検討することが現状の技術ではほとんどつねに正しい。

製作の方式について、当初から実用システムを目指すのは通常ほとんど無理である。プロトタイプを1年でつくり、実用システムはその後3年かけるような計画が無難である。

いったん推論方式を決めてしまうと推論エンジンを制作するのに大した手間は通常かからない。作業のほとんどは、知識ベースの構築とインタフェースの改良に費される。

知識システムの改良は、知識の同化、調節、均衡化のような、現状の技術では困難な課題を伴う。また、効率向上の課題もある。これらの課題によっては、エキスパートの知識の獲得のステップにまでさかのぼって問題領域や中核概念の変更を必要とするケースも出現する。

1.2 今後の課題

1.2.1 知識の次元

知識工学の現状と今後の課題をみるには、対象知識の次元がまず前提となる。一つの視点は、図1-4のような枠組みである。

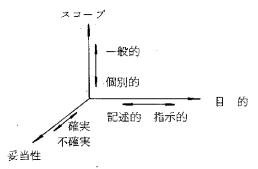


図1-4 知識の次元

原点の近傍ほど、現状の技術で問題を扱いやすく、近傍から離れるほど今後の課題となる知識工学手法が必要となる傾向にある。三つの次元による知識のタイプを表1-1に例示しておく。

表1-1 知識のタイプ

知識の種類	知 識 例
個別的, 記述的, 確実	人は動物である。
個別的, 記述的, 不確実	人は2本足をもつ。
個別的,指示的,確実	「 X が人である」のを示すのに, 「 X の親が人である」ことを示せ。
一般的,指示的,確実	P巛を示すのに,~P巛が成りた たないことを示せ。
一般的,記述的,不確実	人には超能力がある。

このような各種の知識のタイプに対して、知識工学がどのような工学的知識や方法を対置できたか、あるいはできていなかったかを表1-2に大雑把に示しておく。わかりにくい工学的知識についてだけ若干の補足を以下に行う。

表1-2 可能な工学的知識と方法

現在利用可能	今後の課題
定義と分類	一般的問題解決法
離散的な記述	アナロジー
単純な制約と不変量	メタ表現による知識
経験則	メタ知識
感覚構造の一部	複雑な帰納法
演繹法	基本原理
単純な帰納法	ノウハウの統合・分割,常識
単純なヒューリステック探索	学習機能

アナロジーは現行の知識システムで実用化されてなく今後の課題である。知識のメタ表現については、若干の試行はあるが知識群間のよいヒューリステックスはまだ実用化されていない。基本原理とは、物理学における一般相対性原理のようなものの謂である。知識表現の統合(decompiling)と分割(compiling)は、通常のプログラミング言語間のデコンパイル、コンパイルに相当する。

常識の扱いや学習機能は、ミンスキーにいわせれば21世紀にかけての課題である。

1.2.2 今後の課題

知識ベースを制作する立場から、過去の手法、現在可能な手法、近未来に可能な手法、今後長期にわたる課題をそれぞれ示せば、表 1 - 3 のようになる。

この表は、1.1.1.で例示したように各分野でバラバラに行われている個別の知識ベースシステムにおいて短期的課題、長期的課題とみなされるものを枚挙したに過ぎない点に留意されたい。

表1-3 知識工学の手法

制作ス テップ 手法		知識システム設計	知識システム製作	知識システム改良
過去の手 法	統計的学習 パーセプトロン 自己制御系 神経系	アソシエーション 一般システム理論 サイバネティックス 変換系	マルコフ過程 決定樹 ブラインドサーチ バックトラッキング	強化法 クラスタリング
現在の手法	エキスパートの情報 事例集・ 知的ルール/分類 事例からのルール 抽出	データ起動 スロットモデル 黒板モデル 相談パラダイム	Lisp / Prolog フレーム ルール ファジー論理 手続き/関係	パラメータ トレース 説 明 コンシスタンシー チェック エディタ
短期的課 題	エキスパート用 ツール 設計文書の利用 知識生成系	改良された商用の ツール群 自然言語 インタフェース ユーザインタフェース	統合知識工学ツール 高品質のプロ グラミング環境	コンシスタンシー チェックのツール テストツール 知識保守ツール
長期的課題	テキスト理解 学 習 経 験 アナロジー 帰納法	多表現システム モデルによる推論 知識コンバイラ デコンパイラ	深層モデリング言語 メタ知識 システム	経験則による学習 実験による学習 常識

第2章 知識ベースは学習的マン・マシン・インタフェース を可能にする

2.1 生涯学習を支援する計算機利用システム

学校教育を終え、社会人となった人々が、知的学習を生涯にわたって続ける 必要ないし続けたいという欲求をいだく傾向は、変化の激しい、しかし低成長 の現代社会において、益々強くなっている。

技術の激しい変化を追う必要性、配置換えや転職に必要な新しい分野の学習、 複雑化する社会構造や人間関係の理解、余暇の充実、などがその主要な動機と して挙げられる。

ところで、このような要求に応える生涯教育は、本質的に個を対象とせざる を得ないところが、学齢における集団教育と異なる。学習者個人によって、そ の興味対象、目的、バックグラウンドが極めて多様なためである。

また、時間的、経済的制約も大きく、希望者が地理的に広く分布していること、強制されないために、質のよいシステムでなければ成立しないことなどの特徴がある。従って、生涯教育においては集団教育を基本とするのには無理があり、かと言って個人教授は高価に過ぎる。そこで、ここでクローズアップされるのが、個々の学習者の、目的、興味、レベルに応じて、学習者の独習を援助する計算機利用システムである。

従来、CAI (Computer Assisted Instruction)と呼ばれる計算機利用教育システムで実用化されているものは、あらかじめ設けられた設問・評価フレームに従って学習者を誘導するものであり、学習者側からの自由な質問を許さない。その上、上述のような、学習者特性の多様さに応じるためには、個々の特性に合わせた別々の教材を用意するしかないが、これは、設問・評価フレームからなる教材を作成する際の多大な労力から考えて、あまりにも不経済である。

一方,とくに生涯学習においては、学習者は、自分が学習したい事柄を、できるだけ効率よく、また定着性よく学習したいのであって、そのためには、自分が所有している知識構造にできるだけ関連させて学習するのが最良の方法で

ある。

この2つの理由により、計算機を利用した生涯学習の支援は、学習者が、自 ら自分の知識構造をつくり上げてゆくために用いるパーソナルな知的生涯の道 具、ないし伴侶であればよいと考えられる。

ところで、できる限り広い適用範囲を有する、できる限り少数の知識の適用の組み合わせによって、自然界との関係、社会における関係を解釈し律しようとするのが人間の営みである。このことによって、数知れぬ具体的場面の一々を記憶し、検索する労を免れることができる。

しかるに,適用範囲の広い一般的な知識とその組み合わせ的適用方法を習得するには,適用例によるのが最良の方法であることは,我々,誰しもの経験からして明白である。

この場合、学習者は、一々の適用例から、一般的知識の適用法を、次第に抽象し、自らの知識構造につくり上げてゆくのである。従って、そのためには、ある程度以上の数の適用例を必要とする。しかも、適用の筋道を、学習者の要求するレベルにおいて提示してやる必要がある。

このような学習の支援を計算機を利用して行う場合,一々の適用例の全ての 筋道を記憶させるのでは、記憶容量がいくらあっても足りない。

さらに、一歩進んで、学習者が、自ら自分に課した具体的問題の解決について、適切な知識の適切な適用の筋道を探索するのを可能にしようとすると、単に適用例を記憶し、提示するシステムでは間に合わない。

さて、最近の知識工学の発達によって、問題解決を行う計算機システムが可能になろうとしている。これは、一般性のある知識や、事実的知識を記憶させたシステムに、問題を与えると、自らの持つこれらの知識の適用の組み合わせを探索して、問題を解決するもので、次章において、その現状が紹介されている。現在のところ、このようなシステムは分野を狭く限れば、かなりの能力を有するまでに至っている。

しかし、生涯学習のような、分野を問わない、あるいはむしろ、諸分野にま たがる広範な問題について、その解決を自動化するのは、容易ではない。また、 学習という視点からすれば、その必要は必ずしもない。

とくに、適用可能な知識を探索する機構については、学習者が、それなりに有する類比や抽象の能力—それこそ養うべき対象である—に依存して、システムと学習者の対話によって進められるものとすれば、最小限、各々の一般的知識について、それに、具体的適用対象の候補の表現が与えられたとき、適用の可否と、適用の過程および結果の表現を学習者に提示する仕組みがあれば、これを、問題解決の「支援」に役立てることができる。

まず、例題のストックについていえば、計算機に、一般的知識はもちろん、 例題について、その解決、即ち知識適用の筋道を、最小限の表現で記憶させて おき、学習者が必要としたときに、適用の筋道を段階的にその場で組み立てて 提示するのに、上述の仕組みを用いることができよう。

この最小限の表現ということは、このような、知識と例題のサブセットを、 大きなデータベースから、通信回路によって、パーソナルなシステムにトラン スファーして用いるような場合には、回線使用料の節約につながる。

さらに、学習者が提起した問題に対しては、システムは後向き、あるいは前向きに、適用可能な知識のメニューを示し、学習者が、その中から選んで指示した知識について、適用(instantiation)を行い、その結果を学習に提示する―もちろん、「学習的配慮」から、まず、学習者に結果を評価ないし予測させた後でなければ提示しないという方略をとることもできる―学習者はこれを見て、判断して、次に適用可能な知識を、同様にして探ってゆく。この際、システムが持っている適用例を提示させ、これを参考にすることによって、強力な支援が得られよう。

以上のようなシステムを実現するためには、

- (1) 学習者が不案内な分野においても、必要とする一般的知識、あるいは例題に、対話的方法によってアクセスすることのできる、学習検索機構
- (2) 学習者に対して、知識や適用課程の、必要な部分を、その場で組み立てて提示する、発生的提示機構
- (3) このような検索と提示を可能にする汎用の知識表現

が、必要であり、さらに、パーソナルな知的生産の道具ないし伴侶としては、

(4) 学習者自ら、知識、あるいは、問題を解いた過程を、自分の知識ベース に付け加えることができ、こうして、自らの知的スットクを増大させてゆ けること

が望ましい。

このように、学習者サイドにおいて、高度な処理ならびにパーソナルなデータベースの作成を行う必要がある。一方、あらゆる分野の知識ベースを1つのパーソナルなシステムで持つことはできないし、生涯学習の性格からして、同じ分野でも、異った内容の知識ベースを提供する多くのproviderがあってよい。

そこで、このシステムは、学習者側に、一定以上の能力を持ったパーソナルな計算機を置き、多数のパーソナルなシステムと多様な知識データベースを提供する多くのソースシステムを、ネットワークに組み入れた形態をとることになるだろうと思われる。

この場合、ソース、ならびに、学習者ノードの知識ベースにおける、さまざまな知識の表現の枠組みを、多数の分野やテーマに対して共通なものとし、そのことによって、知識表現に基づく検索、提示、ならびに知識ベース編集のブログラムを汎用なものとし、こうすることによって、パーソナルな部分のヴァリエーションを犠牲にせずに、低価格なシステムを表現する必要があるだろう。

2.2 学習の対象としての知識の表現

J. Piajet によれば、人間は、その発達段階において、自分をとりまく世界を自分の認識の内に同化するための図式、あるいは、枠組みを、はじめは、大摑みなものから、次第に分化させ、調節してゆくことによって、より精致な認識を可能にしてゆくという。また、外界から取り込んだ表象に対するさまざまな操作、論理操作、などを発達させてゆくことによって、同化と調節の能力を組識化してゆくという。

このような,認識の発達の段階に即して考えるならば,図式ないし枠組みとい

われるものの多くは、外界との相互作用と成人の模倣によって習得されたものであって、内部的に厳密に定義されることによって作られたものではないであ ろう。

論理的および数学的概念については、例外であって、これは、カテゴリ化された表象に対する操作と検証に関する図式であり、矛盾のない操作と検証の手段を組織化したものである。それは、人類が共有する経験によって試されては、修正され、伝達によって築きあげられてきた、言わば、内省的に再構成された認識作用の枠組みであり、厳密な定義によって、閉じた世界をつくることにより、初めてそれを操作することが可能となる。

一方,自然科学をはじめとする諸科学とよばれる方法においては,極くブリミティブな図式は,経験的なものとして前提としつつ,一回性の個別的現象ないし事実の集まりから得られる表象のカテゴリ化ないし概念化と因果性を帰納するという操作(この操作自体,既得の図式に依存する)に基づき,そこから論理的,ないし数学的な操作によって個別的現象の予測を可能にするような仮説あるいはモデルとよばれる図式を求め、これを,人間の活動の指針にしようとするものであると考えられる。

そうであってみれば、諸科学を学習するとは、このような図式ないし枠組みの間の階層構造と、それを具体的な認識の場で活用する為の帰納的、演えき的ないし数字的操作の方法に関する知識を学び、それに習熟することであるにちがいない。

さて、上述のように、多くの現実的な表象ないし概念は、他の概念を用いて 厳密に内包的に定義されることのないまま、その識別に役立つ複数の属性とと もに、登録されているものが多いと考えられ、一方、それらの間の階層的ある いは構造的関係(定義できる概念については、ここで定義することを含む)と、 それを、検証する手続きや、具体的場面において評価する手続きが知織とよば れるに相応しいものと考えられる。そして、この知識の記述は、漠然としたも のから、厳密に論理的なレベルまで、多様な記述を、許すものでなければなら ない。このような観点から、概念の枠組みと、概念の間の関係の記述は、分離 しておくことが必要である。そこで、われわれは、学習の対象としての知識の表現を、つぎの二つの記述に分けることとした。

- 概念的フレームユニット (Conceptual Frame Unit 略して CFU)
- 知識記述ユニット (Knowledge Desciptinn Unit —略して KD U)

CFUの各々は、物、事、行為、操作、手続きなど、ある概念と、その属性の組み合わせについての情報を与えるもので、概念自身を表わすフレーム識別スロットと、一般に複数の属性に対する属性スロットを持つ。

KDUは、CFUを、述語とみなして、これを引用することにより、CFUの間の階層的関係、構造的関係あるいは、ある操作の前提条件と結果など、さまざまな関係を表わすとともに、必要に応じて、そのような関係の妥当性を証明する手順を書いたり、あるいは、引用されたCFU('s)のコンテキストの下に、そのある変数を評価する手続き、さらには、その手続きを決定する問題の解法を記述する。

2.2.1 CFU

先に述べたように、CFUの各々は、物、事、行為、操作、手続きなど、ある概念と、その属性の組み合わせについての情報を与えるもので、概念名をそのCFUの名前とし、概念自身を表わすフレーム識別スロットと、一般に複数の属性に対する属性スロットを持ち、各スロットには、変数または、値を与えることができる。

CFUの各スロットは、それぞれにある概念を表わし、その値は、そのスロットが表わす概念の外廷の要素である。

たとえば、

CFU1. 鋼塊を圧延する

self *, 鋼塊*, 製品*, 圧延システム*

は、CFUであり、「self」は、フレーム識別スロットであって、ある時、

ある場所,ある圧延システムによるある鋼塊の圧延事象を値とする。「鋼塊」「製品」、「圧延システム」は、属性スロットの例である。

属性スロットは、フレーム識別スロットに対して関数依存関係があるものとする。すなわち、各属性スロットの値は、フレーム識別スロットの値がきまれば、一意に決まる筈のものとする。この値は、集合であることも許す。その場合には、当該スロットの外延は、集合族になる。

例えば、CFU1において、一回性の圧延事象であるselfの値に対し、 その時圧延されている鍋塊は、ひとつに決まるし、結果としてできあがる製品も一意に定まる。

また, 例えば,

CFU2.数ベクトル空間

self # (ベクトル*), 次元#, 基礎体# (#と*の使い分けについては、後出の系参照 C F U を見よ)

において、selfのとる値は、特定の数ベクトル空間であって、それは、ベクトルの集合である。

すべて、属性は、定義されたフレームの文脈によって規定され、同じ名前 の属性でも、フレームが異なれば、異なる外延をもつ、 例えば、

CFU3.ハミング符号方式

self井(情報ブロック米,符号語※),

符号器并

ح

UFU4. Reed Solomon 符号方式

self#(情報ブロック*,符号語*),

符号器井

において,名称の同じスロットどうしでも,すべて,互いに排反な,外延をもつ。

次に、あるCFU、F1のスロットAのあらわす概念に対して、別のCF

U, F2のスロットBの表わす概念が、その上位概念であるとは、F1の下でAに与えられる任意の値のそれぞれが、F2の下でBに与えられる値でもあることをいう。逆に、F1のAは、F2のBの下位概念であるという。 互いに他の上位でもあり下位でもある概念は、同値であるという。 例えば、

CFU5. 角錐台

selt*P. 元の維体*C1, 取り除かれた錐体*C2

CFU6. 部分と全体

全体*C1, 部分*C2, 残りの部分*Pにおいて, CFU6-「部分」 スロットは、CFU5-「取り除かれた錐体」スロットの上位概念である。 ただし、こうしたCFU間の関係の記述は、全て、KDUで行われる。

とくに、F1のスロットAが、F2のフレーム識別スロットを、その上位概念とする場合には、F2は、F1ーAスロットの上位フレームであるといい、F1は、F2の属性概念の全てを、F1の文脈規定の下に、自らの属性スロットとすることができる。このことの妥当性は、属性スロットの、フレーム識別スロットに対する関数依存性によって保証される。ここに、F1の文脈規定のもとに、といったのは、F1のスロットである以上、スロットAを通じてF2のフレーム識別スロットの外延をAのそれに一致させ、その結果として、F2の属性スロットの外延が狭まったものを、F1の属性とすべきであるからである。

この考えは、上位概念の属性は、下位概念に承け継がれるという周知のものであるが、上述のように、下位概念の文脈規定の下にこれを引用するという意味で、遡行引用と称することにする。

さて、先に、ひとつのフレームにおいて、属性スロットは、フレーム識別スロットに関数依存していなければならないとした。それは、一つのフレーム Fに属する2つの属性スロットA、Bを考えたとき、両スロットは、フレーム識別スロットSを介して対応をつけられるが、もし、Sのとる一つの値に対し、一方の属性Aの値が二つあったとすると、属性Bの値は、そのどち

らか一方と組になるのか、それともそのいずれと組にしてもよいのかが明らかでなくなる。このようなアンビギュイティは、推論を誤らせるもとになるので、避けることとしたのである。このことは、CFUが、関係データベースで正規形を要求する考え方と同様な立場に立つことを意味する。

遡行引用は,属性スロットの,フレーム識別スロットに対する関数依存関係を保証するかぎりでの,引用なのである。

例を挙げよう。

CFU5. 鋼塊の圧延

self*A,鋼塊*K,製品*S

CFU6。鋼塊

self *K, 炭素含有率 *C

CFU7。工業製品 「

self *S, 用途*U

において、CFU6。は、CFU5一「鍋塊」スロットの上位フレームであり、CFU7、は、CFU5一「製品」スロットの上位フレームである。

従って、CFU6。の「炭素含有率」スロット、CFU7。の「用途」スロットは、CFU5。から、その文脈規定の下に、遡行引用することができる。

このようにして遡行設定されるスロットについては,

CFU5. 鋼塊の圧延

self*

鋼塊*:鋼塊(炭素含有率*)

製品*:工業製品(用途*)

のように, 遡行パスを記して CFU リストに記載する。

尚、遡行については、引用するスロットが固有のスロットとして定義されているCFUに遡ることとする。たとえば、

CFU8。女教師

self*:人間(住所*,年令*,生年月日*)

:女性(子供の数*)

: 教師(教師としての収入*)

のように,

さて、CFUには、つぎの二種類のものがあり、その取り扱いが異なる。

- 単純CFU
- 系参照CFU

一般に、系とは、要素の間にある一定の関係があるとか、その間にある再 現性のある操作が可能である場合の、関係ないし操作と、その対象たる全て の要素を込みにしたものをいう。ここに関係というのには、共通の属性を持 つといったものもふくまれる。

単純CFUは、個体としての物、あるいは、個別の事象、行為、操作など、個別の対象をそのselfスロットの値とし、それに対する属性値を属性スロットの値とするCFUである。値自身は、系であってもよいが、その要素を参照することは、できない。したがって、このCFUは、一階述語とみなされる。

これに対し、系参照 CFUは、あるカテゴリの系の集まりについて定義され、その self スロットが系を値(井を付す)とし、系の属性値(井を付す)を属性スロットの値とする場合であり、かつ、self スロットの値である系のそれぞれについて、その要素(*を付す)を引用することもできる。したがって、これは、二階述語である。この CFUは、その self スロットをインスタンス化することによって、単純 CFUになる。

[単純CFUの例]

CFU9。鳥

self*, 羽毛*, 翼*, くちばし*

CFU10.美しいと思う

self*,主体*,対象

CFU11. コンピュータでAV機器を制御する self*. コンピュータ*, AV機器*

[系参照CFUの例]

CFU12,群

self # (要素*), 二要素に対する演算#

CFU13. 数ベクトル空間

self #(ベクトル*),基礎体井,次元井

CFU14。 二要素に対する演算

self # (要素 1 *, 要素 2 *, 結果*)

CFU15。 二要素に対する可換な演算

self # (要素 1 * , 要素 2 * , 結果*)

--これは、CFU14- $\lceil self \rfloor$ の下位フレームである。

CFU16. 積分する

self #(被積分関数*,積分区間*,結果*)

CFU17. 多項式の割り算

self #(割られる多項式*,割る多項式*,商*,剰余*),係数体

CFU18。符号化

self # (メッセージ*,符号系列*),符号#,符号器#

CFU19. ディジタル伝送

self #(送信シンボル列*,受信シンボル列*),伝送誤り特性#さて。ここで、例えば、上記、設計モデルとしての系参照CFU符号化と、現実に目の前で行われている事象に対応した単純CFU符号化事象との関係が問題になる。現実には、符号器に発生した故障の為に、正しい符号化が行われないかもしれない。実際、現実の符号器が、設計モデルとしての符号化方式に則って作られたのであって、その意味で、系参照CFU符号化は、単純CFU符号化事象のスロットとして引用されるものであると考えられる。工学上の設計の問題の関心事は、ほとんどが、モデル化された対象に対するモデルとしての系の考案なので、多くの場合に、系参照モデルが用いられると思われる。

[系参照CFUの単純化の例]

CFU12 群→

CFU20. 5次置換群の要素 (self*)

CFU18 符号化→

CFU21、ハフマン符号の符号化(メッセージシンボル*、符号語*) また、たとえば、パーソナルコンピュータシステムのカタログというCF Uを考えるとすると、これは、一つの型名のパソコンシステムの集合を値としてとるような、selfスロットをもつ。そこで、これは、系参照CFUであって、型名を、たとえば、PC9801システムとインスタンス化することによって、単純CFUに落とすこともできる。たとえば、次のようになる。CFU22、パーソナルコンピュータシステムのカタログ

self # (パソコンシステム*S) CPU # , 主記憶サイズの範囲 # (*M) , 接続可能なディスクユニット # (*D) \rightarrow

CFU23。NEC-PC9801システム

self*S, 主記憶サイズ*M, 接続ディスクユニット*D

このような対応を用いれば、KDUに記述されたPC9801 の一般的特性をインスタンス化されたPC9801 システムになんとか継承できないことはない。しかし、このような場合は、むしろ

CFU24。パーソナルコンピュータシステム

self*S。CPU*,主記憶サイズ*M,接続ディスクユニット*D という単純CFUを用いてCFU23。NEC-PC9801の一般的特性 をKDUで記述する方が、Sをインスタンス化することによって、自然にこ の特性を継承できる。

また、CFUとして登録する概念は、必要最小限にすべきである。たとえば、

CFU5。鋼塊の圧延self*,鋼塊*,製品*があれば。

CFU圧延される鋼塊

self*

は、いらない。CFU5-「鋼塊」が後者のselfそのものであるからである。

また:

CFU18。符号化

self #(--),符号方式#,符号器#

があれば、設計仕様としての符号器については、最早別のCFUをたてることは、必要がない。

また, 例えば,

CFU25. 国

self*,首都*:都市,所属大陸*

CFU26. 隣接する

self*, 対象1*, 対象2*,

という二つのCFUがあって、そのインスタンスとして、

CFU25(アルゼンチン。ブエノスアイレス, 南米)

CFU26(アルゼンチン, チリ)

CFU26(アルゼンチン, ウルグアイ)

CFU26 (チリ, ブラジル)

などが置かれていれば、国フレームに隣接国というスロットを置くまでもな く、

国(self*X).

隣接する(対象{*X,*Y)}

という質問にこたえることができる。

主観的CFUについては、判断主体をスロットに置くことができるように しておくと面白い。たとえば、

CFU10。美しい

self*, 主体*, 対象×

においては、selfは、ある主体が、ある対象について、美しいと思う事実を値とする。

2.2.2 KDU

KDUは、CFUを引用する事によって、CFUの間の関係を記述したり、CFUに関するインスタンスとその属性値のtupleからなるテーブルを記載する。また、あるCFU('s)のコンテキストの下に、ある概念の値の評価手続きを記述する。さらに、関係の妥当性を証明する問題(証明問題)、評価手続きを導く問題(決定問題)の解法を記載することもできるようにすることによって、学習者の、問題解決を支援する。

KDUは、つぎのようなカテゴリに大別される。

- フレームリンクKDU
- フレームインスタンスKDU -

2.2.2.1 フレームリンクKDU

CFUのスロットに変数あるいは、値を代入したものを引用し、その間の 関係を記述するものである。

また、その関係の妥当性を証明する手順や、ある変数を評価する手続きを 導く手順の筋道を記載することができる。

このカテゴリのKDUは、第一義的には、次に例示するような論理的関係を表わす。

なお、限量子が示されていない変数は、全て、存在称とする。また、フレーム識別スロットに対する属性スロットの関数依存性の故に、限量子は、フレーム識別スロットにおかれる変数だけについて付ければすむ。

以下、いくつかの例を示す。

KDU1.

O。 雷鳥(self * P, 羽毛* F)

- 1. 生活する(主体×P,季節***冬*)
- 2. 白い(対象*F)

a P. $0 \cdot 1 = 2$

KDU 2.

 $0. \nu - \nu \text{ (self } *R)$

1。鋼塊の圧延(製品*R)

aR. 0 = > 1

KDU2。のコンテキストにおいて、後の引用に備えて、"レールの製造" (レールは全て圧延で造られるとすれば、圧延によるという限定は、不要) というCFUを作ることもできる。その場合には、KDU2を変更して、 KDU2.

- 0. レールの製造(self * S, 製品レール* R)
- 1. レール (self * R)
- 2. 鋼塊の圧延(self * S, 製品 * R)

aS 0 < = > 1.2

aR, 1 = 2

というKDUにする。そして、「圧延システム*」、「原鋼塊*」などのスロットは、"鋼塊の圧延"CFUへの遡行引用スロットとして、また、レールの圧延に用いられる原鋼塊の炭素含有率、製品としてのレールの用途、などについても、"圧延によるレールの製造"CFUのなかで、遡行引用により、次のように定義される。

CFU26. レールの製造

self*:鋼塊の圧延(ミル*。

原鋼塊*:鋼塊(炭素含有率*))

レール*:工業製品(用途*)

このように、遡行引用によるスロットをCFUのなかで定義することにより、KDUにおけるスロットの対応づけを極めて簡略化することができる。 そして、このCFUを引用するKDUは、そのスロットを、遡行引用である かないかとは、無関係に、同質のものとして引用できるので、もし、階層構造に変更が生じた場合でも、CFUでそれを吸収してしまえば、引用KDUには、何等の変更を行なわないで済むのである。

次に掲げる二つのKDUの例も、そのなかに引用してある二つあるいは三つのCFUのANDの存在から、第一行のCFUが導入されたとみることがでる。そして、遡行引用によるスロットの導入が極めて有効であることがわかる。

KDU3.

- 0. 女教師(self*W)
- 1. 教師(self*W)
- 2. 女性(self*W)

aW, 0 < = > 1. 2

(eW, 1, 2という情報は,上式に含まれているとみなす)

CFU8。女教師における遡行引用については、既に示した。

KDU4.

0. コンピュータでAV機器を制御する

(self * P, コンピュータ* C, A V 機器* A)

- 制御する(self*P,動作主*C,対象*A)
- 2. コンピュータ (self * C)
- 3。AV機器(self*A)

 $aP \cdot 0 < = > 1 \cdot 2 \cdot 3$

CFU。コンピュータでAV機器を制御する

self*

コンピュータ*:コンピュータ(オペレーティングソフトウエア*,主記憶 サイズ*, CPU*)

A V 機器*: A V 機器(製造元*,方式*)

.CFUにおける、スロットの導入は、KDUとは関係なく行なえるのであるが、KDUにおける記述を参照して、それまで、固有スロットとしていた

ものを遡行引用にしたり、KDUの変更によっては、逆に遡行をはずしたりする。また、単純な遡行引用による属性だけに限らず、フレーム識別スロットに対する関数依存性にのっとってきめ、上位概念との対応をKDUでつけておけばよい。例えば、"陶磁器を焼く"と"器に西洋料理を盛る"という2つのCFUから"西洋料理を盛る為の陶磁器を焼く"というCFUを作ったとした場合には、次のようなスロット間対応ができる。

KDU5.

- 0. 西洋料理を盛る為の陶磁器を焼く(self*Y, 陶磁器*T, 作者*S, 料理*R)
- 1。陶磁器を焼く(self*Y,陶磁器*T,作者*S)
- 2. 器に西洋料理を盛る(器×T,料理×R)

$$aY$$
, $0 < = > 1$. 2

次の例は、"構成する"というブリミティブなCFUを用いて、スロット間の、構成的関係を記述したものである。

KDU6.

- 0. 符号通信(符号化通信路井CC,符号器井E,ディジタル通信路井DC, 復号器井D)
- 構成する(結果*CC,要素{*E,*DC,*D})
 aCC。0=>1

KDUによって、構造的ルールを記述することもできる。例えば、KDU7。

- 0. 全体と部分(全体*X,部分*Y)
- 1. 全体と部分(全体*Y,部分*Z)
- 2. 全体と部分(全体*X,部分*Z)

aX. 0. 1 = > 2

KDUでは、CFUの否定を用いることもできる。

次の例では、"あらゆる類の脚を考えたとして、そのいずれを部分とする 全体でもありえない動物が存在する事をいい、へびは、その仲間である"こ とを記述している。

KDU8.

- 0. へび (self * S)
- 1。動物(self*S)
- 2. 脚(self*L)
- 3. ~ 部分と全体(全体*S, 部分*L)

$$eS$$
, 1. [aL , 2 = > 3]

aS,
$$O = > 1$$
, (aL, $2 = > 3$)

つぎも否定の例である。

KDU9.

- 0. 飛べない鳥(self * BN)
- 1. 鳥(self *BN)
- 2. ~飛べる(主題*BN)

e B N. 1. 2

a B N, 0 < = > 1. 2

KDU10.

- 0. 既約多項式(self * I R, 係数体 * K, 次数 * N)
- 1. 多項式(self*IR,係数体*K,次数*N)
 - $2.1 \leq M \leq N$
 - 3。多項式(self*DV,係数体*K,次数*M)
 - 4. ~ 多項式が多項式を割り切る(割られる式*IR,割る式*DV,係数体*K)

a IR, 0 < = > 1. [aM, aDV, 2. 3 = > 4]

次は、系参照CFUが引用されているKDUの例である。

KDU11:

- 0. 多項式が多項式を割り切る(self*S。割られる多項式*A,割る多項式*B,係数体*K)
- 1。多項式の割り算(self#,割られる多項式*A,割る多項式*B,剩

余*R, 係数体#K)

 $2 \cdot R = 0$

a S. 0 < = > 1. 2

これに見られるように、手続きを参照するCFUであっても、一般のCFUと同じように、引用され、結局、その手続きCFUを記述するKDUを介して推論に参加できるのである。

やはり,系参照 C F U を引用する例をもうひとつ掲げる。

KDU12.

- 0. 符号化通信方式(self #T,送信符号系列CT,受信号系列*CR,符号#CS,符号化方式#E,内部ディジタル伝送方式#DT,復号方式#D)
- 1. 情報伝送方式(self #T, 情報源からのメッセージ*MT, 情報届け 先へのメッセージ*MR)
- 2。符号化方式(self # E, メッセージ*MT, 符号系列*CT, 符号#CS)
- 3. ディジタル伝送方式(self×DT,送信シンボル列*CT,受信シンボル列*CR)
- 4. 復号(self #D, 符号系列*CR, メッセージ*MR, 符号#CS) aT, 0<=>1.2.3.4

この例では、一つの手続きを表わす系参照CFUを、他の幾つかの手続きを表わす系参照CFUの接続として表わしている。

系参照 C F U を 用いなければ絶対に記述ができない例として、次のものを 掲げておく。

KDU13.

- 0. 演算について閉じている集合(self #G.要素*,演算#OP)
- 1. 集合(self #G, 要素*X)
- 2. 集合(self #G. 要素*Y)
- 3. 二要素に対する演算(self #OP, 要素1*X, 要素2*Y, 結果*

R

4。集合(self #G,要素*R)

a G,
$$0 < = > [$$
 a X, a Y, $1 < 2 = > 3$, 4]

やや、特別な例として、"地球"のように、インスタンスではあるが、これといっしょにして外延を形成し得るインスタンスが極めて少ない場合などのために、直接そのインスタンスを記述するフレームリンクKDUを置くこともできる。

KDU14.

- 0。地球*E
- 1. 球形物体(self*E, 半径×6350Km, 質量*ton, 表面*S)
- 2。包括的部分と全体(全体*S,部分{*SE,*LA})
- 3. 海(self * SE)
- 4。陸(self×LA)
- 5。部分と全体(全体*SE,部分{*"太平洋",*"大西洋",*"インド 洋",*"地中海",*"北海",*"北氷洋",*"南氷洋")
- 6. 部分と全体(全体*LA, 部分{*"ユーラシア大陸", *"アフリカ大陸", *"オーストラリア大陸", *"北アメリカ大陸", *"南アメリカ大陸", *"南極大陸")

0 = > 1. 2. 3. 4. 5. 6

フレームリンクKDUとしては、論理的表現をとり得ないような漠然とした関係も許容することによって、常識のレベルから入って、段階を追って、知識を獲得することを可能にしたい。たとえば、

KDU15.

言葉による伝達(self*COM,送りたい情報*JT,話者の発する言葉*LT,聞き手の受け取る言葉*LR,聞き手が受け取る情報*JR,伝達における優乱*DT)

通信路符号方式(self #COM,送信二値情報*JT,送信符号系列*LT,受信符号系列*LR,受信二値情報*JR,伝送路における擦乱#D

T)

のように比論的な関係について,変数の共有の形で,対応をつけることを許す。後で3.に述べるように,学習者に対しては,各KDUに対応した自然言語と図を用いたインターフェースを取るので,このような対応が比論的であることを断わるようにする。

フレームリンクKDUを用いた推論のしかたは、通常の述語論理における ものと同様であるが、属性スロットは、すべて、フレーム識別スロットの関 数と考えてよいので、スロット対応をフルに生かすことによって、ユニフィ ケーションは、単純化される。

」以下,記述上のいくつかの便法について述べる。

-集合化

買う(場所* "カドの八百屋", 時* "昨日", 対象* { X }) りんご(self * X)

においては、"昨日、カドの八百屋で買ったりんご"が一般に、複数であることを表わしている。

一部分

矩形(self * S[点{*P, *Q}], 面積*A)

においては、矩形Sの部分としての二点P、Qを引用する書き方が示されている。

—関数

線分(端点*(X,Y),長さ*L(X,Y)) 等しい(L(A,B),L(A',B')

フレームの属性スロットは、selfスロットあるいは、他の属性スロットの関数である。よって、関数の形の引用を用いるのが便利な場合には、上記のように、フレームの引用によるローカルな関数の定義を、コンテキスト中に示して、これを用いることができる。

定理とか、証明問題の類も、このフレームリンクKDUで記述される。そして、一般に、証明をすることのできる関係については、証明手順の筋道を、

そのKDUのなかにおいて、利用するKDU名を引用することによって記すことができる。じっさい、証明は、前提と結論について記述しているKDUのコンテキストのなかでないと、表現できない。

たとえば」

KDU16.

- O, 平面(self *S[直線{*OA, *OB, *OC}])
- 2. 直線と直線が垂直(直線 * O B, 直線 * O P)
- 3。直線と直線が垂直(直線*0C,直線*0P)

a S, a OA, a OB, $0 \cdot 1 \cdot 2 = > 3$

if requested then call proof

この証明手続きの筋道の表現については、付図3に示してある。

また、あるKDUにおいては、一つの評価手続きを表わす系参照CFUを、 与えられるもの、満たすべき条件、そして、求めるものを記述することによって、定義することができる。

また,評価手続きを導く問題の解決の筋道を記載することができる。 例えば,

KDU17.

- 0. 多項式(self * A, 基礎体 * K)
- 1. 多項式(self * B, 基礎体 * K, 次数 * N)
- 2. 多項式(self * Q, 基礎体 * K)
- 3。 多項式 (self * R, 基礎体 * K, 次数 * M)
- 4. $A = B \cdot Q + R$, $(0 \le M < N)$

a A, a B, $0 \cdot 1 = 2 \cdot 3 \cdot 4$

eval:多項式の割り算(self #S(割られる多項式*A,割る多項式*B→商*Q,剰余*R),基礎体#K)

if requested, then call solve for (A, B \rightarrow Q, R) KDU18.

- 0. 錐台(self*T,上底の面積*SU,下底の面積*SL,高さ*H, 体積*V)
- 1。関数関係(self #, 変数 {*SU, *SL, *H, *V})
- 2. $V = (SU + sgrt(SU \cdot SL) + SL) \cdot H/3$

 $aT \cdot 0 = > 1$

 $aT \cdot 0 = > 2$

eval:錐台の求積(self*(上底の面積#SU,下底の面積#SL,高 さ#H→体積#V),錐台*T)

if requested, then call solve for (SU, SL, $H \rightarrow V$) この後者の決定問題の解法については、付図 2.にその筋道を表現してある。さらに、料理法、測定法、機械器具の取り扱い法、コンピュータのコマンドシステムの使い方、など、操作の系列を表現し、学習者、を支援するには、ちょうどRSTRIPS [6]における表現とおなじように、

KDU 操作名

前提条件一(CFU('s)で記述)

結果——(CFU('s)で記述)

という形のストックを作り、これを、ちょうど、問題解法の手順の記述と同じような形式で引用して、操作の系列を記述し、できあがった操作を再びこの形に登録して、また別の操作の記述に引用する。こうすることにより、操作の階層が出来てゆけば、ある操作の内容とその使われ方の検索ができ、学習者が、自分の課題に適した操作の系列を作り出すのを支援することができる。

2.2.2.2 フレームインスタンスKDU

このカテゴリのKDUは、あるCFUについて、そのスロットに与えることのできる値のセットを例示するものである。

ひとつのCFUについて設けられた外延記述KDUにおいて、このような値のセットは、いくつ置いてもよい。

いくつかの例を挙げよう。

KDU19.

買う(self*1,場所*2,日付*3,対象*4,購入者*5)

(@1, 東急SC, 1985年10月10日, <りんご>,)

(@2, かどや, 1984年5月8日, <ブラウス@1>, 降)

KDU20.

商店(self * 1, アドレス* 2, 経営者* 3, 取り扱い商品* 4)

(かどや,美しが丘4丁目,角野フク,<洋品>)

(東急SC, あざみの2丁目, 後藤敬治, <食料品>)

K D U 2 1

果物(self * 1)

(<りんご>)(<製>)(<西瓜>)

KDU22.

りんご(self*1,品種*2,最盛期—季節*3,特産県*4)

(@1,紅玉,秋,長野県)

KDU23.

ブラウス(self * 1, サイズ* 2, メーカー* 3, 素材* 4)

(@1, LL, 万紡。木綿100%)

原理的には、スロットの値として、個別のユニークな対象を対応させるべきであるとしても、そこまで、識別可能な呼び方をすることができない。あるいは、そのようにする意味がなくて、それを値としてとりうるCFU名を示せば済むことも多い。このような場合、そうしたCFUが知識ベースのなかに存在するならば、上記のように、<>を用いて、それで括られた名前を持ったCFUのselfスロットがとる値の一つであることを示す。

こうした場合、そのフレームに対する外延記述KDUを参照することによって、さらに詳しい属性値のサンプルを得ることができる。

むろん,異なるフレームの間でユニークな対応をつけようとする場合には, 対応をつける限りにおいてユニークに識別可能な記号を導入する。例えば, KDU19の<ブラウス@1>に従って、<ブラウス>フレームのインスタンス KDU23の@1— tupleをみることにより、隆が、1984年5月8日に"かどや"で買ったブラウスは、LLサイズで、万紡製の綿100%であったことを知る。

一般に、外延記述KDUによって、ある概念を、その属性とともにサンブルによって理解させることができる。

また、同じ値が、いくつもフレームの外延記述KDUに表われることによって、その値の意味を内包的に理解することができる。

2.3 知識の学習的検索と段階的提示

上述したような知識ベースを、学習支援に用いる方法は、さまざまに考えられるが、ここでは、学習者が、自ら、自分の知りたいことを、システムとの対話により、探索し、システムは、検索を学習的に支援するとともに、学習者の要求に応じて、段階的に、その知識を提示するといった、もっとも直接的な使い方を提案する。どんな上部構造をこの知識ベースの上にかぶせて学習支援システムを構成するにせよ、このような直接的なモードを利用できるようにしておくことは、システムの透明性―成人に対する学習支援においては、特に、システムに対する信頼を確保する意味において、重要である―という観点から、是非必要である。

また、学習者とシステムとの対話については、自然言語を用いることができれば、学習者に対してシステムに適応するための負担をかけずに済むのであるが、学習者が、自分の言いたいことを正確に表現してくれることを期待できないこと、また、音声理解システムを用いないかぎり、学習者が自然言語による文章で入力するのは、必ずしも能率的でないこと、ここで提案するような、知識ベースに直接アクセスさせるシステムにおいては、自然言語入力は、かえって、直接性を損なうこと、などの観点から、ここでは、キーワード入力、システムによるメニューの表示、それに基づく選択を用いることとし、知識の提示については、自然言語表現、図、音声を用いる。

知識の提示においても、KDUの表現形式に基づいて、自然言語表現、図表現、音声表現などを提示の必要が生じたときに生成することができれば、自然言語表現などをそのままストアする場合に比べて大幅な記憶容量の節約になると考えられるが、この点については、今後の課題としたい。

ただし、KDUのなかには、極くブリミティブなCFUを、テキスト、図などで表現するだけで、最早他のCFUとの関係をつけないものとか、いくつかのCFUを引用してはいるが、それらの間の漠然とした関連を、テキストや図で表現するだけのものがある。これらについては、上述のようなジェネリックな方法は、とりえない。

さて、提案するシステムにおいては、初め学習者に、自分の学習したい項目一概念あるいは関係一を、あるキーワードのセットを入力することによって、システムに伝えさせる。ところが、システムにとっての概念一概念そのものに興味がある場合にせよ、指定した概念に係わる関係を知りたい場合にせよ一は、全て、いずれかのCFUのいずれかのスロット以外のなにものでもない。そこで、システムには、学習者が入力する可能性のあるキーワードと、CFUのスロットへの対応表を持たせる。これを、用語リスト(Term List)と称する。このリストは、同義語などの処理を含むものであり、学習者がいくつかのキーワードを入力すると、それぞれのキーワードに対応する一般に複数のCFUとスロットを表示する。学習者はこのメニューのなかから自分の意図するところに一番近いものを指示する。

すると、システムは、そのCFUのスロットについて、どんな知識を得たいのかについてメニューを示す。学習者が、そのなかから選ぶと、システムは、対応したKDUへ跳んで、そこに置かれたインターフェースによって、その内容を自然言語表現、図、音声などを用いて、学習者に提示する。学習者は、この提示のなかのポイント(複数でもよい)にアクセスすることによって、次の段階の提示を求めるためのキーを与えたことになり、ここで再び、関係するCFUとスロットについての対話があって、指定するフォームが決まると、システム側から、得たい知識のカテゴリ(後述)についてのメニューが与えられる。

学習者が、そのなかから選ぶと、対応したKDUの提示が行われる。以下同様にして、学習者は、知識ベースのなかを、探索することができる。

この際,システムは、学習者の辿っている道筋を記録し、適宜、戻り道などのサジェスチョンを与えながら、学習者を誘導する。

特に、問題解決の支援を行う場合には、G. Polyaが提案[12]しているような、一般的な忠告を、適切な場面で与えることを考えている。学習者が、問題解決のどのような場面に居るかは、学習者が要求する知識のカテゴリーに加えて、状況に関する階層化されたメニューのなかで選ばせることにより判断することになろう。

一般に学習者は、必ずしも、自分が学習したい分野についてよく知らないので、常識的な概念にまずアクセスさせて、そこから、KDUを介して、次第に、より専門的な概念に、導くことができるように、さまざまなレベルのKDUならびにCFUを持った知識ベースを構成する。

さて、学習者が指示したキーに対して、システムが表示したCFUの候補に対し、CFUのいくつかのスロットに値を代入してもよく、そうしたいくつかの候補をしめしてもよいし、あるいは、いくつかのCFUについて、変数を用いてスロットの間の対応をつけたWFF形式で指定してもよい。そのなかのある変数を指定をして、その概念の説明や例示または他の概念との従属関係などを問うことができる。あるいは、こうして得られるWFFが真であるとしたときに得られる帰結とか、このWFFを真にするような前提を問うたりすることができる。

実際には、上述のようにして概念が指定されると、システムは、次のような メニューを表示し、学習者が、指定した概念についてどんな知識を得たいのか をたずねる。

- (1) それは,なに?(変数を指定した場合─上位概念による説明)
- (2) たとえば? (変数を指定した場合--下位概念を引いた例示)

(3) それが言えると何が言えるか?

(WFFを指定した場合―与えたものを前提としたときの帰結の例示)

(4) 何が言えればそれが言えるのか?

(WFFを指定した場合---与えたものを帰結とするような前提の例示)

(5) それは,何が決まれば,決まるか?

(変数を指定した場合―指定された概念の値を決定するに十分な値を与えるような概念)

(6) それが決まれば,何が決まるか?

(変数を指定した場合―指定された概念の値からその値を決定できる他の概念)

(7) それを解法に用いた証明問題は?

(ルール,定理の名前を指定したとき)

(8) それを解法に用いた決定問題は?

(ルール,定理の名前を指定したとき)

(1)の場合,指定された概念を直接説明するKDUが存在しないときには,当該CFUに関する何等かの説明をしているKDUを引用する。

(2)の場合,すぐ下の下位概念(一般には、複数ありうる)を示すが、要求があれば、いくらでも下位の概念を引用できるようにする。

付図 1.に,誤り制御符号通信に関連した知識ベースの例を示した(これだけでは、十分なものでないことをお断わりしておく)。

各KDUには、そのKDUで述べたい内容を自然言語のテキストおよび、必要に応じて図を用いて表現したものが蓄えられており、それらのテキストあるいは図においては、背後のKDUにおけるCFUのスロットに対応した表現の部分に当該KDUにおけるCFUの番号とスロットに置かれた変数とが記されていて(ディスプレイ上には、表示されない)、学習者がテキストや図のそのような部分付近をポイントしたとき、システムにおいて、直ちに該当するCFUとスロットを対応づけられるようにしてある。

付図 2.は、正四角錐台の体積に関する公式を求める決定問題の解法を示すの

に必要な知識(テキストは省略されている),ならびに,解法の表現と,解法 を図によって段階的に表示する表示法の試案を示したものである。

付図 3.は、ある平面上の二直線に垂直な直線は、同平面上の任意の他の直線に垂直であることを証明する手順の表現と、図を用いて証明手順を段階的に示す提示法の試案を示している。

証明や解法の手順の提示に於けるテキストは、用いるKDUに付されたテキストの変数相当部分に当該問題におけるインスタンスを置いて表示することが考えられる。

参考文献

- (1) S. Otsuki and A. Takeuchi: 'A unified CAL system for authoring, learning and managing aids, 'Proc. I FIP TC-3, pp. 249-256, 1981
- (2) M. Minsky: 'A framework for representing knowledge.' in P. Winston(ed.) The psychology of computer vision, Mc-Graw-Hill, 1975
- (3) J. Piaget and B. Inhelder: La psychologie de 1 enfant. Collection QUE SAIS-JE? No 369,9th ed. Presse Universitaires de France, 1980
- (4) D. G. Bobrow and T. Winograd: 'An overview of KRL, a knowledge representation language, 'Cognitive Science, vol. 5, no. 4, Oct-Dec. 1981
- (5) D. G. Bobrow and D. A. Norman: Some principles of memory schemata. in D. G. Bobrow and A. Collins(ed.) Representation and understanding, Academic Press, 1975
- (6) N. J. Nilsson: Principles of Artificial Intelligence, Springer, 1982
- (7) R_o C_o Schank: Conceptual information processing, North-Holland, 1975
- [8] J. H. Larkin: 'A General knowledge structure for learning or teaching science, ' "in A. C. Wilkinson (ed.), Classroom computers and cognitive science.

 Academic Press, 1983
- [9] C. J. Date: An introduction to database systems, vol. 1,2 3rd ed., Addison Wesley, 1982

- [10] 伊藤, "学習支援システムの為の知識の構造化について" 信学技報 E T 84-5,1984
- [11] 堀,斉藤,猪瀬, "帰納的推論の記述に有用な知識表現の一提案" 情報 処理学会論文誌 24, 1, pp 72 - 79, 1983
- [12] G. ポリア, "数学の問題の発見的解き方"上,下,柴垣,金山訳,みすず書房,1964
- 〔13〕 伊藤, 浅野, 堀, 横尾"学習支援システムにおける知識の検索および提示の為の知識表現について" 昭和60年度信学総全大S18-4,1985年3月

附図1. 知識ベース-誤り制御符号

CPU 数ベクトル空間self#(V),数ベクトル*V,基礎体#、次元#

CFU 数ベクトルが直交する self* ベクトル (*、*) 基礎数ベクトル空間 米

CFU 数ベクトル部分空間と表現 self#(T,E)
 部分空間のベクトル米T ベクトルの表現米E 部分空間次元# 基底ベクトルの行列#

直交補空間#:数ペクトル部分空間(次元[直交補空間次元]#) 基礎数ペクトル空間#:数ペクトル空間(次元[基礎空間次元]#,基礎体#)

CFU 多項式剰余環 self#(P) 剰余類*P 法多項式# 係数体#

CFU 多項式イデアル
 self#(A)
 イデアル要素*A
 生成多項式#
 基礎多項式剰余環#:多項式剰余環(係数体#,法多項式#)

CFU原始多項式 self* 次数* 係数体*

CFU 情報の変換self#(A>>B),変換前情報*A,変換後情報*B,変換方式#,変換手段#,

変換速度#

CFU 情報の伝達

self井(A>>B)、情報源からのメッセージ*A、 情報届け先へのメッセージ*B、情報源井、情報届け先井、伝達媒体井

CFU 符号化通信

self#(M1>>C1>>C2>>M2)

:情報の伝達(情報源からのメッセージ*M1,情報源#,情報届け先# 情報届け先へのメッセージ*M2)

: ディジタル伝送(伝送路 [符号化通信路] 井) ・

送信符号系列*C1

受信系列米 C 2

符号#

符号化井

:符号化(符号器井)

:復号(復号器弁)

内部ディジタル伝送井:ディジタル伝送(伝送路井、伝送路誤り率井)

CFU 情報源符号化通信 :

self#(M1>>C1>>C2>>M2):符号化通信,:ディジタル伝送

:情報の伝達(情報源からのメッセージ*M1,

情報届け先へのメッセージ×M2、情報源井、情報の届け先井)

送信二值符号系列 * C 1

受信二值符号系列 * C 2

情報源符号井

情報源符号化#

: 符号化(符号器 [情報源符号器] #)

情報源復号井・復号は復号器[情報源復号器]井)

内部ディジタル伝送井:ディジタル伝送(伝送誤り率井)

許容平均歪井 情報伝送速度#

CFU 通信路符号化通信

self#(M1>>C1>>C2>>M2):符号化通信,:ディジタル伝送

送信二値情報*M1

受信二値情報*M2

送信符号系列米 C 1

: シンボル列(アルファベットサイズ [送信アルファベット

サイズ] #)

受信系列 * C 2

: シンボル列(アルファベットサイズ [受信アルファベット

サイズ] 井)

通信路符号#

通信路符号化# : 符号化 (符号器#) 通信路復号# : 復号 (復号器#)

内部ディジタル伝送井 :ディジタル伝送(伝送路井,伝送路誤り率井)

符号化通信路井 符号化誤り率井 情報伝送速度#

CFU 2元プロック符号通信

self#(M1>>C1>>C2>>M2):通信路符号化通信

送信情報プロック * M1

受信情報プロック米M2

送信符号語 * C 1

送信符号語要素米

受信語 * C 2

2元プロック符号#

符号長井

情報ビット長井

CFU 2元線形プロック符号通信

self#(M1>>C1>>C2>>M2)

:2元プロック符号通信#(送信情報プロック*M1、送信符号語*C1、 受信語*C2、受信情報プロック*M2、符号長井、情報ビット長井)

2元線形プロック符号#

符号化#

:符号化

生成行列#

パリティチェック行列#

復号# :復号

パリティビット長井

符号問最小距離井

CFU 2元巡回符号通信

- self#(M1>>C1, P1>>C2>>M2)
 - :2元プロック符号通信(送信情報プロック*M1,送信符号語*C1, 送信符号語要素米、受信語米C2、受信情報プロック米M2、符号長井、 情報ビット長井)
 - : 2元線形プロック符号通信(バリティビット長井、符号間最小距離井、 生成行列井、パリティチェック行列井)

符号多項式 * P 1 : G F (2) 上の多項式

生成多項式#: GF(2)上の多項式

符号化并:符号化(符号器件)。

誤り検出復号#:復号(復号器#)

CFU 2元BCH符号通信

self#

: 2元巡回符号通信(生成多項式#:GF(2)上の多項式(根#))

: 2元線形プロック符号通信 (パリティビット長井,符号間最小距離井) …

: 2元プロック符号通信(符号長井、情報ビット長井)

mパラメータ#

tパラメータ井

CFU ハミング符号通信

self#:2元線形プロック符号通信・

: 2 元巡回符号通信

: 2元BCH符号通信

KDU 部分空間とその直交補空間

- O. 数ペクトル部分空間と表現(self#W. 部分空間のベクトル*V. 表現*U. 部分空間の次元#K, 基底ベクトル行列#G, 基礎数ペクトル空間#VV. 基礎空間の次元#N. 直交補空間#WT)
- 1.数ベクトル部分空間と表現(self#WT,部分空間のベクトル米VT、 部分空間の次元#M、基底ベクトル行列#H、基礎数ベクトル空間#VV, 直交補空間#W)
- 2. 数ベクトルが直交する(ベクトル {*V,*VT},基礎数ベクトル空間*VV)
- 3. V = UG
- 4. VH' = 0
- 5. N = K + M

aW, 0 = > 1.2

数ベクトル空間における部分空間の直交補空間は、また部分空間であり、部分空間のベクトルと直交補空間のベクトルは、直交する.

aW, 0 = > 1, 3, 4, 5

数ベクトル空間(次元N)における部分空間(次元K)の基底ベクトル行列をGとし、部分空間のベクトルVの、Gによる表現をUとするとき、V=UGが成り立つ、また、直交補空間(次元M)の基底ベクトル行列をHとするとき、VH'=0が成り立つ、なお、

 $N = K + M \tau \delta \delta$.

KDU 上多項式イデアル

- 0. 多項式イデアル (self # ID, イデアル要素*A, 生成多項式 # G, 基礎多項式剰余環 # R C, 法多項式 # M)
- 1. 集合(self # ID, 要素*A)
- 2. 多項式が割り切れる(割られる式*M、割る式*G)
- 3. 多項式剩余環(self#RC,剩余類*A,法多項式##M)
- 4. 多項式剰余類が割り切れる(割られる剰余類*A. 割る剰余類*G. 多項式剰余類井RC))
- a I D, aA, $0 \le > 1.2.3.4$

多項式M(x) を法とする多項式剩余環をRCとする、M(x) を割り切るある多項式 G(x) とするとき、RCの剰余類で、G(x) で割り切れるものを要素とする集まりのことを、M(x) を法とし、G(x) を生成多項式とするイデアルという。

KDU 符号化通信1

- 0. 符号化通信(self#T. 符号#CD. 符号化#E. 内部ディジタル伝送#DT, 復号#D,送信符号系列*CT,受信系列*CR)
 - 1. 情報伝達(self #T, 情報源からのメッセージ*MT, 情報届け先への メッセージ*MR, 情報源#S, 情報届け先#A)
 - 2. 符号化(self#E,メッセージ*MT,符号系列*CT、符号#CD)
 - 3. ディジタル伝送(self*#DT,送信シンボル列*CT, 受信シンボル列*CR)
 - 4. 復号(self#D、符号系列*CR、メッセージ*MR、符号#CD)

aT, 0 = > 1, 2, 3, 4

符号化通信は、情報源からのメッセージを、ある符号によって符号化して送信符号系列とし、それを、ディジタル伝送し、受かった受信系列から件の符号に基づいて復号することによって、メッセージを得、これを、情報届け先へ届けるというものである。

KDU 符号化通信2

O. 符号化通信(self#T,符号器#CR,復号器#DR,内部ディジタル 伝送路#DC,符号化通信路#CC)

1. 構成する(構成結果*CC, 構成要素(*CR, *DC, *DR))

a T, 0 = > 1

符号化通信の符号化通信路は、符号器、内部ディジタル伝送路、復号器とから構成される。

KDU 情報源符号化通信

0.情報源符号化通信(self#SC,情報源からのメッセージ*MT, 情報届け先へのメッセージ*MR,平均歪の許容値#DP,情報伝送速度#R)

情報源符号化通信は、平均歪、すなわち、情報源からのメッセージと情報届け先へのメッセージとの相対差異の平均を、許容値より小さくしながら、情報伝送速度を下げることを、目標とする。

KDU 通信路符号化通信1

- O.通信路符号化通信(self #CC,送信二値情報*DT、 受信二値情報*DR、送信符号系列*CT、受信系列*CR、 通信路符号化#CE,通信路復号#CD, 伝送路誤り率#ER, 符号化通信路#CN)
- 1. 符号化通信(self#CC、情報源からのメッセージ*DT, 情報届け先へのメッセージ*DR、送信符号系列*CT, 受信系列*CR、符号化#CE,復号#CD,情報源#S,情報届け先#D, 伝送路誤り率#ER,符号化通信路#CN)
- 2. 情報源符号化通信(送信二値符号系列*DT,受信二値系列*DR, 情報源符号器#S,情報源復号器#D,伝送路#CN)

a C C, 0 = > 1. 2

通信路符号化通信は、符号化通信であって、その情報源は、情報源符号器であり、その情報届け先は、情報源復号器である。この場合の情報源からのメッセージ、および届け先へのメッセージは、それぞれ、情報源符号化通信の送信二値符号系列、および受信二値符号系列である。

KDU 通信路符号化通信2

- 0.通信路符号化通信(self#CC,送信二値情報*DT,受信二値情報*DR, 送信符号系列*CT,受信符号系列*CR,伝送路誤り率#ER,,符号化ビット 誤り率#PB,情報伝送速度#R)
 - 1. ~極く小さい(対象*ER)
 - 2. 冗長性あり(対象 {*CT, *CR}, 冗長性*R)

a C C, $0 = > 1 \cdot 2$

通信路符号化通信は、伝送路誤り率が、極く小さいとは言えないときに用いられ、符号 系列の冗長性を用いて、一定の情報伝送速度を確保しつつ、送信二値情報と受信二値情報 の相対差異の平均,即ち符号化ビット誤り率を極く小さくすることができる.

KDU 2元線形プロック符号通信

- 0.2元線形プロック符号通信(se!f#LB, 生成行列#G, パリティチェック 行列#H, パリティビット長#M)
- 1. 2元プロック符号通信(self#LB,送信情報プロック*U,送信符号語 *V,情報ビット長#K,符号長#N)
- 2. 数ペクトル部分空間と表現(部分空間ペクトル*V,表現*U、部分空間#W,基底ペクトルの行列#G,部分空間の次元#K,直交補空間#C,基礎空間の次元#N,基礎体#KK:GF(2))
- 3. 数ベクトル部分空間と表現(部分空間#C、基底ベクトルの行列#H、 直交補空間#W)
- $4 \cdot V = UG$
- 5. VH' = 0
- 6. N = K + M
- -1 2元線形プロック符号は、符号語がある部分ベクトル空間をなすプロック符号。 a L B 、0 <=> 1 、 2

2元線形符号化通信は、2元プロック符号化通信で、送信符号語が、あるベクトル空間のベクトルであり、その基底ベクトルによる表現係数が、送信情報プロックであるようなものである。

-2 2元線形プロック符号の符号化とパリティチェック aLB, 0.1=>4.5.6.7

2元線形プロック符号の生成行列をGとし、送信情報プロック(長さK)をUとすれば、送信符号語 V(長さN)は、V=U G で与えられる。また、パリティチェック行列をHとすれば、V H'=0である。なお、Hの階数をMとすれば、これはパリティビットちょうであって、N=M+K が成り立つ。

- KDU 2元巡回符号の符号器は、バイナリシフトレジスタで実現できる
 - 0. 二元巡回符号通信(self#S,符号器#EC)
 - 1. 実現することができる(対象*EC,手段*M)
 - 2. バイナリシフトレジスタ回路(self*M)

aS. 0 = > 1.2

2元巡回符号の符号器は、バイナリシフトレジスタによって実現できる。

- KDU 2元巡回符号通信は、イデアルを符号として用いる
 - 0. 2元巡回符号通信(self#S,符号多項式*A(x), 生成多項式#G(x))
 - 1. 2元線形プロック符号通信 (self#S,符号語要素*{Ai}, 符号長#N,パリティビット長#M)
 - 2. GF(2)上多項式イデアル(イデアル要素**A(x), イデアル要素の係数*{Ai},生成多項式井G(x), 生成多項式の次数井M、多項式剰余環の法井M(x))
 - 3. M(x) = X 1

 $aS, 0 \le > 1.2.3$

2元巡回符号は、2元線形プロック符号の一種であって、M(X) = X - 1を法とする多項式剰余環のイデアルの要素を符号語として用いるもの、イデアルの生成多項式をG(X)とし、G(X)が割り切る最小の次数のX-1という多項式の次数がNであるならば、G(X)の次数はパリティビット長に等しく、符号長は、Nに等しい。

KDU BCH符号は、原始元からつくられる巡回符号

- O. 2元BCH符号通信(self#S,mパラメータ#m,tパラメータ#t)
- 1.2元巡回符号通信(self#S,生成多項式の根#{R})
- 2. GF (2) 拡大体 (self #EF, 次数#m, 原始元 # {α})
- 3. R = α 1 \leq i \leq 2 t

aS, 0 < = > 1, 2, 3

 $2 \pi B C H$ 符号は巡回符号の一種であって、GF(2)のm次の拡大体の原始元を α とするとき、 α $1 \le i \le 2$ tを根とする最小次数の多項式を生成多項式とするものである。ここに、m、t は、B C H 符号を決める 2 つのパラメータである。

KDU BCH符号の符号長、符号間最小距離、パリティビット数

- 0. 2元BCH符号通信(self#S, mバラメータ#m, tパラメータ#t, 符号長#n, バリティビット長#p, 符号問最小距離#dm)
- 1. n = 2 1
- 2. $dm \ge 2t + 1$
- 3. p≦mt

aS, 0 = > 1.2.3

2元BCH符号通信のmパラメータ、tパラメータを用いて、符号長は2 -1であり、符号間最小距離は、2t+1以上、パリティビット長は、mt以下であることが保証され

る.

KDU ハミング符号通信 1

- ハミング符号通信(self#S)
- 1. 2元線形プロック符号通信(self#S,パリティチェック行列#H,符号長#n.パリティビット長#m.最小符号間距離#dm)
- 2. 行列 (self#H,行数#m,列数#n,列ベクトル*hi)
- 3. {hi=BE, BEは0でない}
- 4. ビットパターン (self*BE, ビット数#m)
- 5. n = 2 1
- 6. dm = 3
- -1 ハミング符号のパリティチェック行列

aS, $0 \le > 1$. 2. 3. 4

ハミング符号は、2元線形プロック符号であって、そのパリテイビット数をmとするとき、パリティチェック行列の列ベクトルとして、ビット数mの零でないすべてのビットパターンを置いたものである。

-2 ハミング符号の符号長とパリティチェック行列長 aS, 0 = > 1. 5

Nミング符号の符号長nは、2 -1である。これは、パリティチェック行列の列におかれるゼロでないmビットパターンの個数として得られる。

-3 ハミング符号の最小符号問距離 a S , 0 = > 1 , 6

ハミング符号の最小符号間距離は, 3である.

proof

KDU ハミング符号のパリティチェック行列 KDU 2元線形プロック符号の最小符号問距離

。 KDU ハミング符号通信2

- ハミング符号通信(self #S)
- 1. 2元巡回符号方式(s'elf#S, 生成多項式#G, 次数#m, 符号長#n)
- 2. 原始多項式 (self #G, 次数#m)
- 3. n = 2 1
- -1 ハミング符号は、巡回符号

a S, 0 < = > 1.2

ハミング符号は、2元巡回符号であって、その生成多項式が、原始多項式の場合である。

proof

KDU ハミング符号のパリティチェック行列

KDU 2元巡回符号のパリテイチェック行列

KDU 2元巡回符号は、2元線形プロック符号

KDU パリティチェック行列から、2元線形プロック符号は一意に決まる。

-2 ハミング符号の符号長と生成多項式の次数

aS, 0 = > 3

Nミング符号の符号長nは、n=2 -1となる。ここに、mは、生成多項式の次数である。

proof

KDU ハミング符号通信は、巡回符号

KDU 2元巡回符号の符号長

KDU 原始多項式の根の位数

KDU ハミング符号通信は、BCH符号

ハミング符号通信(self#S)

1. BCH符号通信(self#S, tパラメータ#t:1)

a S. 0 < = > 1

ハミング符号は、BCH符号であって、そのセパラメタが、1の場合である。

proof

KDU 2元BCH符号は、原始元から作られる巡回符号

KDU 既約多項式の共役根

KDU 2元巡回符号の生成多項式

KDU ハミング符号は,巡回符号

附図2. 角錐台の求積の為の知識ベース

CFU 部分と全体 self * * P 全体 部分 *P1 残りの部分 *P2





CFU 2図形が相似の位置にある

self *

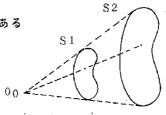
図形1

* * S.

図形2

** S₂

相似の中心 *0



CFU 2図形が交わる

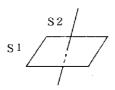
self *

図形 1

** S,

図形2 ** 52

交わり、 🔭





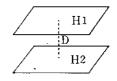
CFU 二面が平行

self *

平行面 1 ** H 1

平行面 2 **H2

二面間距離 *D



で CFU 線分

self *

端点 **(A,B)

長さ *

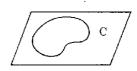


CFU 平面分

self *

周辺 * * C

面積 *



CFU 立体 self * 体積 *

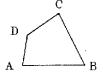


CFU 四辺形

self *平面分(面積*)

* (AB/BC/CD/DA)

頂点 * (A/B/C/D/)



CFU 長方形

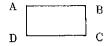
self· *平面分(面積*)

対辺 1

(AB/DC):線分(長さ→対辺1の長さ※)

対辺2

(AD/BC) :線分(長さ→対辺2の長さ*)



CFU 正方形

self *:平面分(面積#):長方形()

* (AB/BC/CD/DA):線分(長さ→辺の長さ*) 辺

頂点 * (A/B/C/D)



CFU 錐体

self *:立体(体積*)

頂点 *****0

B:平面分(面積→底面の面積) 底面

S:曲面(面積→側面の面積) 側面

高さ * H



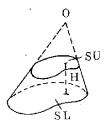
CFU 錐台

self *:立体(体積*)

上底 *SU:平面分(面積→上底面積*)

***SL:平面分(面積→下底面積*)** 下底

仮想頂点 *0 高さ * H 大錐体 *0 S L 切除錐体 半〇SU



KDU 正方形

- 正方形(self*S,辺の長さ*L,面積*A)
- 1. 長方形 (self*S,対辺1の長さ*L1,対辺2の長さ*L2)
- 2. L1 = L2 = L
- $3. \quad A = L^2$

 $[0 | 1 \land 2]$ [0 | [3]]



KDU 錐体

- O. 錐体(self*C,底面積*A,高さ*H,体積*V)
- 1. 立体(self*C)
- 2. $V = A \times H / 3$

 $[0 [1_{\wedge} 2]]$



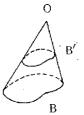
KDU 錐体の決定

- 0. 錐体(self*C,底面*B,頂点*O)
- 1. 平面体(self*B)
- 2. · 点(self*0)

 $[0 | 1_{\wedge} 2]$

KDU 錐体を切る

- 0. 錐体(self*C,頂点*0,底面*B)
- 1. 2図形交わる((図形1**C,図形2**B)交わり*B))
- 2. 二面平行((面**B,面**B))
- 3. 2図形相似の位置((図形1*B,図形2*B),相似の中心O)) [0_1_2 [3]]

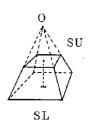


CFU 正四角錐台

self *:錐台(仮想頂点*O,高さ*H)

上底 *SU 上底:正方形(辺*:線分(長さ→上底辺の長さ))

下底 *SL 下底:正方形(辺*:線分(長さ→下底辺の長さ))



KDU 部分と長さ

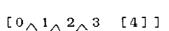
- 0. 部分と全体(全体*S,部分*S1,残りの部分*S2)
- 1. 線分(self*S),
- 2. 線分(self*S1)
- 3. 線分(self*S2)

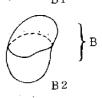


$$[0 \land 1 \land 2 \land 3 \quad [4]]$$

KDU 部分と体積

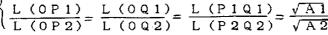
- 0. 部分と全体(全体*B,部分*B1,残りの部分*B2):
- 1. 立体(self*B)
- 2. 立体(self*B1)
- 3. 立体(self*B2)
- 4. { 立体 (self*X,体積*V(X)) V(B)=V(B1)+V(B2)



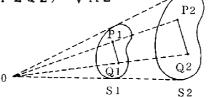


KDU 平面分が相似の位置にある

- 0. 2 図形が相似の位置((図形1**S1,図形2**S2),相似の中心*0))
- 1. 平面分(self*S1[点{*P1,*Q1}],面積*A1)
- 2. 平面分(self*S2[点{*P2,*Q2}],面積*A2)
- 3. (線分(端点**(X1, X2), 長さ*L(X1, X2))



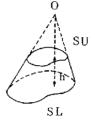
[0, 1, 2 [3]]



KDU 錐台

- 0. 錐台(self*T,大錐体*C1,切除錐体*C2,上底*SU, 上底面積*AU,下底*SL,下底面積*AL,仮想頂点*O, 高さ*h,体積*V)
- 1. 立体(self*T)
- 2. 錐体(self*C1,頂点*O,底面*SL)
- 3. 錐体(self*C2,頂点*O,底面*SU)
- 4. 二面平行(面1*SU,面2*SL,距離*h)
- 5. 2 図形相似の位置(図形1*SU,図形2*SL,相似の中心*0)
- 6. 部分と全体(全体*C1,部分*C2,残りの部分*T)
- 7. $V = h/3 \times (AL + \sqrt{AU \times AL} + AU)$

 $[0 \quad [1_{\wedge} \quad 2_{\wedge} \quad 3_{\wedge} \quad 4_{\wedge} \quad 5_{\wedge} \quad 6_{\wedge} \quad 7]]$



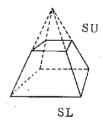
 $C_1: OSL$

 $C_2:OSU$

KDU 正四角錐台

- 0. 正四角錐台(self*ST)
- 1. 錐台(self*ST,上底*SU,下底*SL)
- 正方形(self*SU)
- 3. 正方形 (self * SL)

 $\begin{bmatrix} 0 & \begin{bmatrix} 1 & 2 & 3 \end{bmatrix} \end{bmatrix}$



```
| a
         KDU正方形の面積
         A_1
                    h
                  KDU部分と長さ
                  f, (x, y)
                    (a, b)
                  KDU相似の位置
                  f_2(x, y)
            [KDU連立方程式の解法]
            (x, y)
         x \leftarrow (x, y)
     KDU錐体の体積。
     V<sub>1</sub>
         KDU正方形の面積
         y \leftarrow (x, y)
     KDU錐体の体積
    1 V2
KDU部分と体積
```

- 決定問題解法 KDUの例

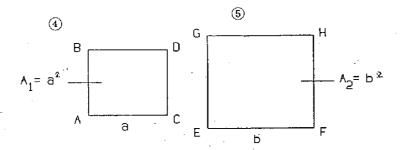
-131 -

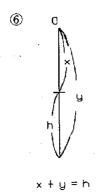
1

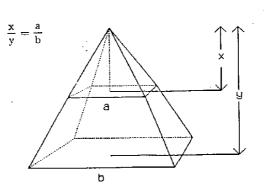
٧2

E

決定問題解法提示例(i)







決定問題解法提示例(II)



附図 3. 証明手順の提示の為の知識ベース

∠P0C=∠P'0C

∠POP'=2∠R

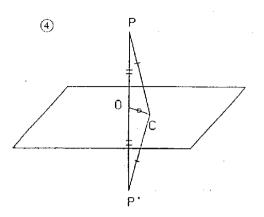
結論

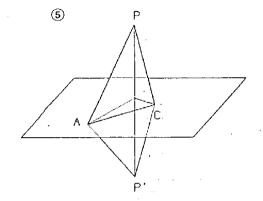
前提。

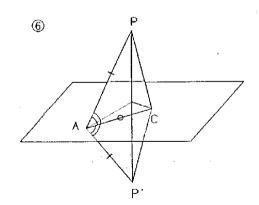
- 134 **-**

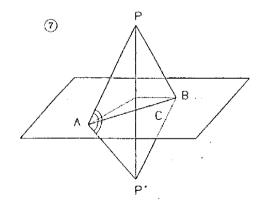
1

証明問題解法提示例(a)

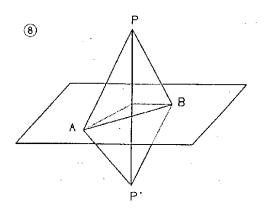


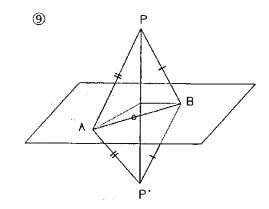


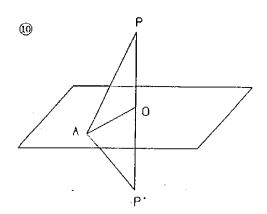


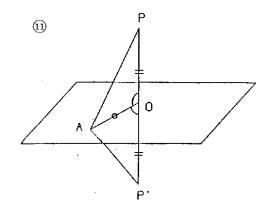


証明問題解法提示例(b)

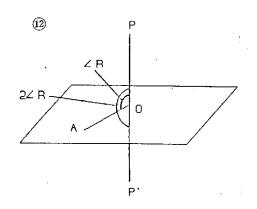


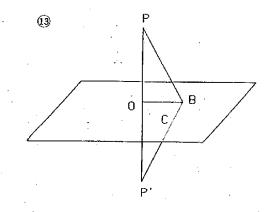


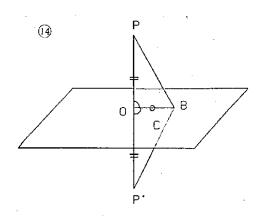


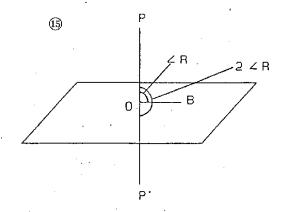


証明問題解法提示例(c)









証明問題解法提示例(d)

第3章 知的情報処理システムの問題理解システムへの応用

3.1 はじめに

「知的情報処理システム」とは、私たちにとって何を意味するのであろうか? 私たち人間の知的機能、たとえば言語処理や推論機能を代替して、代りに知的情報処理をやってくれるシステムを意味するのであろうか?

米国では10数年前から研究が始められ、わが国ではつい数年前から流行語のきざしを見せ始めた「知識工学」が知的情報処理システム研究の典型だとすれば、それは確かに、私たちの知的機能の一部を代替してくれるシステムだと言えるだろう。

しかしながら、人間の知的機能には、現在の人工知能研究ではとてもカバーできていない、きわめて多くの機能がある。たとえば、確かにコンピュータは三段論法推論を行なうことができる。しかし、コンピュータは自分で三段論法の定式化を行なうことができるであろうか。つまり、多くの混沌とした情報の中から、大前提、小前提、結論の3つ組を構成することができるであろうか。

一般的なかたちでは、今の研究段階では困難だろう。混沌とした情報を問題として理解するのは、コンピュータには難しい。現在の人工知能システムにおいて欠けている機能のひとつは、推論ではなく、問題の理解機能なのである。

しかしながら、一方では、私たちにとって問題を理解する能力はきわめて大切な能力である。そうだとすれば、この問題理解の能力を育てることは、もし可能ならば大変重要なことであり、それが人工知能システムの助けを借りてできうるならば、人工知能技術の応用として、今後考えるべき大きな問題だと思われる。

そこで、この稿では、人工知能の応用としての、「問題理解のための知的教育工学システム」(Intelligent Computer-Assisted Problem Understanding System)について、筆者のグループで行なわれている2つの研究 — 算数の応用問題の解決システムと初等物理学の問題解決システム — について紹介することにしたい。

はじめにお断りしておくが、人工知能、知的情報処理、知識工学一般につい

ての紹介は、他にいくつかサーベイがあるので、ここでは割愛させて頂くことにしたい。本稿の主旨は繰り返しのべておくとすれば、知的なコンピュータシステムの応用として、人間の知的機能を代替する、つまりある意味では人間が自己の能力を発揮しなくてもよいようにしてくれることを主目的とするもののほかに、人間の問題理解のための知的機能をより高めるための補助的役割としての応用があるということを、算数と物理の2つの例を引き合いにして考えてみることにある。

3.2 算数の文章題の問題解決システム

3.2.1 何のためのシステムか

まず、算数の文章題を解くシステムの研究について簡単に紹介しよう。

この研究の主目的は、小学校低学年の児童が「自分で作った」算数の文章 題をコンピュータに日本語で入力すると、その問題を解いてくれるようなシステムを試作することであった。

これまでの多くのICAI (Intelligent Computer-Assisted Instruction) の研究では、問題をコンピュータが選び、学習者に解かせることを繰り返すというタイプのシステムを作ることを目的とするものが多かった。たとえば最近では、学習者の解き方を解析して学習者のモデル(student model)を構成し、そのモデルに合わせて問題の出し方を変えてゆくというシステムも考えられている。

しかし、このような方法では、いつまでたっても、学習者側は、システムに「教えてもらう」、「単に」解法の手順を学ぶ、という気分から抜け出ることが難しい。

しかし、特に小学校低学年の算数の文章題では児童にとって難しいのは計算ではなく問題の文章を理解する過程だというのは、現場の教師の間ではよく知られたところである。そこで、問題の理解能力を育てるために、これまでのICAIの発想を逆転して、学習者に問題を作らせ、システムに解かせることを考えたわけである。問題を作るという方法は小学校でもある程度行なわれており、定量的な評価は難しいが、問題の構造の理解を助けるといわ

れている。図1にこれまでのシステムとここで述べるシステムの違いを図示 しておこう。

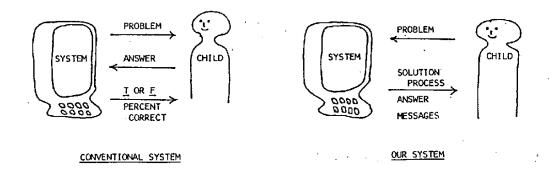


図1 これまでのICAIシステムとわれわれのシステム

3.2.2 システムの概要

まず,小学校低学年生が使えるような算数の文章題を解くシステムには, 次のような機能が必要である。

- (1) 日本語文章をひらがな主体の大きな文字で入力できるようなインタフェース。
- (2) 入力された文章の意味解析機構。
- (3) 解析された意味をさらに問題解決のための表現に変換する機能。
- (4) (3)の表現を数式表現に変換する機能。
- (5) 数式処理の機能。
- (6) (1)~(5)およびエラーメッセージ,なぜそう解くのかの説明などを,やさ しい日本語と式で表示するインタフェース。

ここに述べるシステムでは、(2)~(5)をVAX11/750 上の FRAN2 LISP で書き、(1)と(6)をSONY TEKTRONIX 4052 のグラフィックディスプレイで行なっている。グラフィック表示はTEKTRONIX内蔵のBASICで書き、LISPとBASICの間でデータ転送を行なっている。ただし、現時点では、(6)はまだ簡単な日本語と式の表示しかできてい

ない。(1)は、TEKTRONIXのキーボードのキーひとつでひらがな1文字を入力できるようにしてある。つまり、学習者から見れば、ひらがなキーボード化されている。

3.2.3 問題文章の意味解析

システムには、たとえば次のような文章が入力される。

「りんごが4こありました。いま9こあります。たろうはりんごをなんこいれたでしょうか。」 …… (1) この文章を意味表現に変換するのは、いわゆる自然言語処理の問題だが、本ンステムでは、エール大学のSchankらが開発した概念依存文法(Conceptural Dependency Grammar)を応用して意味表現への変換を行なっている。

上記の文章(1)を意味表現に変換した結果を図 2(a)に示す。図 2 (a)はLISP の構文構造をしており、I Sは存在、ACTORは主体、LOCは場所を表わす。また、PTRANSは物理的対象の移動を表現しており、(ANY (ID (1))、(ANY (ID (2))は、文章には直接表現されていない場所を示す。HOWMANYVARはもとの文章中での未知数を表わす。

別の例として、次の問題から得た意味表現を図2(b)にあげておくことにする。

「たろうはピーナッツを6こもっていました。はなこが2こたろうにあげました。たろうはピーナッツをなんこもっていますか。」 (2) 図 2 (b)では、POSSは所有を、ATRANSは、抽象的な対象 (ここで所有権) の移動を表わしている。

(a)

```
(IS (ACTOR (APPLE (NUMBER (4 (UNIT (KO)))))
  (LOC (ANY (ID (1))))

(IS (ACTOR (APPLE (NUMBER (9 (UNIT (KO))))))
  (LOC (ANY (ID (1)))))

(PTRANS (ACTOR (TÄROW))
  (OBJECT (APPLE (NUMBER (HOWMANYVAR (UNIT (KO))))))
  (FROM (ANY (ID (2))))
  (TO (ANY (ID (1)))))
```

(b)

```
(POSS (ACTOR (TAROW))
(OBJECT (PEANUT (NUMBER (6 (UNIT (KO))))))

(ATRANS (ACTOR (HANAKO))
(OBJECT (PEANUT (NUMBER (2 (UNIT (KO))))))
(FROM (HANAKO))
(TO (TAROW)))

(POSS (ACTOR (TAROW))
(OBJECT (PEANUT (NUMBER (HOWMANYVAR (UNIT (KO)))))))
```

図2 問題文章の意味表現

- (a) 問題(1)
- (b) 問題(2)

3.2.4 意味表現の問題表現への変換

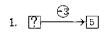
図2のような意味表現そのままでは、問題を解くことはできない。図のような意味表現を問題解決に適した表現に変換する必要がある。

 態の遷移(transfer of states), 状態の比較(comparison of states), 遷移の比較(comparison of transfer), 状態の結合 (unifying of states), 遷移の結合 (unifying of transfer) に分類できる。13種類それぞれの例を図3に示す。

問題表現

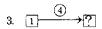
例

A. 状態の遷移



たろうはなしを3こたべた。 かごには5このこった。 はじめなしはいくつあったか。

きょうかごにはオレンジが10こあった。 きょうは6こある。 いくつたべたか。



かどにはりんごが1とだけある。 たろうは4とかった。 かごにはいまりんごはいくつあるか。

B. 状態の比較



オレンジのほうがりんごより 3 こすくない。 オレンジは 9 こある。 りんごはなんこあるか。



オレンジが12とある。 りんごが8とある。 りんごはオレンジよりなんとすくないか。



りんごが7 こある。 オレンジはりんごより5 こおおい。 オレンジはなんこあるか。

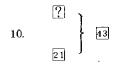
、C. 遷移の比較

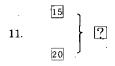




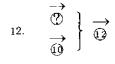


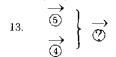
D. 状態の結合





E. 遷移の結合





たろうはオレンジをはなこより6こおおくたべた。 たろうは8こたべた。 はなこはなんこたべたか。

しろぐみはたまいれでたまを10といれた。 あかぐみは15といれた。 あかぐみはしろぐみよりなんこおおくいれたか。

しろぐみはたまいれでたまを13といれた。 あかぐみはしろぐみより4こすくなかった。 あかぐみはなんといれたか。

1くみにはおとこのことおんなのこが43にんいる。 おんなのこは21にんいる。 おとこのこはなんにんいるか。

ジャイアンツのいちぐんには15にんのせんしゅがいる。 にぐんには20にんいる。 ジャイアンツにはぜんぶでなんにんのせんしゅがいる か。

たろうのチームはこうはんに10てんとった。 チームは12てんとってかった。 チームはぜんはんになんてんとったか。

たろうはねこを5ひきもらった。 じろうは4ひきもらった。 ぜんぶでなんびきもらったか。

図3 13種類の問題表現

システムは、図2のような意味表現のパタンから、問題の構造が図3の13 種類のどれにあたるかを判断するようになっている。たとえば、

「たろうはなしを3こたべた。

かごには5このこった。

はじめなしはいくつあったか。」

という問題では、意味表現のパタンが、IS-PTRANS-ISのかたちで、はじめのISの中にHOWMANYVARが含まれる(正確にいうと、PTR-ANSではなくまず INGESTという意味素が使われ、システムはそのレベルの意味表現を生成してから、それをPTRANSを用いたより単純な意味表現に変換する。)。このようなパタンの場合は、図3の最初の問題表現、つまり $\Box \to \Box$ のかたちの状態遷移表現が生成される。

3.2.5 数式への変換と答の導出

図3のような問題表現から式を導くのは簡単だし、計算して答を導くのも 単純な方法ですむので、ここでは省略する。現在では計算式と答を表示する ようになっているが、たとえば図3の問題構造の図などをうまく表示したり することによって、問題理解を助けるなど、いろいろな方法が考えられる。

3.2.6 システムの評価

システムの試作時点では、図3の構造を導くのに用いた問題211題のうち、そのまますぐに解けたのは59題に過ぎなかったが、解けなかった問題について原因を調べると、ほとんどが辞書の未整備によるものであった。

たとえば、211題中142題は、入力された日本語文章の意味解析の段階ですでにエラーを起こしており、しかも、そのエラーは、「ずつ」という語の処理、色に関する語の処理などの単語レベルでのエラー(115題)、助詞「の」の処理や並置句の処理など句レベルでのエラー(91題)が多く、文や文章レベルでのエラーはわずかであった(たとえば「……番目」といった序数を含む問題や、「どちらがどれだけおおいか」、「ちがいはいくつ

か」など特殊な設問形式の問題,「こどもが8にんいます。おんなのこは4にんです。おとこのこはなんにんいますか」のように,背景的な知識(子どもには女の子と男の子がある)を要する問題などがある。)。

これらのエラーを分析した結果、辞書と意味解析レベルの改良だけで、つまり問題表現の部分のアルゴリズムは変えないでも、211題中少なくとも 181題を解くことができるということがわかった。

一方、本システムを実際に児童に使用させてその反応を調べる試みも、非公式ながら実施された。この点については、未だ定量的な評価を行なう段階に至ってはいないが、実際にシステムを使用した子ども達の報告や使用時の観察から、子ども達がシステムから受動的に算数を教わるのではなく、ある問題を作ると「なぜ」答が出たり出なかったりするかを確かめようと、自発的に問題の構造に興味を持ち、別の問題を考えてみたりすることが多いことがわかった。この点は本システムの目的に合致することである。しかし、子ども達がキーボードの扱いに不慣れで入力に実験者の介助を要すること、自由に問題を作れるという環境にすると、とんでもない単語を使うことがあり、(たとえば友達の名前など)、システム側の辞書が追いつかないことなど、いくつかの問題点も浮かび上がった。

これらの点と教育効果を考慮すると、今のところ、児童とコンピュータ、 それにもう1人、教師か別の児童の3人で学習を行なう方法もありうると考 えている。

さらに、筆者のグループで、小学校の算数関係の先生方に、本システムの数室での利用に関してアンケート調査を行なった。アンケートは横浜市内の20小学校の100名の教員に依頼され、12小学校41名から返答が寄せられた。紙面の都合で詳しいことは省略するが、ここに述べたようなシステムがあったら数室で使えるだろうかという間には54%がなんらかのかたちで使えるだろうと答え、導入によって学習効果は上がるだろうかという間には100%が上がるだろうと答えている。特に、全体的には、このようなシステムが児童による問題の意味の理解を助けるだろうというコメントが多か

った。ただし、一般的なレベルでのネガティブなコメントとして、現在のところ小学校教員のコンピュータ理解のレベルが、コンピュータを教育に利用できるほど深くないことが多い(単にプログラムが書けるとかコンピュータンステムの知識があるというだけでは、小学校教育の道具としてコンピュータを使えるとは言い難い)こと、および、経済的に、大規模なシステムを教育に利用できる予算を持つ小学校は現在のところ少ないことの2点が多くあげられたことは、忘れるわけにはいかない。

3.2.7 関連研究と今後の課題

算数の文章題を解くシステムは、人工知能の分野で昔から研究されている。古くはBobrow の STUDENT がすでに1960年代に提案されており、最近でも日本では、東工大の志村教授の研究室などで研究が進められている。これらの研究は、自然言語処理と問題解決の統合化されたシステムの研究として今後も人工知能研究のひとつの分野を成してゆくであろうし、教育工学などへの応用も、今後試みられるようになるだろう。

ただ、はじめにも述べたように、人工知能ンステムの応用は、人間の知的機能を代替してくれるという範囲に留まるものではない。むしろ、人間の知的機能をよりよく引き出してくれるための補助的なシステムとしての応用を考えてゆく必要もある。上にのべたシステムは、未だ試作段階にあるが、この方向への発展をねらったものであり、今後さらに、マンマシンインタフェースの新しい側面を開発するためのプロトタイプとして、特に人間からコンピュータへのよりスムーズな働きかけを行えるインタフェースの開発を進めてゆきたいと考えている。

3.3 初等物理学の問題解決システム

3.3.1 何のためのシステムか

今度は、初等物理学、特にニュートン力学の問題解決システムの研究を紹介しよう。

この研究の主目的もまた、さきにのべた算数の文章題を解くシステムと同様に、力学の問題を入力するとそれを解いてくれるようなシステムを試作することであり、力学の問題の解法ではなく、問題の構造の理解を深めることを目的としたものである。ただし、算数のシステムと違って、対象は高校ないし大学初級の学生である。これら力学を初めて学ぶ程度の学生にとっても、難しいのは計算よりもむしろ問題自体の理解だと考えられるため、本システムが効果を持つ可能性があるわけである。

3.3.2 システムの概要

本システムを稼動することを当初の目的として、筆者らのグループでは、VAX11/750上にFRANZ LISP で書いたオブジェクト指向型言語システムOOPSを開発した(現在ではOOPSは物理問題解決システムと離れ、独自の方向で開発が進められている。)。ここでのべるシステムは、このOOPSを用いて書かれたもので、力学の問題におけるブロックや滑車、糸斜面や壁などをそれぞれオブジェクトとみなし、それらの間の力の関係をデオブジェクト間のメッセージパッシングによって表現することによって、問題の構造を自然なかたちでシステムの内部表現に落とすようにしている。

一方、問題理解のための知的教育工学システムとしての目的からいって、マンマシンインタフェース、特に学習者がスムーズに問題を作成してシステムに入力することができ、また問題作成時や問題を解いている途中での問題の構造がグラフィック表示されるようなインタフェースが望ましい。そこで本システムでは、メニュー形式の問題作成エディタを整備するとともにVAXとパソコンPC9801Fを接続し、PC9801F上のBASICを用いて図を描かせるようにしている。たとえば、滑車にブロックが3個ぶらさがっているような問題を頭に浮かべながら問題作成エディタで入力していくと、たとえば図4(a)のような図をグラフィック表示することができる。もちろん、入力途中の未完成の図も表示することができる。

入力された物理的対象 (滑車やブロックなど) は、自動的にオブジェクト 表現に変換される。そのあとで、システムのトップレベルのオブジェクトに メッセージを送ると、システムはまず物理的対象の間に働く力を求め、それに基づいて問題を解く。これらの力もグラフィック表示できるようになっており、たとえば図4(b)のように表示される。

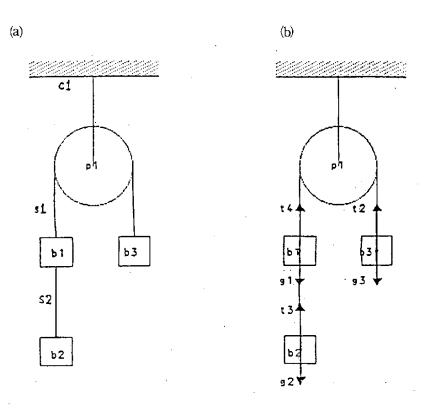


図4 滑車とブロックの問題のグラフィック表示 (a) 問題入力後 (b) 力の導出後

3.3.3 定性的推論と「系」の導出

本システムの特徴は、物理的対象間の力の関係を求めてから直ちにニュートン方程式をたてるのではなく、ひとつにまとめられる物理的対象はできる限りまとめたあとで方程式をたてるようにしていることである。

たとえば、図4(a)におけるブロックb1とb2は、合わせてひとつの物体 とみなしてもかまわない。このような、仮想的につくられた物体を、この研 究では「系」(subsystem)と呼んでいる。本システムでは,系を導出するために,定性的推論のための発見的ルールをいくつか持っており,それらを用いて物理的対象を系にまとめてゆく。たとえば,図 4 (a) の問題では,

「2つのブロックが鋭直方向に糸でつながり、上側のブロックは別の糸で さらに上方からつるされ、しかもこれらのブロックにかかっている力が これらの糸の張力、および重力だけであるならば、これらのブロックを ひとつの系とみなせ」

というルールを用いて、ブロックb1とb2を系にまどめることができる。

3.3.4 例 題

オブジェクト表現の細かい内容やOOPS自体の紹介は本稿の目的ではないので別の機会に譲り、ここでは例題として次の問題を考えることにしたい。

「図5のように、5個のブロックが置かれています。ブロック b 1、b 2、b 3、b 4、b 5 の重さは、それぞれ、5 kg、3 0 kg、1 0 kg、1 5 kg、5 kgです。さらにブロック b 5 は、図に示すように、右側から 2 0 0 Nの力で押されています。なお、摩擦はないものとします。このとき、(a) 糸 s 1 の張力を求めなさい。(b)(a) の結果を利用して、ブロック b 3 の加速度を求めなさい。(c) ブロック b 1 の加速度はいくらになりますか。」 …………… (3)

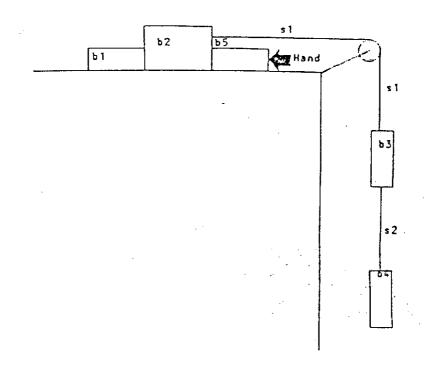


図5 問題(3)のグラフィック表示

上の問題(3)を解く過程を図6に示す。問題を解くためには、まずmonitorというオブジェクトにstartというメッセージを送る(図6(分)。これによってシステムが起動し、まず問題のゴールが何か(上の問題の(a)では張力)などを学習者に聞いてくる(図6(中)。これらが済むと、注目すべき物理的対象が定められ(図6(十)、次に、物理的対象間に働く力が求められる(図6(中)。それから、物理的対象を系にまとめるための定性的推論ルールが働いて、まず図5のb3とb4が系(kei-1)にまとめられ、さらにこの系に働く力が求められる(図6(中)。次に、b5とb2もひとつの系(kei-2)にまとめられ(図6(中)、さらに、kei-1とkei-2がひとつの新しい系(kei-3)にまとめられる(図6(中)。次に、このkei-3 についてのニュートン方程式がたてられ、それが加速度について解かれる(図6(中))。さらに、この加速度の値を用いて、糸s1の張力が求められる(図6(中))。

これで問題(3)の(a)が解けたことになる。問題(3)の(b)と(c)については省略するが、上と同じように解ける。図7に、上の問題についての力のグラフィック表示の例を示しておく。

```
(1)
monitor start
>>>>> Please input goal object(s) ((((
                                                                                     (m)
etringl
))))) Please imput goal attribute (((()
tension
>>> all#objects --> (hand) block! block2 block3 block4 string! string2
pulley! table! block5)
                                                                                     (1)
>>> attention --> (block1 block2 block3 block4 block5)
))) partial#attention --> ((block2 string1 block3 string2 block4))
>>>>> Make 1 gravity-1 of block1
>>>>> Make : gravity=2 of block2
>>>>> Make : gravity=3 of block3
>>>>> Make : gravity-4 of block4
>>>>> Make ': gravity-5 of block5
                                                                                     (<u>--</u>)
)>>> Make : hand-force-1 of block5
>>>> Make : resistance-| of block1
>>>>> Make : push-force-l of table1
>>>>> Make : resistance-2 of block2
>>>>> Make : push-force-2 of table!
>>>>> Make : resistance-3 of block5
>>>>> hake : push-force-3 of table1
***** CURRENT GOAL ---> make-one-kei
>>>>> (block3 block4) are regarded as kei-1.by gravity-Kei strategy.
                                                                                     (元)
>>> Change : gravity-3 of block3 ---> force-1 of Kei-1
>>> Change : gravity-4 of block4 ---> force-2 of Kei-1
>>> Composit : force-L + force-2 --> force-3 of Kei-L
>>> attention --> (block! block2 kei-1 block5)
>>> partial*attention --> ((block2 string1 kei=1))
 >>>> (block5 block2) are regarded as Kei-2 by force-Kei ethategy-
 >>> Change : hand-force-t of block5 ---> force-4 of Kei-2
                                                                                     (~)
 >>> attention --> (blockt Kei-1 Kei-2)
 >>> partial*attention --> ((kei-2 string1 kei-1))
 >>>>> (Kei-2 kei-1) are regarded as Kei-3 by string-Kei strategy.
 >>> Change : force-4 of Kei-2 ---> force-5 of Kei-3
 >>> Change : force-3 of kei-1 ---> force-6 of kei-3
                                                                                     (F)
 >>> Composity: force=6 - force=5 --> force=7 of kei=3
 >>> attention --> (block1 kei-3) *
 >>> partial*attention --> ((kei-3))
```

図 6 問題(3)の解決過程

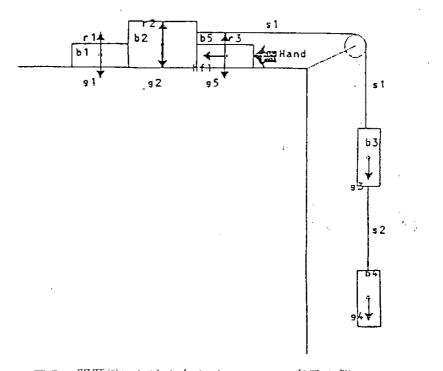


図7 問題(3)における力のグラフィック表示の例

3.3.5 システムの評価

現在、本システムでは、ブロック、滑車、斜面、床、天井、壁をふくむかなり複雑な問題を扱うことができる。摩擦については単純な問題ならば解けるが、一般的なかたちで解けるようにはなっていない。ただしこれは、システムの問題というよりは、ニュートン力学そのものの問題といった方がよい。また、本システムの本来の目的である問題のより深い理解を助けるという点については、未だ定量的評価の段階には至っていない。特に、問題作成エディタとグラフィック表示をはじめとするインタフェースについては、より一層スムーズなマンマシンインタラクションを可能にするようなシステムに移行してゆく必要がある。しかし、これまでの評価では、学習者に積極的な興味を起こさせるばかりでなく、学習者自身が自分でよく似た問題をいくつか作り、それらをシステムに解かせてみて、問題の構造の一見小さな違いでも、それが力学の理論から見て質的に大きな差異であれば、解決過程も大きく変わることがありうるといったことを実感として理解するのに役立つ可能性があると考えられる。

3.4 おわりに

本稿では、問題の構造的理解を助けるための知的システムの例として、算数と物理に関する2つのシステムを紹介した。これらはともに未だ試作の段階にあるが、知的情報処理システムの応用として、人間の知的機能を代替してくれるシステムばかりでなく、より高度の知的機能の発揮を助けてくれるようなシステムもありうるということが、本稿でのべた例から示唆できたとすれば幸いである。

第Ⅳ編 技術トピックス

			•
		-	

第Ⅳ編 技術トピックス

第1章 国産マイクロプロセッサ「Vシリーズ」

マイクロコンピュータ分野において、着々と実績を積み重ねてきたNEC日本電気は今や世界のマイクロコンピュータ市場の2割のシェアを占める世界のトップメーカとして、その地位を確保している。更に近年では他の国産メーカもCPUの生産シェアの上位にかなり食い込んできているが金額ベースでは未だに欧米メーカには追いついていない。

これは4ビット/8ビット系のシングルチップにおいて、国産製品が世界に浸透してきた事を意味していると同時に8ビット/16ビットマイクロプロセッサにおいては国産メーカのほとんどが、未だ欧米メーカのセカンドソーサとなっている為であり、より高度なハイエンドマイクロプロセッサにおいては、セカンドソーサである為、常に開発が立遅れてきた事を示すものでもある。

そこでNEC日本電気はオリジナル路線により、名実ともにマイクロプロセッサのトップメーカとしての地位を築き上げ、より高性能なマイクロプロセッサをより早く市場に出すため、NEC独自の"Vシリーズ"の開発に踏み切ったわけである。

そしてVシリーズ開発のねらいは,

- ① 高速 CMOSプロセスの採用
- ② 従来品との継承性を重視した新しい16ビットアーキテクチャ
- ③ 斬新な32ビットアーキテクチャ

この3つが柱となっておりこれに加えて周辺チップ、リアルタイムOS、開発支援ツールなどを含めたトータルサポートの推進が基本的な方針と言えるであろう。 (図1-1)今回、御紹介するのは、まずVンリーズの第一弾の製品として開発をすでに完了している $\mu PD70108(V20)/\mu PD701116(V30)$ とその周辺 LSIそして今年製品化が予定されている $\mu PD70208(V40)/\mu PD70216(V50)$ 、最後にこれらの品種をサポートする開発支援システムである。

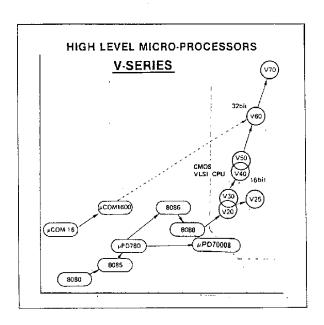


図1-1 ハイレベルマイクロプロセッサVシリーズ

1.1 \(\mu \text{PD70108(V20)} \/ \mu \text{PD70116(V30)} \)

V > y - x第一弾として開発された μPD70108(V20)/μPD70116(V30) (以下 V20/V30と呼ぶ)は、従来 CPUとの継承性を重視しながら独自の新しいアーキテクチャを駆使した 1.6 ビットマイクロプロセッサである。V20/V30ともに内部は完全 1.6 ビットのアーキテクチャを採用しているが、V20は外部データバスが 8 ビット幅であり、V30は外部データバスが 1.6 ビット幅となっている。このため要求されるシステムに対応した形で V20/V30を使い分ける事が出来る。以下に製品の特徴を示す。

- 8/16ビット(V20/V30)データバス
- メモリ空間1Mバイト、I/O空間64Kバイト
- CMOSブロセスの採用による低消費電力化
- 専用ハードウェアによる命令実行の高速化
- 豊富な101種の命令セット(μPD8088/8086の84種を含む)
- μPD8080AFエミュレーション機能
- 5 M Hz / 8 M Hz クロック動作

次に主な特徴である、CMOSプロセス、ハードウェア、命令セット、 μPD 8080AFエミュレーション機能の 4 点について簡単に述べる。

(1) **CMOS**プロセス

V20/V30 はプロセスとして高速CMOSプロセスを採用しており動作 電力を従来の名とし低消費電力化を実現している。また、プログラムの待機 中に消費電力を更に低減するスタンバイ機能を備えており、この機能により 一部の回路を除いてクロックの供給が停止され消費電力が通常動作名とする 事が可能である。

(2) ハードウェア

図1-2にV20/V30の内部ブロック図を示すV20/V30は命令実行:

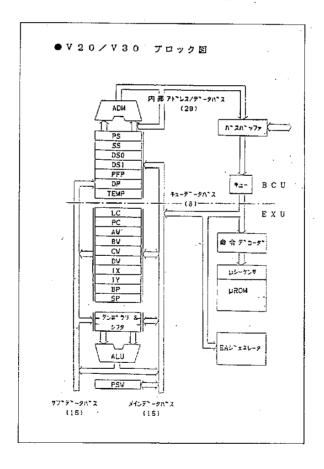


図1-2 V20/V30ブロック図

の高速化を計るため、内部構造に新しいアーキテクチャを採用している。従来CPUにおけるシングルデータバスに対し、V20/V30 はデュアルデータバス方式を採用しており、更に実効アドレスの生成回路、ループカウンタの内蔵、プログラムカウンタとしてBCU(バスコントロールユニット)にPFPを、EXU(実行ユニット)にPCを内蔵し実行クロック数の短縮を実現している。

(3) 命令セット

V20/V30 は 101 種の命令セットを用意している。これらの命令セットのうち 84 種類は従来のマイクロプロセッサ μ PD8088/8086 の命令セットとオブジェクトレベルで同じ形式を採用している。しかしながら前項で述べた様な高速処理のための専用ハードウェアの採用により実行処理速度の改善が計られており、平均 1.3 倍~ 1.5 倍の速度 U P を実現している。更に追加された 17 種の命令には、画像データ処理等に有効なビット操作命令及びビットフィールド操作命令をはじめ、10 進のストリング演算命令、高級言語サポート命令等を用意し、要求されるシステムに対応したオリジナル命令を追加している。

(4) µPD8080AFエミュレーション機能

V20/V30 には、本来の命令セットを実行するネイティブモードと#PD

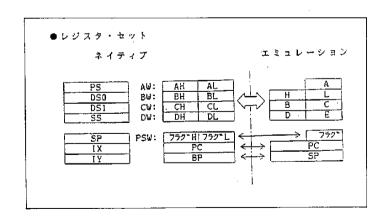


図1-3 レジスタ・セット

8080 AF の命令セットを実行するエミュレーションモードがある。この機能により現在まで開発された膨大な量の $\mu PD8080 \text{AF}$ 用のプログラムを V20 V30 上で実行する事が可能となる。

エミュレーションモードでは図1-3に示すようにネイティブモードで使用されているレジスタ資源を共有して使用する。

ネイティブモードからエミュレーションモードへの状態遷移図を図1-4 に示す。ネイティブモード中のBRKEM 命令によりエミュレーションモード へ制御が移り、エミュレーションモードにおけるRETEM 命令によりネイティブモードに戻る。なおエミュレーションモード中での外部インタラプト及 びCALLN 命令によってネイティブモードに移行し割込み処理がなされRETI 命令によって再びエミュレーションモードに戻る。

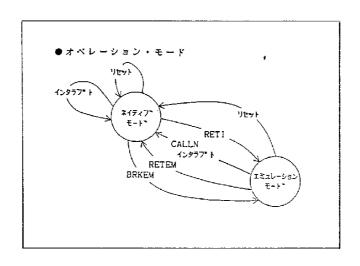


図1-4 オペレーション・モード

(5) 従来 CPUとの性能比較

V20/V30 の従来 CPU との性能比較としてベンチマークテストの結果を図1-5に示す。

これはそれぞれのCPUの同一周波数における実効クロック数をV20を

相対パフォーマンス (V20=1.0)

		V20/V30	8088/8086	68000
EDN	Ε	1.00/0.96	1.27/1.24	1.33
インチ マーク	F	1.00/0.88	1.89/1.77	1.12
\ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \	Н	1.00/0.77	1.30/1.09	0.61
キヽフヽソンミ	シクス	1.00/0.74	1.71/1.42	1.19

EDN APR.'81,SEP.'81 E:Character string search
F:Bit set,reset,test
H:Linked list insertion

図1-5 パフォーマンス比較

1.0 とした時の相対値で示している。

このデータからもV20/V30 は従来CPUよりパーフォーマンスが高い製品である事がわかるであろう。

1.2 周辺LSIシリーズ

NEC日本電気はV > y - xの第一弾の製品であるV20/V30 の性能を十分に引き出すための周辺LSIをV20/V30 と同時に開発しすでに製品化を完了している。

V20/V30 をサポートするLSIとしてはまずメモリやI/Oベリフェラルとのバスインターフェースを扱う μ PD7108Xシリーズとクロックジェネレータ μ PD71011, マイクロプロセッサの周辺インターフェース用LSIとして μ PD7105 χ シリーズ及び μ PD71071を用意している。

 $\mu PD7108 X シリーズ及び \mu PD71011はすべて高速 CMOS プロセスを採用しており基本ゲート遅延時間を <math>0.7ns$ と高速化を実現している。

μPD7105 Xシリーズ及び μPD71071はマイクロプロセッサと周辺機器とのインターフェースを効率よく行う上で好適なデバイスである。なお、すべて CMOS プロセスを採用しているため、従来の周辺インターフェース用LSIと 比較し低消費電力化を実現し、実に機能的にも向上させている。以下表 1 − 1 に各製品の一覧を示す。

表1-1 Vシリーズ周辺LSI

品	名	プロセス	機能	パッケージ
#PD710	11C		クロック パルス ジェネレータ/トライバ	18PIN DIP
#PD710	82/83C	CMOS	8ビットラッチ	20PIN DIP
#PD710	86/87C	CMOS	バスドライバ/レシーパ	20PIN DIP
#PD710	88C		システム バス コントローラ	20PIN DIP
μPD710	5 1 C/G		シリアル コントロール ユニット	28PIN DIP/44PIN FLAT
μPD710	5 4 C/G		プログラマブル タイマ・カウンタ	2 4PIN DIP/4 4PIN FLAT
#PD710	5 5 C/G	CMOS	ハラレル インターフェース ユニット	40PIN DIP/44PIN FLAT
μPD710	59C/G		インタラプト コントロール ユニット	28PIN DIP/44PIN FLAT
#PD710	71C		16ピット DMA コントローラ	4 8PIN DIP

以上の周辺LSIを用いたV20/V30のシステム構成例を図1-6に示す。構成デバイスがすべてCMOSプロセスを採用しているためシステム全体の消費電力を低減する事を可能としている。

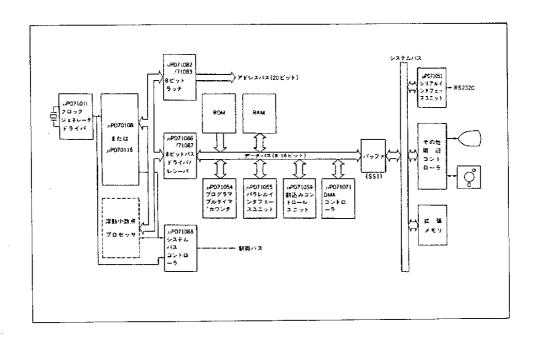


図1-6 V20/V30システム構成例

1.3 $\mu PD70208(V40)/\mu PD70216(V50)$

μPD70208(V40)/μPD70216(V50)(以下V40/V50と呼ぶ)は,

V imes y = V imes y imes y = V imes y imes y = V imes y ime

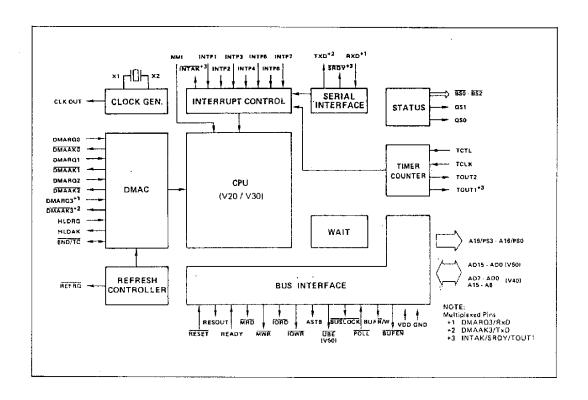


図1-7 V40/V50ブロック図

以下、内部の各ブロックにおける機能概要について簡単に述べる。

(1) CPU部

V40/V50に内蔵されている CPUは前述の V20/V30 を採用しているため、 V20/V30 用に開発されたプログラムをそのまま実行でき、ソフトウェア上、完全に互換性が保たれている。従って、 CPUがアドレス可能なメモリ空間及び I/O空間はそれぞれ 1Mバイトと 64Kバイトであるが、このうち、 I/O空間の上位 16 バイトは、 内蔵ペリフェラルのリロケーションレジスタ等に使用しているため専用空間として割付けられている。

(2) 割込みコントローラ

内蔵される割込みコントローラは μPD71059 のサブセットとなっており、 8本の割込み要求を調停しCPUに割込み要求を発生すると同時に該当する 割込みベクタをCPUへ知らせる役割を持つ。割込み要求は7本。 INTP 端子を通して外部 I / Oから発生するものと、内蔵されている周辺コントロラーラであるタイマ/カウンタ、シリアルインターフェースから発生するものとがある。

-- (3) タイマ/カウンタ

V40/V50は16ビットタイマ/カウンタを3ch内蔵しており、カウンタのは出力を割込みコントローラの最高優先としているためシステムタイマとして利用でき、カウンタ1はシリアルインターフェースと共に使用しボーレートジェネレータとしての利用が可能である。また、カウンタ2は入力としてトリガ入力を備えており、汎用タイマとして使用する事ができる。なお、3個のタイマの入力クロックは、CPUのシステムクロックのプリスケーラによる分周入力または、外部のクロック入力端子からの入力を選択することが可能である。

(4) **DMA**コントローラ

内蔵されているDMAコントローラは #PD71071 の機能をほとんどサポートしており、外部端子として4chの独立したDMA端子を持っている。なお、内部レジスタは1Mバイトのメモリアドレッシングが可能な20ビットのアドレス・レジスタと最大64Kバイトのデータ転送を可能とする16ビットのカウントレジスタを有している。DMAコントローラを内蔵することにより、システムの小型化が計れるばかりではなく、CPUとDMAコントローラのバスの使用権のスイッチングを高速に行う事が可能となる。

(5) シリアルインターフェース

V40/V50 は各種端末機とのシリアルデータ通信用として、シリアルインターフェース(非同期)を備えている。端子としてRXD.TXDと \overline{SRDY} を有し、内部に送信バッファ/受信バッファを独立して備えている。ボーレートは最大 3.8.4 Kボーであり高速なシリアル転送が可能となっている。実にキャラクタ長を 7 or 8 に、ストップビットを 1 or 2 に設定でき、パリティの発生、及びオーバーラン、フレーミング、パリティエラー等のエラー検出

機能を備えている。

(6) DRAMリフレッシュコントローラ

V40/V50は外部メモリをDRAMとした場合の、メモリに対するリフレッシュ制御機能を内蔵しているため、DRAM周辺をコンパクトに構成する事を可能にしている。このリフレッシュ制御機能は、内蔵されるバスアービタによりCPUのバスの空を検出してリフレッシュサイクルを駆動するサイクルスチール方式を採用しているため最少のオーバーヘッドでリフレッシュが実現でき、更にバスの空が検出されずにリフレッシュ要求がある一定数に達するとバスの使用優先順位が最高優先となりDMA転送中においてもリフレッシュサイクルを駆動することが可能となる。

(7) V40/V50のシステム構成例

V40/V50 を用いた一般的なワークステーションへの応用例を図1-8

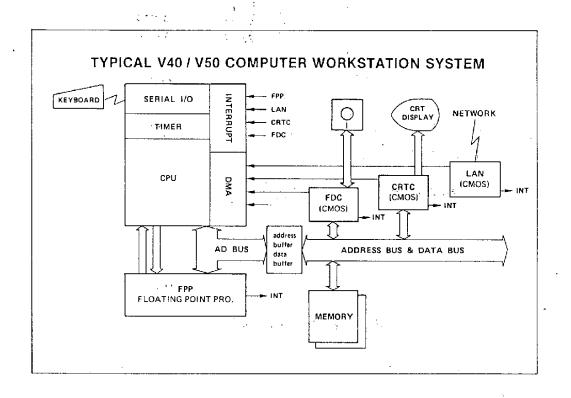
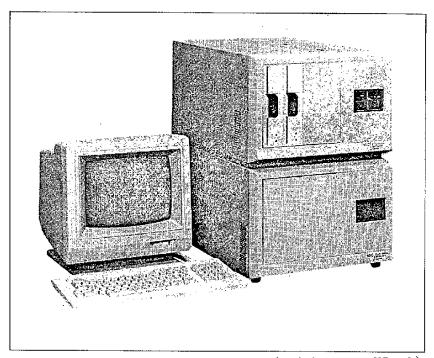


図1-8 V40/V50のワークステーションへの応用例

に示す。V40/V50 は標準的な周辺コントローラを内蔵しているため、システムをコンパクトに設計する事ができ、更に、CPU部にV20/V30 を採用しているため、V20/V30 システムからV40/V50 システムへの移行を容易に行うことができる。

1.4 Vシリーズ開発支援システム

V > y - xの第一弾の製品である V20/V30及び今年製品化される V40/V50,4品種の開発をすべてサポートするホストマシンが図1-9に示す MD-086 シリーズである。 MD-086 シリーズにはハードディスクペース



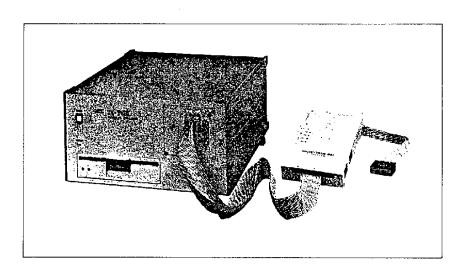
(写真はMD-086 HD-10)

図 1-9 MD-086 シリーズ

のMD-086HD-10と、フロッピーディスクベースのMD086FD-10があり、更にフロッピーベースからハードディスクベースにバージョン \mathbf{UP} を可能とするMD-086DKを準備している。 このMD-086シリーズによっ

てVシリーズをはじめ 4 Bi t から 1 6 Bi t までのすべてのN E Cのマイクロプロセッサを開発する事ができる。Vンリーズでは,MD-0 8 6 上で動作可能なリロケータブルアセンブラを用意しており,更にVシリーズ用のCコンパイラについても開発中である。

V20/V30及びV40/V50 用のデバッガとしては,スタンドアロンで動作するインサーキットエミュレータを用意しており,V20/V30 用については製品化を完了している。V20 用の IE-70108-S 及びV30 用の IE-70116-S は互いにポッド部の交換にて両方の用途に使用でき,更に今後製品化されるV40/V50 についてもコントロールボードとポッド部の組合せでそれぞれの用途に合わせて変更する事が可能である。なお,フロッピーディスク 1 台と内部にメモリディスクを内蔵しているため,ホストマシンMD-086 シリーズのリンカーの出力ファイルをそのままフロッピーディスクに入れ,ターミナルからのコマンドによりシンボルファイルをメモリディスク上に生成する事により高速なシンボルデバッグが可能である。更にV20/V30 の特徴である $\mu PD808080$ AFのエミュレーション機能についても完全にサポートしている。



 $\boxtimes 1 - 10$ IE-70116-S

1.5 Vシリーズの今後の展開

NEC日本電気は、今回御紹介したVシリーズの今後の展開として、32ビットの内部アーキテクチャをもつ次世代のマイクロプロセッサとしてV60/V70の開発を進めている。V60/V70は仮想記憶管理機能を備えた新しい32ビットマイクロプロセッサである。V60は内部バスが32ビット、外部バスが16ビットの構成となっており、V70は内部バス、外部バス共に32ビットの構成である。これらのマイクロプロセッサは、もはやスーパーミニコン並の性能を有しており、今後、製品化されるであろうスーパーパソコン、スーパーワークステーション等、幅広い用途に使用可能な製品となるであろう。

第2章 オブジェクト指向型言語 Smalltalk - 80

Smalltalk-80は、富士ゼロックス(株)が国内販売している言語であり、かつ開発環境である。以下、その特徴と応用分野について解説する。

2.1 開発の歴史と特徴

次世代のプログラミング言語として注目されている人工知能研究開発用の Smalltalk-80は、1970年代初頭から米国ゼロックス社のパロアルト研究 所で研究開発されてきたソフトウェアシステムで、1983年の発表以来、コン エピューターサイエンスの分野に大きな反響を巻き起こしている。その大きな特 徴は,オブジェクト指向型である。BASICやPascal,FORTRANなど従 来の言語が手続き指向型であるのに対し、Smalltalk-80は、オブジェクト 指向の言語を代表している。オブジェクトとは、物体、あるいは対象の意で、 オブジェクト指向とは、対象にメッセージを送る行為自体がプログラムになる ことを意味する。対象がメッセージに応じたアクションを起こすことで、処理 は進展する。これは、複数の人間が仕事を進めていくようすをプログラムの世 界に取り入れたもので、理解しやすく強力な形態のプログラミング言語を実現 している。 Smallalk-80を使用してプロトタイピングを行い、次に本シス テムを開発する方法をとれば、 FORTRANや COBOL に比べ、数倍の生産性を 実現できる。さらに,オブジェクトをひとつひとつの知識に対応させることに より、知識ベースシステムを構築することも可能である。Smalltalk-80に よってすでに開発されたシステムとして,ニューヨークタイムズ日曜版のレイ アウトシステムがある。このシステムは,システムの仕様に変更が入った場合 など本質的な修正でも、容易に、しかも単時間のうちに行うことができる。こ のほか、Smalltalk-80の応用分野には、CAI(コンピューター支援によ る教育)、シミュレーションなどがある。このように、オブジェクト指向型言 語の代表である。Smalltalk-80は、これらの人工知能分野における有力な 言語といえる。

2.2 Smalltalk-80の概要とソフトウェア開発

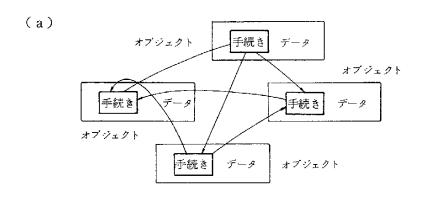
Smalltalk-80は言語仕様であり、かつ高度なユーザーインターフェイスを提供するソフトウェア開発環境である。それと同時に、Smalltalk-80は、ソフトウェアの生成、保守をどのように行うかというソフトウェアのライフサイクルの立場からとらえることができ、その場合、従来のプログラミング言語とは全く異なる様相を呈する。Smalltalk-80により既に開発されたシステムとして、ニューヨークタイムズの日曜版のレイアウトシステムがあるがこのシステムの特徴は、その保守性にある。システムの仕様に変更が入った場合、つぎあて的にではなく、本質的に、修正が行われ、しかもその生産性は極めて高い。ソフトウェア生産性改善への期待は大きく、これに対する一つの回答を与えているのが、Smalltalkである。Smalltalk-80は、Smalltalk-72とそれを反省材料として創造された Smalltalk-76を祖先にもつ、この3つの各バージョンで行われた研究開発のステップは、次のとおりである。

- ① ゴール設定
- ② 実 装
- ③ ユーザへの配布
- 。 ④ ユーザからのフィードバック

この間の経験を通じて、Smalltalk-80は、これを構成するプログラムモジュールであるクラスが、種々の変更に対し弾力性のある強力なものとなっていった。250のクラスは、階層構造をもち、それらはお互いに密接な関係を持ちつつ、きれいに分離することができる。ユーザは、これらのクラスと、その中に含まれる手続きである合計5000のメソッドを使用し、プログラムの開発を行うのである。必要な機能を実現してくれる部品、すなわちオブジェクトを取りそろえ、あるいは変更により作り出し、一つのシステムを構成する。これが従来の言語システムと異なる。Smalltalk-80流のソフトウェア開発である。

2.3 オブジェクト指向とデータ/手続き指向

オブジェクト指向とデータ/手続き指向の考え方の違いは、図2-1のように表すことができる。オブジェクト指向のソフトウェア・システムは、図2-1(a)のようにオブジェクトとオブジェクトとの間でメッセージをやりとりする。データと手続きが一つのかたまりになっており、データ構造の変更があってもそれは一つのオブジェクト内の問題であって、他のオブジェクトには波及しない。一方、データ/手続き指向のソフトウェアは、図2-1(b)のように手続き間の呼び出し(Call と return)や、手続きからデータへのアクセスという形で実現される。この場合、一部の手続きやデータの変更が他に影響しやすいという問題が生じる。この問題を回避し、ソフトウェアの生産性や保守性を向



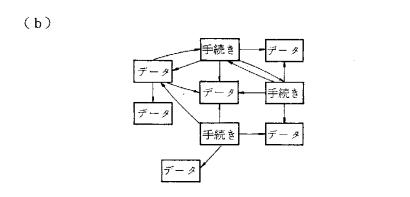


図2-1 オブジェクト指向とデータ/手続き指向

上させる有力な手段として、オブジェクト指向プログラミングが注目されているのである。

Smalltalkは、このようなオブジェクト指向プログラミングの枠組みを提供する、最初のソフトウェア開発環境である。

2.4 Smalltalkの応用分野

Smalltalkは、現段階では実際にこれを使ったアブリケーションの実積はほとんどなく、その考え方や内容を多くの人達が研究している段階である。従って、Smalltalkの特徴からどのような応用分野に適合するかを考察すると、次の4つの分野への応用が期待される。

- ① プロトタイピング
- ② 知識ベース
- ③ シミュレーション。
- (4) CAI

この応用分野と Smalltalk の特徴の対応は表 2 - 1 のとおりである。

表 2-1 Smalltalk の特徴と応用分野

応用分野	Smalltalkの特徴
プロトタイピング	システムの構成要素がすべてオブジェクトからなり、すべてのオブ
	ジェクトの更新状況を管理する仕組みがあること。
知識ベース	クラスおよびインヘリタンス(相続)の概念がある。
	すなわち,クラスには親子関係があり,子は親の性格を受け継ぐと
	いう性質は人口知能の分野でいうフレームの性質と共通であること。
シミュレーション	インスタンスは、クラスという母親から生まれた子供であるが、イ
	ンスタンス同士は似通った性格をもつ。このインスタンスの性質を
	利用。

(1) プロトタイピング

従来のプログラム開発では, できあがったシステムを実際に使用してみな

ければ、その良否を確かめられなかった。しかも、いったん組み上げたソフトウェアを修正することは困難であり、ユーザは不便を強いられてきた。このような難点を解決し、ユーザにとって、より使いやすいプログラムの開発を可能にしたのが、プロトタイピングと呼ぶ手法である。これは、実験的に作成したプログラムをユーザの要求に応じて改良しながら、実際のプログラムをつくりあげる方法である。プロトタイピングには、仕様が明確でない段階からでもプログラミングできる言語が必要で、従来のプログラミング言語には、この要求に十分応えうるものはほとんどなかった。ゼロックス社が開発したSmalltalk-80は、プロトタイプ・システムの開発を可能にし、さらにこれからの人工知能分野で広く応用される最も新しい汎用言語である。

(2) 知識ベース

従来のデータベースシステムでは、データ(ファクト)の管理が中心に行われていたが知識(ファクト+ルール)の管理をする、いわゆる知識ベースシステムの確立が、今後のコンピュータシステムの大きな課題といわれている。Smalltalk-80ではこの知識をクラスに対応させることにより、強力で柔軟性のある知識構造をつくりあげることができる。クラスとして表現された知識は、その知識がどのように使用されるかについて知っている。これを知識ベースシステムの構成の視点から見た場合、if-added (知識ベースに情報が付加された場合に機能する手続き)、if-removed (知識ベースから情報が削除された場合に機能する手続き),if-needed (必要な状況において情報の生成を行う)などの機能は、まさにSmalltalk-80のメソッドにより容易に記述できることを示しており、インヘリタンス(相続)の機能とともに、知識ベースシステム構築に有効なシステムであることを示している。知識ベースシステムにインプリメントされるべき機能は、Smalltalk-80の基本機能となっている。



if-added 知識ベースに情報が付加された場合に機能する。 if-removed 知識ベースから情報が削除された場合に機能する。 if-needed 必要な状況において情報の生成を行う。

図2-1 知識ベースのもつ3つの基本操作

2.5 Smalltalk-80の動作環境

Smalltalk-80は、富士ゼロックス1100SIPで稼動する。 この1100SIPの仕様は、表 2-2のとおりである。

表 2-2 Smalltalk-80を実行する 富士ゼロック 1100 SIP の仕様

プロセッサ	
主記憶	2 M ペイト
仮想アドレス空間	8 M バイト
サイクル時間	200ナノ秒(マイクロインストラクション)
固定ディスク	2 9 Mバイト
ロカルネット接続	イーサネット
外部インターフェイス	R S 2 3 2 C
ディスプレイ	
画面	17インチ
表示サイズ	1 0 2 4 × 8 0 8 F y F
走査方式	ラスタースキャン方式
キーボード/マウス	
キーボート	非エンコード6 4 キー
ポインティングディバイス	3 ポタンオプティカルマウス
外部インターフェイス ディスプレイ 画面 表示サイズ 走査方式 キーボード/マウス キーボード	RS232C 17インチ 1024×808ドット ラスタースキャン方式 非エンコード64キー

また、Smalltalk-80の実装方式は図2-2のとおりである。

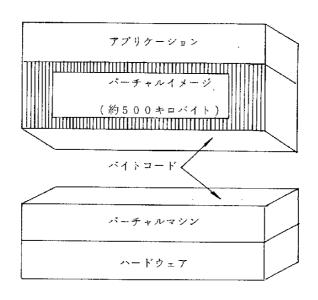


図2-2 Smalltalk-80の実装方法

バーチャルイメージは、Smalltalk-80システムを記述した、大きさが約500キロバイトのソフトウェアで、テキスト編集のためのエディタ、プログラムミスを発見するソースレベル・デバッガやダイナミックローダなどのプログラム開発ツール、プログラムミスを未然に防ぐスペリング・コレクタなどが含まれている。

バーチャルマシンは、バーチャルイメージにあるソフトウェアを実行するハドウェアで、Smalltalk-80の命令コードであるバイトコードを実行する。

第3章 エキスパート・システム構築用ツールOPS5

OPS5は、日本DECが販売しているエキスパート・システム構築用ツールである。

これを用いると、エキスパート・システムに必要な知識や経験則を、IF-T-HEN (イフ・ゼン)ルールの形式で表現するだけで実用的な人工知能の応用システムが構築できる。

つまり、Lispを人工知能のアセンブラに例えるならば、OPS5は、人工知能の簡易言語とも言えるのである。

3.1 OPS5の概要

OPS5は、米国カーネギー・メロン大学で開発された汎用のエキスパート・システム構築用ツールである。開発されて以来、現在までにDEC社内外で様々なエキスパート・システムの実用化に使用され、実績もある。

実際に実用化された例としては、DECのXCON, XSEL、米空軍AIRP-LANなどが有名である。

OPS5は、DECのスーパー・ミニコンピュータVAXファミリで利用可能で、それ以外に特別なハードウェアは不要である。従って、エキスパート・システムに必要なハードウェアの能力、あるいは価格に応じて、VAXファミリの中から、最適のコンピュータ・システムが選択できる。また、VAX/VMSオペレーティング・システムのもとで、効率のよいシステム開発ができる。

さらに、OPS5は、BLISSと言う言語で記述されている。BLISSは、DECでシステム記述のために用いているプログラミング言語で、*UNIXの記述に用いられたC言語に似た言語である。BLISSを用いて記述したことによって、LISPや、Prolog などで同種のソフトウェアを開発するよりも少ないメモリで、しかも非常に高速の推論を行うことが可能となった。

また、VAX/VMSの標準呼び出し規則に従って開発されているので、他の高

(脚注) * UNIXは、米国ベル研究所で開発されたソフトウェアである。

級言語やアセンブラとも容易に結合して利用できる。つまり、アプリケーションに応じて、エキスパート・システムの利点と、従来の高級言語の利点をうまく組み合せて、効率のよい最適のシステムが構築できるのである。例えば、研究所や工場のようなところでの、リアルタイム・アプリケーションの場合には、測定器などからのデータをリアルタイムで取り込む部分は従来の言語で記述し、そのデータを用いて推論を行い結論を導き出す部分の記述にOPS5を用い、またその結論から機器の制御をする部分にも従来の言語を用いる、といった利用方法が可能である。他の言語との結合のみでなく、データベースなどのソフトウェアとも結合し、利用できる。つまり、複数の人が使用するシステムのような場合には、そのシステムの利用するデータを、集中的に管理し、共有して利用できるということである。

また、OPS5では、知識や経験則を表現するIF-THENルールの形式が 非常に簡単である。さらに、ルールの記述、つまり、プログラミングに日本語 (漢字)が利用できるので、作成したルールが読みやすく、プログラムの保 守も容易である。

3.2 OPS5の構成

OPS5を利用して構築するエキスパート・システムは、3つの要素から構成される。

プロダクション・メモリ、ワーキング・メモリ、インタプリタの3つである。

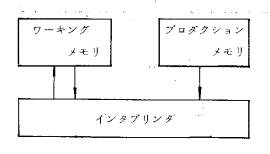


図3-1 OPS5の構成

プロダクション・メモリは、IF-THEN ルールの形で知識情報を格納する 知識ベースである。この中には、知識情報が利用しやすい形に整理されている。

ワーキング・メモリは、プロダクション・メモリの中から、使用する知識情報、つまりルールを選択する前提となる情報を格納するデータベースである。 この内容は、選択されたルールの実行によって更新され、次に使用するルール の選択に用いられる。

インタプリタは、プロダクション・メモリ内のルールと、ワーキング・メモリ内の情報を用いて、結論を導びく推論エンジンである。インタプリタは、この他に、利用者とのデータの入出力も行う。

OPS5を利用したエキスパート・システムは、以上3つの要素から構成されるが、OPS5には、この他、コンパイラもある。しかし、OPS5のインタブリタとコンパイラは、BASICなどで言うインタブリタとコンパイラとは性格の異なるものである。

インタブリタは、前述のとおり、推論エンジンであり、エキスパート・システムを利用するために不可欠である。これに対し、OPS5のコンパイラは、エディタなどを用いて作成した IF-THEN ルールのファイル、つまりプログラムをインタブリタが利用しやすい形式に整理してコード化しプロダクション・メモリを作成するものである。すなわち、OPS5の場合、エキスパート・

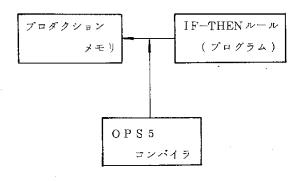


図3-2 OPS5コンパイラ

システムの開発に用いるのがコンパイラで,実際の運用に必要なものがインタ プリタである。

3.3 OPS5の知識表現

OPS5において、知識情報などのデータを表現する形式を紹介する。

まず、推論をすすめる前提となる情報を格納するワーキングメモリ内のデータの表現形式を見る。ワーキング・メモリには、ワーキング・メモリ要素と呼ばれるデータが格納される。それぞれのワーキング・メモリ要素は、カッコで囲まれ、最初にあるのは、その要素が何であるかを示めすオブジエクト名であ

図3-3 ワーキング・メモリ要素

る。そして、その後にいくつか、要素の性質を表現する属性の名前とその組が 並ぶのがワーキング・メモリ要素の型式である。(図3-3(a))属性の名前は、 その値と区別するために最初にアップアロー(上向き矢印)を付ける。

例えば、赤くて重さが10 Kgの箱があるという情報を表現するためには、図3-3 (b)のような要素をワーキング・メモリ内に格納すればよいのである。

ワーギング・メモリの内部には、このような情報を表現するための要素がたくさん格納され、推論を進めるための前提条件として用いられるのである。ワーキング・メモリ内では、単に、こういった情報がある。ということを記録するだけでなく、推論のために必要な情報として、その情報がいつワーキング・メモリ内に作成されたかを示すタイム・タグと呼ばれる一連の整数値がそれぞれの要素に付加されている。このタイム・タグの利用方法は後で説明する。

次に、プロダクション・メモリ内の I F-THEN ルールの表現形式を説明す

る。

プロダクション・メモリ内のルールも全体をカッコで囲まれている。そして,

```
(p ルール名
(条件要素 1 )
: (条件要素 n )
--->
(アクション 1 )
: (アクション n))
```

図3-4 ルールの形式(一般型)

最初にIF-THEN ルールつまりプロダクション・ルールを示す「P」の文字がある。次はルールの名前である。この名前は、デバッグ時や、構築したシステムの保守のために用いるものである。ルールの残りの部分は大きく2つの部分に分かれる。矢印の前の部分が「もし~ならば」を示すIF部、つまりルールを使用する前提条件を示す条件部である。また、矢印の左側という意味でレフト・ハンド・サイド(LHS)とも呼ぶ。残りの矢印の後の部分が「~をする」を示すTHEN部、つまりルールが使用された結果、何を行うかを示す実行部である。これも矢印の右側という意味で、ライト・ハンド・サイド(RHS)とも呼ぶ。矢印は、マイナス2つと不等号記号の合計3文字で構成される。

LHSは、さらにいくつかの条件要素と呼ばれる要素から構成される。それぞれの条件要素は、前に説明したワーキング・メモリ要素と非常によく似た型式をしている。この条件要素をワーキング・メモリ要素と比較して、実行するルールを選択するのである。

RHSにも、いくつかのカッコに囲まれた要素がある。これは、ルールが選択された時に実行される一連の作業で、それぞれをアクションと呼ぶ。アクションには、いろいろな作業が記述できるのである。代表的なアクションとして

は、MAKE, MODIFと、REMOVE, WRITEなどがある。

MAKEは、ワーキング・メモリ内に新しいワーキング・メモリ要素を作り出すものである。これは、推論の結果、ワーキング・メモリ内に新しい状況を作り出すことになる。これによって、次に実行する別のルールを選ぶのである。

MODIFとは、既に存在するワーキング・メモリ要素を変更するものである。 また、REMOVEは、既に存在するワーキング・メモリ要素を削除するもので ある。

MAKEやMODIFと、REMOVEを用いてワーキング・メモリを更新しながら推論を進めるのである。つまり、複数のルールの実行の結果作成された新しいワーキング・メモリ要素の組合せで、別のルールが実行されることがある。これによって、複雑に絡みあった推論が行えるのである。

WRITEは、利用者のターミナルにメッセージを出力するものである。

RHSには、以上のようなワーキング・メモリの変更、ターミナルへの出力の他に簡単な計算や、ファイル入出力の指示などを記述することもできる。

IF-THEN ルールの例として、前にワーキング・メモリ要素の例として作

(p 赤い箱を青く塗る (箱 ∧色 赤 ∧重さ ⟨X⟩) ---> (MODIFY ! ∧色 青) (WRITE (CRLF)⟨X⟩"ぬの赤い箱を青く塗りました"))

図 3 - 5 ルールの型式(例)

成した赤い箱を青く塗るルールを示す。このルールの条件要素は、前に示したワーキング・メモリ要素とマッチする(<X>は変数で、どんな値ともマッチする)ので、ルールのIF部が満足され、ルールが実行される。すると、まずTHEN部のMODIFYが実行される。MODIFYの次の '1'は、条件要素の中のどの要素にマッチした要素を変更するかを示している。この場合には、1

番目のもの、つまり最初のものである。このルールの実行によって、箱という要素で色という属性が赤のものがあれば、その部分がすべて青に変更される。 この時、そのワーキング・メモリ要素の他の属性の値は、まったく影響を受けない。従って、今の例では、重さの属性は値10はそのままである。

次のWRITEのアクションでは、今、行った変更をターミナルを通して、利用者に通知する。この< X >は、条件部で重さ10にマッチしているので

「10Kgの赤い箱を青く塗りました」

というメッセージがターミナルに出力される。(CRLF)はターミナル上での 改行を指示するものである。

3.4 **OPS**5の推論の手順

OPS5では、認識一実行サイクルと呼ばれる手順で推論を進めて行く。このサイクルには、照合、競合解決、実行の3つのステップがある。OPS5では、認識一実行サイクルで、知識情報つまりIF-THENルールを順次使用して行くのである。

照合のステップでは、ワーキング・メモリ内のすべての要素と、プロダクション・メモリ内のすべてのルールの条件要素を比較してLHSにあるすべての要素がワーキング・メモリ要素とマッチするルールをリストアップする。ここでリストアップしたルールは実行可能であるという。この実行可能なルールの集合を競合集合と呼ぶ。OPS5では一度には1つのルールしか実行できないのでこの競合集合の中から1つのルールを選択する必要がある。このステップを競合解決と呼ぶ。

この時の選択の方法は2種類用意されており、利用者が自由に選んで使用することができる。それぞれの方法についてくわしい説明はしないが、どちらの方法でも選択の基準は、条件要素とマッチしたワーキング・メモリ要素の新しさと、マッチするために要した比較の回数である。このワーキング・メモリの新しさを調べるために、前に説明したタイム・タグと呼ばれる整数値が用いられるのである。OPS5では、常に最新のワーキング・メモリ要素とマッチし

たルールを優先的に実行する。つまり言葉を変えると深さ優先の探索をしているとも言えるのである。

このような手順で選択されたルールの実行部(RHS)を処理するのが最後の実行のステップである。ここでは、選択されたルールのRHSに指示されたアクションの処理を行うのである。

この結果、ワーキング・メモリに変更が加わると、それによって、また新たなルールが実行可能になり、競合集合に取り込まれる。

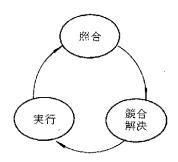


図3-6 認識-実行サイクル

OPS5では、IF-THENルールの条件部(LHS)とワーキング・メモリを比較してルールを実行し、新しい状況をワーキング・メモリ内に作り出し、次のルールを選択して行くので、いわゆるフォワード・チューニング、前向きの推論によって結論を導くのである。

3.5 ま と め

これまで説明したように、OPS5は、非常に高速な前向きの推論エンジンを持った汎用のエキスパート・システム構築用ツールである。

そして、知識情報は、非常に簡単な IF-THEN ルールの形式で表現できる。 また、VAX/VMS の効率的なソフトウェア開発環境で開発を進めることが でき、VMSで提供する他の高級言語などと容易に結合し利用できるので、と ても効率のよいシステムが開発できる。

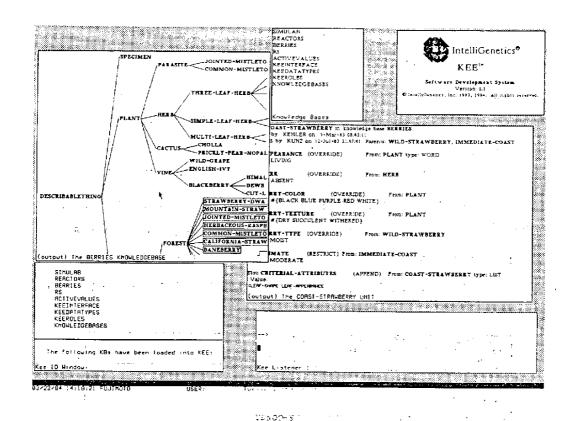
従って、このOPS5は、実用的なシステムとして、エキスパート・システムの構築を考える場合に最適のツールと言える。

第4章 Knowledge Engineering Environment —KEE—

KEEは、米国・インテリコープ社によって開発され、日本における販売をコンピュータサービス(株) および(株) CSK総合研究所が行っているエキスパートシステム開発支援ツールである。

4.1 はじめに

人間の知的機能を代行する能力を持ったコンピュータ・システムとして、人工知能(AI)に対する関心がにわかに高まってきている。なかでも、その応用分野として、専門家の知識を蓄えておいてコンピュータに判断させる、いわゆるエキスパートシステムは、人工知能の具体的な実現として、国内・外において日増しに脚光を浴びてきている。しかしその開発には膨大な費用と時間を要するのが実情である。そこで、効率的なエキスパートシステム開発支援ツールの出現が強く求められていた。今回紹介するKEEは、米国・インテリコープ社の開発による強力かつ効率的なエキスパートシステム開発支援ツールである。KEEは、AIにおいて有用である幾つかのテクノロジーをパッケージ化したものでこれによって、AI技術者、コンピュータ技術者、あるいは特定の専門家達はグラフィック処理能力に富みスムーズな対話形式による計算機環境の下で、これらのAI技術を活用することができる



7.....

図4-1 KEE使用時の画面例

7400 m \$75155

4.2 KEEのアーキテクチャ. Www.domin

AI技術者に対してKEEは次のような機能を提供する。

・ フレームによる記述力に富んだ知識表現。さらに手続的な知識の表現も可能である。

i. Edit dour e

- 関連する知識ユニット間での属性や属性値の継承。
- 多重の階層構造を持った継承システム。
- データ・タイプをユーザが定義できるため、知識ベースが高い完全性を 持てる。
- オブジェクト指向プログラミング。
- 複数の推論エンジン/ルール・システム。

- 複数の知識ベースを用いることによりシステムのモジュラリティが高い。
- スムーズな対話型処理と高度なグラフィック処理機能。

KEEはユニットによって構成される知識ベースによって知識を表現する。知識ベースの構成単位であるユニットは"属性"と、他との関係によってある概念を描き出す。KEEの威力の主たる源泉はこのユニットの操作によってもたらされている。さらにユニットはスロットの集合体として構成されている。各スロットはユニットの属性の値を保持する。スロットには主に3種類のもの、既ち"メソッド""変数"および"記述スロット"がある。メソッド・スロットの値は、"プロシージャー(Lispのコード)"である。プロシージャーはメッセージによって起動される。変数スロットは、あるオブジェクトがメッセ

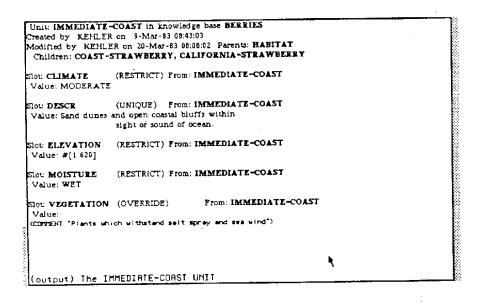


図4-2 ユニットの例

ージに応答する際の効率的な局所的記憶領域を提供する特殊なスロットである。 記述スロットは任意の情報を保持することができる。

4.3 継承(Inheritance)機能による知識管理

知識ベースを構成するユニット間には階層関係が存在する。抽象的概念を表現するユニットが階層の上位に存在し、具体的対象を記述するユニットが下位に存在する。より一般的あるいは、より抽象的概念を表わすユニットは、それより具体的対象を記述するユニットの"親"と呼ばれる。KEEではユニットは親のユニットから層性を継承することが出来る。さらに複数の知識ベース内に複数の親ユニットを持つことさえKEEでは可能である。属性や属性の値を継承することにより、親のユニットは子のユニットに知識を伝えることが出来る。ユニットの各属性は継承の形式を特定することにより、属性の子ユニットへの送り渡し方を制御することが出来る。この継承機能により次のようなメリットが得られる。

• 知識の整合性の保証

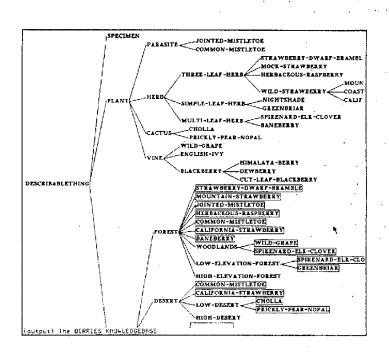


図4-3 知識ベースにおける階層構造の例

- 知識ベースの組織的構成。
- 重複する知識を共有化することによる節約。

4.4 **KEEのルール・**システム

KEEではルール・インタープリタ用知識ベースを用いてルールシステムを実行させることが出来る。このルール・インタープリタ用知識ベースはKEEシステムの一部として提供される。このルール・システムでは次のような形式のポピュラーなプロダクション・システムによってヒューリスティックな知識や"良い推測"や"経験"に基づく知識を表現している。

```
(OVERRIDE)
                                         From: ACCIDENT type: LIST
Slot RULES
Value:
(ORULE 1
       (IF OPRIMARY-COOLING-SYSTEM PRESSURE IS DECREASING) (HICH-PRESSURE-INJECTION-SYSTEM STATUS IS ONO)
       CTHEN CPRIMARY-COOLING-SYSTEM INTEGRITY IS CHALLENGED>>>
       (IF (PRIMARY-COOLING-SYSTEM TEMPERATURE IS INCREASING))
       (THEN (SECONDARY-COOLING-SYSTEM HEAT-TRANSFER IS INADEQUATE)))
 (RULE 3 (IF (STEAM-CEMERATOR LEVEL IS DECREASING)) (THEN (STEAM-CEMERATOR INVENTORY IS INADEQUATE)))
 (RULE 4
       (IF (CONTAINMENT-VESSEL RADIATION IS HIGH) (CONTAINMENT-VESSEL PRESSURE IS HIGH)
       (THEN (CONTRINMENT-VESSEL INTECRITY IS CHALLENCED>>>
 CRULE 5
       (IF (SECONDARY-COOLING-SYSTEM HEAT-TRANSFER IS INADEQUATE) ONOT (FEEDMATER-PUMP FLOM IS LOWD))
       (THEN (ACCIDENT TYPE IS LOSS-OF-FEEDURITER))
       (UNITPUT 'ACCIDENT 'TYPE 'LOSS-OF-FEEDMATER))
       (1F (STEAN-CEMERATOR INVENTORY IS INADEQUATE) (NOT (FEEDMATER-PUMP FLOW IS LOUD))
       (THEN (ACCIDIENT TYPE IS LOSS-OF-FEEDMATER))
       CUNITPUT "ACCIDENT "TYPE "LOSS-OF-FEEDMATER")
 CRULE 7
       (IF OPRIMARY-COOLING-SYSTEM INTEGRITY IS CHALLENGED) (CONTRINIENT-VESSEL INTEGRITY IS CHALLENGED))
       (THEN (ACCIDENT TYPE IS LOSS-OF-COOLANT))
       CUNITPUT 'ACCIDENT 'TYPE 'LOSS-OF-COOLANT))
 CRULE 18
       (IF OPRIMARY-COOLING-SYSTEM INTEGRITY IS CHALLENCED) (STEAM-CENERATOR LEVEL IS INCREASING))
       (THEN (ACCIDENT TYPE IS STEAM-CENERATOR-TUBE-RUPTURE))
       CUNITPUT 'ACCIDENT 'TYPE 'STEAM-CENERATOR-TUBE-RUPTURE))
 OPULE 11
       (IF (STEAM-GENERATOR INVENTORY IS INADEQUATE) (STEAM-GENERATOR STEAM-FLOW IS HIGH))
       (THEN (ACCIDENT TYPE IS STEAM-LINE-BREAK))
       CUNITPUT "ACCIDENT "TYPE "STERM-LINE-RUPTURE>>>
(output) The ACCIDENT UNIT
```

図4-4 原子炉故障診断モデルシステムにおけるルールの例

IF <前提条件>THEN <結論>こういったルールはスロットの集合として蓄積される。プロダクション・システムの一般的使用方法としては"後向きの推論"がある。これは結論部分が正しいと仮定して、その結論が成立するために必要な前提条件の真偽を調べ、要求された前提条件が満たされた時、結論が成立すると判断する仕方である。もう一つの一般的使用方法に"前向きの推論"がある。これは事実の集合から始めて、使えるルールが無くなるまでルールを何度も繰り返して適用して新事実を加えていく方式である。KEEのルール・システムでは両方向の推論が可能である。

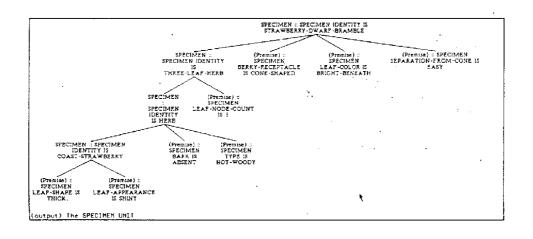


図4-5 植物分類モデルシステムにおける推論過程の表示例

4.5 KEEは結論に関する説明を行う

KEEのルール・システムではHOWとかWHYと呼ばれるメソッドにメッセージを送って起動することにより、ユーザーは推論の過程を確認することが出来る。メソッドによる推論が終了した時にアクセスされたスロットをたどって推論の過程を追跡することが出来る。この適切な語彙の使用と推論過程説明機能により次のようなメリットが得られる。

システムの下した結論が、その領域の専門家に対して明快で検証可能で

ある。

システムの信頼性が高まる。

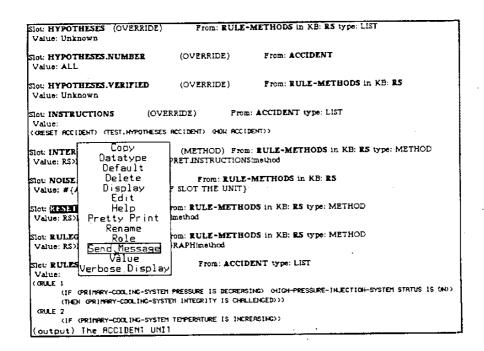


図4-6 メッセージを送ることによりメソッドが起動される

4.6 KFFはメソッドを用いることにより手続的な知識を表現できる

ルールに加えて、メソッドを使用することにより、直接ユニット内にヒューリスティックな知識を記述することが出来る。メソッドの実体はLisp のコードである。ユーザはLisp によって専門領域独自の言語や手続的な知識を記述することが出来る。これらの言語のためのインタープリタやコンバイラはルール・インタープリタと同様にKEEの知識ベースとして提供されている。

4.7 対話型グラフィック・インターフェース

ユーザはマウスと呼ばれるポインティング装置とキーボードによってKEE システムを使用する。ディスプレイ画面はウィンドウと呼ばれる領域に分割し て使用される。それぞれのウィンドウは知識ベースからの様々な情報を表示するために使われる。ウインドウに結び付けられた種々のグラフィック・シンボルを用いて、ウィンドウに対して様々の機能を作用することが出来る。マウスはウィンドウ内のオブジェクトを選択する際にも用いられる。特定のオブジェクトが選択されると、そのオブジェクトに適用可能なコマンドのメニューが、その場所に表示され再びマウスを用いて、コマンドの選択がなされる。

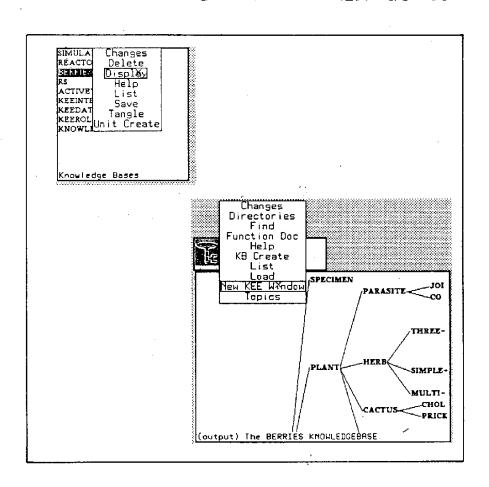


図4-7 メニューによるコマンドの選択の例

4.8 KEEシステム・デバッグ機能

KEEシステムにおける主なデバッグ機能には次のようなものがある。

- エラー・トラップ
- エラー・リカバリー
- メソッド呼出しのトレース
- スロット値へのアクセスおよび代入のトレース
- 推論過程の追跡

エラーに対するトラップやリカバリーの機能はユーザの知識ベース内のメソッドに対しても利用可能である。トレースの機能はメソッドの起動やユニット中のスロットに対するアクセスにも有効である。メソッドに対するトレースは、Lisp環境のトレース機能に接続されている。スロット値のトレースは"active"

```
(RULE 2
      (IF (PRIMARY-COOLING-SYSTEM TEMPERATURE IS INCREASING))
      (THEN (SECONDARY-COOLING-SYSTEM HEAT-TRANSFER)
                                     TS.
                                     INADEQUATE))) applied.
(FEEDHATER-PUMP FLOW IS LOW) is false for the current KB state.
(RULE 5
      (IF (SECONDARY-COOLING-SYSTEM HEAT-TRANSFER
                                   IS
                                   INRDEQUATE)
          (NOT (FEEDHATER-PUMP FLOW IS LOW)))
      (THEN (ACCIDENT TYPE IS LOSS-OF-FEEDWATER))
      (UNITPUT 'ACCIDENT 'TYPE 'LOSS-OF-FEEDWATER)) applied.
Verified: (ACCIDENT TYPE IS LOSS-OF-FEEDWATER)
Is It true that (PRIMARY-COOLING-SYSTEM PRESSURE IS DECREASING)
(HIGH-PRESSURE-INJECTION-SYSTEM STATUS IS ON) is false for the
current KB state.
Mot verified: (RCCIDENT TYPE IS LOSS-DF-COOLANT)
(HIGH-PRESSURE-INJECTION-SYSTEM STATUS IS ON) is false for the
current KB state.
Not verified: (ACCIDENT TYPE IS STEAM-GENERATOR-TUBE-RUPTURE)
(STERM-GENERATOR LEVEL IS DECREASING) is false for the corrent
KB state.
**MORE**
         Kee IO Window
```

図4-8 推論過程の追跡により、ルールのデバッグが可能となる

なスロットによってなされる。このスロットはアクセスされたり代入された時にシステムに反応を示す。この"active"なスロットにより値のトレースの他に、メッセージの送受信やスロットへのアクセスもトレースすることが出来る。

4.9 KEEによるプロトタイプ設計

KEEによって提供されるAI技術を用いることにより、システム設計者は知識表現や推論の方式について選択の余地を与えられる。ルール・インタープリタのようなモジュールはKEEの一部であり、この他にユーザは問題の特質に最も適した知識表現や推論の方式を選択出来る。KEEシステムではユーザ・インターフェースも知識ベースとして実現されている。KEEによって構築されたシステムでは、KEEの持つユーザ・インターフェース機能やオンライン・ヘルプ機能を容易に組み込むことが出来る。従って、これらのプロトタイ

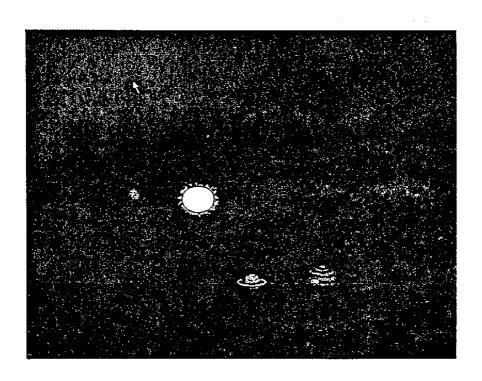


図4-9 KEEによる天体運動シミュレーションモデル

ブ・システムでは均質なインターフェース機能が保証されることとなる。さらに、データ・タイプや継承方式の制御といったシステムの機能をユーザが拡張可能であることもKEEの大きな特徴の一つである。この種の拡張は既存の知識ベースに新しいユニットを加えることにより実現される。このことは、ユーザがそれぞれの固有の問題に最も適した環境を構築することが可能であるという意味に他ならない。

4.10 KEEと知識表現の動向

(1) システムの目標と機能

KEEの設計目標は明らかに実用に耐え得るAI開発環境を早く市場に送り出すことであろう。そのためにIntelli Genetics社が採用した方式は妥当と言えよう。

同社はシステムの基本ハードウェアおよびソフトウェアをOEM(Original Equipment Manufacturing = 他社プランド)またはこれに準ずる方法で購入したが、これによって得られた利点は次の通りである。

- システム全体の開発時間の短縮
- ハードウェア、ネットワーク・アクセス・ソフトウェア、O/S等の基本システム要素を既存のもので済ませたことによる開発コストの低減化
- 新規開発部分を最小限にすることによるシステム全体の信頼性の向上
- ・ システムの移植性を当初から頭に入れての設計であるため、製品ライフの延長を製品に組込むことに成功(例:もし超小型ポータブルWSがパソコン・メーカによって発表された場合、これにKEEを移植して、持ち歩き可能なAI開発環境を作るようなことも比較的低コスト、短期間でできる。)

同社が下した判断として, もう1つ正しいと思われるものは,

Top-Down(トップ・ダウン)開発方式を採用したことである。これにはまずシステム要素のモジュール化を追求したことと、そのモジュール化がモ

ジュール間のダイナミックな動きをサポートするような型で行われたことを 意味する。このような形でシステムを作ると完成後に自由度の大きいパッケ ージになる、製品保守の手続が簡略化される、等の恒久的な利点がある。さ らに作りつつあるシステムが開発途中からデモが可能になるという大きな特 長が現われる。逆の見方をすると、現在のKEEはいまだ開発の可能性を内 に秘めているということも可能であるが、製品としての実用度はほぼ達成し ていると言えよう。

ゼロックス社が最近発表したLoops もKEE 同様のAI開発ツール(環境) であるが、ともに「AI型グラフィックス」とも呼ばれるべきグラフィック ス・インターフェースを用いている。AIでは特に複雑になりやすいユーザ ・インターフェースを整理するのに効果が大きい。もし現在通常のコンピュ ータに用いられているキーボードとスクリーンによる会話型入出力に頼らね 「はならない場合,AI特有のこみ入った入出力を処理するのはほとんど不可 - 能に近い (生産性の低下が著しい) 。さらにユーザ・フレンドリネス (使い やすさ)を追求した場合,入出力インターフェース自体がAI化を余儀なく されよう。音声入出力や視覚入力処理など「第5」で問われている入出力形 態はこれを意味する。これに比べると現在のKEEのインターフェースにお ける使いやすさは「中間型ユーザ・フレンドリネス」とでも呼ぶべき形態で、 ゼロックス1100シリーズ,シンボリクス3600機,VAX-11シリーズ 用のVAXワーク・ステーションのすべてに共通している。但しKEEの場 合はシステムの移植性を強調しているので、上記コンピュータのどれとも一 - 歩離れた形でグラフィックスを設計している。重要なことは基本となるホス トのどれもがすでにグラフィクス入出力に必要な諸機能を有していることで、 これによって開発が簡略化されることを目指している。

(2) システムの外観

以上ですでに解れたように、KEEは独立のワーク・ステーションをネットワークで結んだ形が現在の標準形である。机上に置いて使う(デスク・トップ)高性能グラフィックス端末と、床上設置型のシステム本体にわかれて

いて、両者はフレキンブルなケーブルで結ばれている。ビット・マップ方式の端末からは3ボタンのマウスが入力装置として出ている。システム本体はイーサネット・コネクタを有し、他のKEEあるいはKEE以外のワーク・ステーションへのLANを介しての接続を可能にしている。ワーク・ステーション型とは別にTSS方式の多ユーザ・システム版も開発中であるが、やはりKEEの本命がネットワーク連結形の独立ワーク・ステーション群という、きわめてアメリカ的形態であることは間違いない。最近OAに関連して日本でもこのイモブル形式が理解され始めたが日本国内では大中型ホストからタコ足式に「ワーク・ステーション」である端末が延びる形がまだ受けいれやすいと言えようか。ネットワーク連結ワーク・ステーション方式の場合、これから拡大的にLANの持つ種々の利点が効果を発揮しようが、現在の課題はワーク・ステーション当りのコストである。

基本となるゼロックス社のD・マシンの上にKEEを積むとやや高価なワーク・ステーションになる。にも拘らずWS+ネットワーク型にすることの意味は設置の容易さ、運用上の自由度、処理能力、信頼性などの面で望ましく、まとまりの良いシステムを作ることにある。すでに述べた「AI型グラフィックス」を顔として持つために、システムのデモに際しての効果は大きい。これは市場にあるこの種AIマシンのすべてに共通した長所であるが、潜在的ユーザにAI製品をアピールするのに有効であろう。グラフィックス端末の画面に現われるLISP表現はスタンフォード系のインターリスプ(Inter LISP)のものである。KEEをLISPの他の方言に組み換える場合にはいくらかの変換手続きを経る必要がある。システム外観とは直接関係ないが、そのようなLISPの有力候補としては、MIT系のMAC一LISP、DEC社が推している Commn LISP、カリフォルニア大バークレー校のFran ZLISPとその68000版、マサチューセッツ大のCLISP等々がある。

- (3) システムの特徴
 - a) パッケージング

KEEはAIツールとしては初めて本格的なパッケージ化に成功した製 品である。現在、他にゼロックス社のLOOPSが知られているが、これは AIアプリケーションを目指した製品というより「AI練習台」の色彩が 濃い教育用である。パッケージの成功は、OEM的アプローチに徹底した ことが大きく貢献しているように思われる。基本として他社によりすでに 開発され、十分テストされたハードウェアとシステム・ソフトウェアを使 用したということ(DEC社VAX,ゼロックス社Dマシン=1100シリ ーズ,サン・マイクロWS)。共通性の高いAI言語(これもすでに十分 テストされたもの)を事実上のマシン・インターフェースに選んだこと。 ビットマップ、グラフィックスとマウス入力の基本機構(ハード、ソフト 共)を持つマシンを使用する方針をとったこと。またこれらのマシンがど れでも標準化の済んだネットワークを使える体制にあること等基本マシン の物理的形態こそ異なっているが、主要な機構上の共通因子は注意深く選 択されているのがわかる。このようにすると,システムの設計と製作は大 筋としては、これらの基本マシンの上に積み上げるアプリケーション向け AIソフトウェアについて集中すれば良いことになる。実際には各種基本 マンンとのすり合わせに若干の注意を払わねばならないが、それでも論理 的方法論が明解に保たれることには変わりない。このようにしてシステム の階層構造の内の下位各層はすでに他で製品化され十分実用化の進んだも のを用いたので、十分のエネルギーをユーザ・インターフェースの充実と、 AIソフトウェアの性能向上に費やすことが出来たと言えよう。

結果としてユーザ・インターフェースの生産性は極めて高く、また今後ネットワーク・インターフェースを本格的に使うようになった場合の利益が感ぜられる。

b) ユーザ・インターフェースを「中間型ユーザ・フレンドリネス」のグラフィックス端末としたことによって生産性がいかに向上するのか?

ユーザがAIで言うアセンブラ・プログラミング(LISP言語)に直接触れないで済むことが第1の理由である。AI用のアセンブラ言語がこの

場合のように特にLISPである場合、プログラムのステートメントの構造をつかむことは容易でない。特に函数表現(LISPは全プログラムが函数のみから成っている)に用いられるカッコ=「(・・・)」の数を左右バランスさせるだけでも相当な努力を要する。これがFORTRANでいう文法上のエラーの問題に留まらず、函数表現の意味、つまりプログラムの意図を大幅に変えることに簡単に発展してしまうから始末が悪い。これに加えてAI用のデータ構造、函数構造は極めて複雑なものが多い。グラフィックを用いてこれを図形で整理したり、表の形にしたりすることがプログラムの生産過程に貢献することは明らかである。グラフィックスの面では常に改良の努力がなされて生のLISPプログラミングを押しつけられるよりはるかにスッキリした形になっている。

4.11 知識表現

現在のAIの理論研究の対象として王者の位置を占めるのが知識表現、あるいは、Knowledge Representation(KR)であることは疑問の余地がない。知識工学(Knowledge Engineering = KE)とも呼ばれる今日のAIにおいて料理すべき知識をいかに設定し、料理後の知識をどういう風に供するかの問題は特に重要であるばかりでなく、考えられる限りの将来においても知識を使い易くかつ効率的に表わすことの必要性は重さを失われてないであろう。この「KR重要」の認識は日本では特に欠けており、最近ではAAAI-83における全米人工知能学会学会長の基調演説における要点のひとつとして、「第5世代いまだ心配に価せず」のような形で話題にされる。KRの研究を含まないAIプロジェクトは表面的なアプリケーション強調型か、底面的なアセンブララ重要視型となり、肝心の中核を成すべき知識工学方法論の欠除した異様な形がいと化す。KEEの最大の貢献はAIプロジェクト担当者に「常に注意を怠ってはいけないのはKRである」ことを指摘していることである。意図としてはKEEを使ってAIシステム(ES=エキスパートシステムであれ、NL=自然言語であれ)を構築しようとするものがまず設計の中心課題をKRに関す

る問題に設定せざるを得ないようにしていることである。

ICOTも含めて、日本のAI研究の方向に米国の研究者が批判するとき、特にこの点を突かれることが多いように思われる。米国における定説はKRの手段としての記号論理学(Logic)には限度がある、ということである。一昨年に比べて昨年のAAAI学会では米国内でも論理プログラミングを見直そう、という動きがかなり出て来たが、主力はやはり他のより柔軟な知識表現を採用している。いろいろ理論的な比較が第一線研究者によって成されていると同時に、ファイゲンバウム、ミンスキー等の米国の東西両巨頭がしばしば日本の「固いアプローチ」を批判する発言を行っている。要旨は「そんなコチコチの知識表現法にたよっていては、高度のAIシステムの製作はおぼつかない」ということである。

そういうアメリカ自身もKRに関して問題がないわけでなく、AIをState of the Art (最先端の研究)とState of the Business (最新応用技術)に分けて見るときそれが明確になる。なおこのState of the Art/State of the Businessの区分による分析方法はAIのように大学と企業・政府研究機関の開発努力が入り組んでいる分野では極めて有効である。例えばTekn-wledge社は、State of the Business に関する業務を行うのに対してスタンフォード大学はState of the Artを追求する。というように、まずState of the Buisness に関して言えばアメリカ、特に西海岸はIFTHENのルール・ベース方式のシステムが圧倒的に多く、より高度のKR手法であるフレームや意味ネットワーク(Semantic Network)あるいはオブジェクト指向プログラミング(Object Oriented Programming)などを採用するシステムが現われ出すのにかなりの時間がかかったというより今年になってその先兵が現われ始めた。

一方 State of the Art に関しては、現在でもルール・ベースを基調にした研究が大学で行なわれたりして、MITのミンスキーをして「ルール・ベース方式のKRの基礎はDENDRAL 研究者などによって15年前詳細部分にまで理論解析が済んでいる。今から研究室においてルール・ベースによるシステ

ムの構築や理論研究を始めようとするような研究者は恥を知れ」と言わしめる 程である。

一方同じMITのR・デービス(現在若手ES研究者間の第一人者)などは、「ESを作るなら初めはルール・ベースから始めなさい」と現在でも勧めている。

他方ゼロックスPARCのS・ブラウンなどの「ルール・ベース」の反省なども知られ始めている。さらに「論理プログラミングの見直し」が最近、とくに今年に入ってからの顕著な傾向のひとつでもある。これは日本の「第5世代」という「将軍ブーム」のために生じた小さい論理学ブームによって、少しずつながら発言の機会を与えられるようになった、というのが現状であろう。

1983年全米人工知能学会大会中の特別討論「AI研究方法論」に於けるエ ール大のR・シャンクとスタンフォード大J・マッカーシーの対立は「ロジッ - ク」の攻防に終始したし、パネルディスカッション「論理プログラミング」が 特に設けられ、R・ジェネゼレス(スタンフォード大)D・ワーレン(SRI) H・シュローベ (MIT)D・マクデーモット (エール大)などが極めて興味 深い議論を展開した。これとは別にブリティシュ・コロンビア大のRライター SRIのRモーアなどはIJCAIなどのヨーロッパを中心にした国際的な発表 機関を通じより掘り下げた論理プログラミング支持の理論展開を行っている。 このような一見混戦状況のKR研究においてKEEの占める位置はどうなるの か。この問いはKR界において今日何が起りつつあるかを知らねば答えが得ら れない。結論から言えばKEEは正しい方向を向いたシステムであると言うこ とが出来るがこれにはいくつかの但し書きが付く。ひとつ重要な傾向は、HY-BRID KR (複合知識表現)である。これはこの1年以内に表面化して来た テーマであるが、論文としてはフェアチャイルドAI研のR・ブラックマン (BBN出身)とH・レベスク(トロント大出身)共同提案の「KRYPTON」 とスタンフォード大のM・ジェネゼレス (M.R. Genesereth)の「メタ・レベ ル・アーキテクチュア」がすでに無視出来ない重さを感じさせる。一体何かと 言えば多種多様なKRの共存を認めてこれらを適者優先で複合的に使おうとい

うことである。つまりルール・ベース使用で充分な分野,例えば組織体のシミュレーションにおける服務規程などにはこれらを積極的に導入し,他方,より高度な知識表現を要する部分には意味ネットワーク(Semantic Network=SN)のような概念表現の豊かなものを入れる。さらに比較的単純ながら多量の論理表現を高速で処理する必要のある場合には「論理プログラミング」による表現と処理手続きを利用しようという考え方である。どのKRが優れているかという理論にあけ暮れしているより各方式の良い面を採用して複合知識代表方式とし、これでもってより高度な応用に対しようというものである。すでにこれらをエキスパート・システム,自然言語理解,視覚処理などに応用して極めて高度の応用システムを作ろうという動きが出て来ているが,いまだに明確にとらえるところまで行っていない。KEEはこの複合KRの理論を認識し、かつこれを今後の研究家(State of the Art)とシステム開発者(State of the Business)のためにパッケージして提供しようとした点で画期的である。

もうひとつのKR界の重要な動きは、動的KR(Dynamic KR)であるが、これは、ここでは直接関係ないので説明を省略する。これはさらに高度なKR研究テーマとも言え、将来知識獲得や学習が具体化してきた時点で必要度を増すKR手法である。現在では若干の実験(例:フランス国立オートメーション及び情報科学研究所=LAASのヒラーレ・ロボット、近日中に日本と共同で移動ロボット開発計画を発足させる予定)と理論的研究(例:最近Artificial Intelligene 誌=North Holland Pubishing社の「Search=検索特集号」に見られるような、AIの古典的分野である「検索」技術の再開発)が出始めた程度である。

上記ヒラーレ・ロボットはLAASのG・ジラールによるもので、移動しながら他の感覚機能からの入力を使って自分の周囲の「世界」に関する知識を逐時動的に構築かつ修正し、この知識から得られる地図を頼りに自己の行動を決するという高度な動作をする。

さて複合KRに戻って商品化第1号とも言えるKEEがどの程度の内容を持

っているかを検討したい。まず複合KRに至った過程が分子生物学(Molecular Biology)という特殊な研究分野における研究活動を深くかつ活発に行ない、KRとは何かという問いを充分追求したことを意味している。KEE内における役割分担は、若干の推測を入れて説明すると以下のようになる:

- ① まずルール・ベースによる知識ベース(KB)が全体の表現分担に対する 知識を持つ。また生物分子間の反応に大きく貢献する酵素の作用などもルール・ベースKRで記述しておく。
- ② 次にDNAの内部構造、トポロシー(相互間構造)、および実験条件をフレーム表現で作る。例えばDNAの4主要エレメントが、それぞれフレームとなり、かつ中心フレームである「DNA」との間に「ISA」関係と「K-ind-of」関係等で結合させ、一種の意味ネットワークを構成する。
- ③ さらにルール・ベース記述で上記実験条件のフレーム表現を読みその解釈 に従って実験の施行を制御する。

この3層のKR集団がいくつかの知識ベースに分散して存在する。とくに②で定義した実験条件フレームを③の実験エンジンが処理するという構図はあざやかである。もしこれが生産工場で、エミュレーションで多種多様の薬品をFM的に多種小量生産をする場合には③の部分が記号論理表現をPROLOGマシンなどで高速実行するように、複合KRの発展が起るようなことも、例として考えられる。そのようなものがICOTの考えている通りに実現するとして、近年中にParallel Prolog Machineが作られた場合には、制御部をPPMで超高速化する応用も当然考えられ、KEEが支える複合KRの利点は明らかである。但し現在のKEEでは「論理プログラミング」はいまだ開発段階であり、またこれを可能にしてもその部分の機能が効率良く行なえるという保障はない。むしろすでにKEE用の物理的ホストと成るゼロックスのDーマシンにすでに存在するネットワーク(LAN)を使って、「近所」に存在するPPMとネットワークを介して「知識」や「結論」「仮説」の投げ合いを行い、連継プレーで処理するようなことが適していると考えられる。

このようにネットワークを介したりして複合KR間で推論などの論理処理を

行うものをマクロ複合KR(Macroscopic Hybrid Knowledge Representation)、一方ひとつの論理処理環境の中で複合KRを作り処理するものをミクロ複合KR(Microscopic Hybrid Knowledge Representation)と仮称する。これによれば上記フェアチャイルドAI研のKRYPTONはミクロ複合KRの典型的な例である。KEEは現在のままですべてが一台のマシン内で行われるという意味でミクロ複合KRである。但し高性能のLANを内蔵し、かつこれとKEE論理処理部とのインターフェースも十分信頼に足りるものが出てくると予想されるので、マクロ複合KR面の発展も大いに期待出来る。

4.12 システムの入出力方式:

通常このシステムでは入出力インターフェースとして、オペレータ・インターフェースとプログラム・インターフェースがある。前者はユーザが直接コンソール(制御端末)やその他の入出力装置に面して作業を行うためのものであり、後者は、ユーザの用意したアプリケーション・ソフトウェアがシステムと直接やりとりする際の窓口である。KEEについてはほとんどの入出力が端末を通してのオペレータ・インターフェースによるものである。後者については、フレームの基本単位である「UNIT」の指定の際に処理方式を指定した「手続き」を指定することが出来るようになっている。

但しこれはシステム全体がオブジェクト(例えば抽象・具象を問わずに、データ構造など)を中心にソフトウェアを構築する、いわゆるオブジェクト中心主義(Object-Oriented)のプログラミング方式であるためかも知れない。この場合にはシステムが手続部分に相当するものの大部分をあらかじめ持っていて、ユーザはいわゆるデータ部分のみを作成する形になる。KEEはそのような方式のようである。従ってKEEを使ってのアプリケーション開発作業のほとんどはグラフィック端末のスクリーン上にマウスとキーボードを操作することによって行われる。作業内容は主に知識ベースを構成する要素(ルール、フレーム、オブジェクト)の指定・詳細記述である。

この種システムにおいては、今後キーボード作業の簡略化(将来的には省略)

とそれに応じてのマウス使用の重視,タッチ・スクリーン方式への移行,音声入力方式への移行などが試みられつつあるが,KEEもその方向をたどっているように思われる。この種システムの代表的なものとしてはアップル社の「リサ」およびTI社が発表した管理職用のパソコンなどがある。但しこれらのユーザ向けシステムとKEEのような開発ツールの異なる点は,インプットする情報の種類と量である。例えば多数のフレームのスロットを指定してゆく場合マウスのみでは無理であろうし,音声入力に頼っていては確度のみならず速度まで不十分である。KEEは,システムの基本設計上このような新入出力機器の進歩に伴って生じる操作モードの変化に比較的容易に対処できる設計になっていると言えよう。このような意味で,現在のKEEにカラー端末を接続することは、基本となるDーマシンがカラー・ディスプレーの導入を始めたこともあって、時間の問題であろう。またユーザもこれを求めることが充分考えられる。

KEEは現在すでに、イーサネットとゼロックス社のXNSと呼ばれる上位プロトコルを内蔵している。このことは将来イーサネットを物理媒体として、かなり高度なネットワーク利用の作業形態を構成出来ることを意味している。上位プロトコルの重要度は日本では最近やっと認識され始めたが、欧米ではパッケージ製品、それもKEEのような特殊な応用分野を目指すシステムの設計段階から考慮する傾向が確立している。

他のワークステーションと接続し、多くのワークステーション群から成る複合システムを有機的に運用することによって得られる利益を計算に入れての「INS型設計」である。事実、KEEに他のワーク・ステーションを数台接続した時点でAIを含んだINSプロトタイプが見えてくると言えよう。この場合各ノードの特殊性の吸収のためと、システム全体の応用のテーマを記述するための上位プロトコル層の重要性は大きい。

予想としてはゼロックス社の「OSI採用宣言」を同社がいかにフォロー・ アップするかにかかっているように思われる。しかしこれは時代の必須である から、やがてXNSの新版が出現しゼロックスばかりでなく、同社のネットワ ーク構想を採用したOEM各社(Intelli Genetics社を含む)が右に習えで改宗することがはっきり予言できる。良かれ悪しかれKEE内ではXNSの本格的利用は未だ先のことであるように思われるので、「改宗」に際しての混乱は余りないことが期待できる。逆に言えばKEE内にLANを持込んだことによって始まった「INS化作業」の大部分は今後の同システムの開発者、販売担当者、ユーザの肩にかかって来る。

さてこのようなネットワークの「基本方式問題」に結着をつけたあとのKE Eを含んだ系の発展はどうであろうか?いわゆるリソースシェアリング(資源 共用)型のシステム構成から考えると「XX サーバ」の思想の発展した形態として「知識ベース(KB) サーバ」のようなものが出現するのは時間の問題であろう。KEE が複合KR 方式(Hybrid Knowledge Representation)をこの段階ですでに採用したこともその傾向を強める結果になる。

例えばKEEを基礎にした窓口サービス・システムを考えると、高度に完成された複合KR方式のKBをネットワーク上に設置し、各窓口には目・鼻・口に相当する部分のみを置く、というような経済設計を考えない方が不思議である。

経済性ばかりでなく,運用形態から見ても(この例の如く),このような形が好ましい場合はいくらでもあろう。ちなみに,現在の段階ですでにKEEを開発システム(Development System)としてでなく,目標システム(Target System)あるいは運用システム(Operational System)として使用することは十分に可能である。

リソースシェアリングはこの他にもいろいろな形が考えられる。

例えば目・鼻・口をひとつにしておいて多数の頭脳をネットワーク上に配置するような場合も考えられる。この場合、共用される「目・鼻・口」ルードがリソースになるが、それぞれの専門分野(domain)を代表した「KD+推論機構」の頭脳がこれをシェアすることによって高度の問合せ応答システムを構成することが可能であろう。

ほぼ全世界が「OS I 忠誠」を宣言し終わった現在, ネットワーク間の相互

乗入れ各社機器の同ネットワーク上での混合使用などが活発化しよう。こうして誕生する分散系では必ずしもすべてのノード(節)がAI型のワーク・システムであることは限らない。むしろ従来型のワーク・ステーションなどと混用することによって可能になる有機的効果を期待してシステム設計を重点的に行うことが、INS思想の実現に近づけることになる。例としては、知的OAネットワークがある。KEEシステムの頭脳としてネットワーク上に常駐させ、これに入出力機構を完備した「窓口ステーション」、ワープロを部下に持つ「文書作成ステーション」、新旧の情報の出し入れを管理する「DBアクセス・ステーション」、作業のコスト計算等と収支の管理をする「会計ステーション」等々を組合わせて「高度OAシステム」を設計することが考えられる。(このようなシステムを分散AIとも呼ぶ)。

こうして述べると、入出力関係ではネットワークの占める位置がいかに重要であるかがわかる。主に目標システムとしてKEEを見た場合を規定したが、現在のKEEの存在形態である開発システムとして考えた場合にも、KEE本体に複数のグラフィックステーションを接続する可能性など無視出来ないインパクトを持つと言えよう。

4.13 システムの完成度

現在 Intelli Genetics 社が力を入れているKEEの実現形態はゼロックス社の1100シリーズワーク・ステーション、いわゆるDーマシン・シリーズ上に組んだものである。1100シリーズはこれまで三機種が発表されているが、いずれも頭文字が「D」で始まるニックネームが付けられていることから、Dーマンンと呼ばれている。これは、小さいサイズのものから順番に並べると、「Dandelion = 1108 = 9ンボボ」、「Dolphin = 1100(1116) = 10、「Dorado = 1132 = 10、この内 Dolphin 版はほぼ開発システムとして完成段階にあるようだが、他の二版に関しては、新しいこと(Dorado)と、型式上若干のつじつま合わせ(Dandelion = 旧ゼロックス・Star 機)が行われている。

日本の現状では高級ワーク・ステーションとも言えるこの種システムに対するユーザの理解が未だ浅い。AI開発とはそのように金のかかるものであると割切ってしまえば良いが、むしろ、ユーザのAIワークステーションに対する理解を深める努力が必要であろう。

これらの開発システムとしてKEEを見る場合の完成度に対し、目標(Target)システムとしてのKEEの完成度はどうであろうか?比較的簡単に解決出来るハードウェアのパッケージの変更・改良の作業の他に、システムの入出力機能を向上させる一連の作業が進行しているように思われる。この場合前節(システム入出力)で述べたように、ネットワーク関係の充実が重要である。こうすることにより、システムの移植性が保たれ(LANはホストと離れて存在するから、ホストを替える度にホストのソフト/ハードをいじる必要がない)目標システムがアプリケーションによって附加せねばならない特殊 I/O機器やソフトウェアの導入を、局所的に処理することが出来る。

4.14 まとめ

現段階における Intelli Genetics 社のKEEシステムの技術的位置づけ に関し、以下のように結論することが出来る。

- (1) KEEシステムはゼロックス社のLOOPSと共に初めての本格的AI開発 ツール(環境)である。LOOPSがAI学習用であるのに対し、KEEはよ りアプリケーション開発に適している。
- (2) KEEをアプリケーションの目標(Target)システムとして使用することも基本的に可能である。
- (3) **KEEがOEM**方式で作られることの利益は大きい。その内でも**AI**システム開発コストの低減化、期間の短縮化と完成したシステムの移植性の確立・保持は特筆に価する。
- (4) 製品の完成度は全体的に極めて高い。
- (5) LANを持つベース・システムを採用しているので、今後の有機的使用, 特にINS的使用に期待がもてる。

- (6) 現時点でこのツールを使って具体的なAIシステムの開発を始めることは 容易に可能である。また、そのような具体的アプローチでシステムに習熟することが望ましい。
- (7) KEEの複合KR方式はユーザに洗練されたKR技術をやさしく提供する。 ここに複合KR(複合知識表現)とは、多種多様なKRの共存を認みて、これれらを適者優先で複合的に使おうということである。つまり、どのKRが優れているかという理論にあけ暮れしているより、各方式の良い面を採用して複合知識代表方式とし、これでもって、より高度な応用に対しようというものである。

KEEは、この複合KRの理論を認識し、かつこれをパッケージングして提供しようとした点で画期的である。

4.15 適用機種

Symbolics 3600 (Symbolics社) XEROX 1100 SIP,1108 (XEROX社)

第5章 知識ベースシステム構築用ソフト「BRAINS」

BRAINSは、(株) 東洋情報システムが開発した知識ベース開発ソールである。 以下、本ツールの開発の背景、概要、特長、構成と機能、知識表現、適用分野、 今後の方向について解説する。

5.1 開発の背景

米国Stauford大学のE.A.Feigenbauwによって、AIの実用研究としての「知識工学」が提唱されて以来、知識ベースシステムは米国を中心として故障診断、油井の開発、データ解析、各種コンサルテーションシステム等、技術分野から事務分野に亘って広い範囲で実用化がはじまっている。(図5-1、表5-1)我が国でも(財)新世代コンピュータ技術開発機構(ICOT)をはじめ知識ベースシステムに関する研究発表は非常に盛んで、一部では実用システムの開発も始まっている。しかし、知識ベースシステムに対するニーズがあり、高い関心を持ちながら、実際の開発までは踏み切れないでいるユーザも多い。これはまだ我が国の知識工学の全体状況が米国と比較して数年程度遅れており、技術者の不足や開発環境が整備されていないことに起因している。とりわけ、知識ベースシステム構築の中核となる推論ソフトウェアが既存の開発環境の下で手軽に利用できないでいることの影響が大きい。

表 5-1 AIマーケット(単位:百万ドル)
(DM Data 社 AITRENDS'84より)

MARKET AREA	1981	1982	1983	1984	1985	1986	1987	1988	1989	1990
Expert System	4	9	18	35	67	126	231	408	678	1017
Natural-Languuge Software	5	8	18	34	64	117	208	357	574	832
Computer-Aided Instruction	3	5	8	14	24	40	66	103	150	195
Visua/Recognition	10	22	50	90	133	210	322	472	642	770
Voice/Recognition	5	7	11	18	35	70	115	180	263	342
TOTALS	27	51	105	191	323	563	942	1520	2307	3156

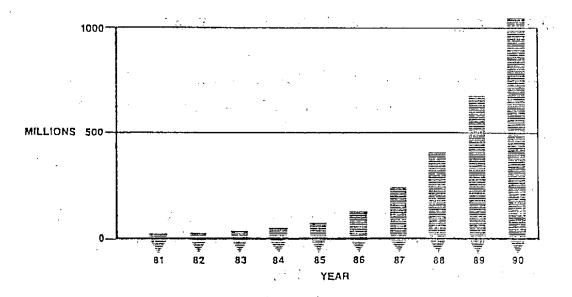


図5-1 エキスパート・システム・ソフトウェアマーケット (DM Data 社 AI TRENDS '84 より)

ところが、近年東京大学で従来の汎用機上で作動する優秀なLisp Interpreter(UT Lisp)が開発されたことや、我が国でもようやくAI専用マシンが普及もはじめたことが相俣って、実用的な構築支援ツール(推論ソフトウェアを含む)開発のソフト/ハードウェア環境が整ってきた。こうした中で、既に具体的な適用分野を決め、知識ペースシステムを開発しようとしているユーザ、あるいは将来開発しようとしているユーザを対象に、汎用性と実用機能に力点をおいた知識ベース構築支援ツール「BRAINS」を設計、開発した。BRAINSは知識ベース開発ツールとして、我が国で開発された最初の商用パッケージである。1984年3月に第1版を発行、85年2月現在で、24社28システムがユーザに導入され、各種知識ベースの開発に使用されている。BRAINSは当初UT Lispで開発され、その後Frautz Lisp版とZeta Lisp版が追加された。表5-2に適用機種を示す。

表5-2 BRAINSの適用機種

IBM308Xシリーズ MVS/TSO 303Xシリーズ 4 3 X X シリーズ 370シリーズ FACOM Mシリーズ OSV/F4 S3000シリーズ HITAC Mシリーズ V.O.S.II Symbolies . 3670 3 6 4 0 V A X11シリーズ UNIX VMS

5.2 BRAINS概要

BRAINSはその基本構造として米国Rutgers大学で開発された「EXPE-RT」の枠組を採用し、Stanford大学の「MYCIN」の機能を取り入れている。これは従来より数多くの知識表現方式が提唱され、インプリメントされてきた中で、上記システムが多方面に適用され、実用性が実証されてきたことによる。また、実際に「EXPERT」を使用して、プラントの異常診断システムを試作した経験に基づいて実用機能を強化した。現在では、導入ユーザ数社と共同で各種知識ベースシステムを構築中であるが、この経験をもとに更に一層の機能性向上を目指して、バージョンアップを続けており、各種故障診断、データ解析から機器、部品や工法の選定、法律のコンサルテーションシステムの構築まで適用領域が広がっている。

さて、BRAINSは専門家のノウハウの知識ベース化を支援する知識ベース 作成支援サブシステムと、知識ベースに基づいて推論を実行する推論実行サブ システムから構成される。概念を図5-2に示す。

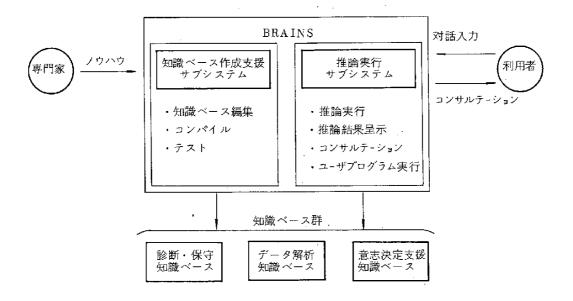


図 5-2 BRAINS 概念図

5.2 BRAINSの特長

BRAINSは次の特長をもっている。

- ① 知識ベースの表現形式としてプロダクションルールを採用しているため, 知識の表現が簡単で汎用性に富む。ルール間の相互作用はデータを介しての み行われるので、ルールのモジュール性が高く,知識の追加,修正,削除が 容易である。このため、システムのプロトタイピングや拡張性に優れている。
- ② 日本語による知識ベース作成機能,知識ベースを日本語でも記述できる。 日本語による入出力機能と合わせて総合的に日本語をサポートする。例えば、 出力機能の一つとして日本語の文章作成機能があり、該当した IF—THEN形式のルールを翻訳して、日本語を生成し、推論結果をわかりやすく説明す ることができる。(図5-3)
- ③ 知識のユニット化による大規模知識ペースの構築が可能 プロトタイプシステムの知識ベースは、一般にルール数が数百のオーダで

ある。しかし、実用レベルの知識ベースでは数千のオーダであることが多い。こうした大規模な知識ベースを平面的にルールで記述すると、知識ベースが読みにくくなり、推論の実行速度も低下する。この問題を解決するため、B-RAINSではルールをグループ化してユニット構造とし、ユニットの適用条件を設定することにより大規模知識ベースの取扱いを可能にしている。(図 5-4)

④ 既存のシステムと接続でき、全体システムの一部として運用できる。

知識ベースンステムの適用方法の一つとして、既存のシステムで従来の手法では表現の点でネックとなっている部分に適用し、システム化の範囲を拡大することがある。例えば、CADシステムにおける部品の選択やプロセス制御の運転監視等は、知識ベースシステムが単独で動くのではなく、既存のシステムと連携をとりながら実行することにより、全体システムとしてより効果の高いシステムが実現できる。また、単にルールに基づいて結論を出すだけでなく、推論実行時に他のプログラムと接続することができれば、ユーザのニーズにより適合した知識ベースシステムを実現することができよう。BRAINSはこうしたユーザプログラムをそれぞれ1つの処理(process)として、知識ベース上に定義することによりユーザプログラムを自由に呼び出し、実行することができる。ユーザプログラムはLisp はもちろんのこと、FORTRANやCOBOLで書かれていてもよい。

⑤ 知識ベース管理機能の強化

知識ベースが大規模化し、複数の専門家によって作成されたり、不断の改訂が図られる状況では、ルール相互間の予盾や包含性のチェックが不可欠の要素となってくる。BRAINSでは知識ベースのコンパイル時と推論実行時にこれらのチェックを行う。それとともに、知識ベースを自動的に樹状構造に展開してグラフイカルに表示することにより、知識ベースの構造を容易に把握できるようにしている。

** FF RULE

FF1-5 : F(加圧圧力,0:69)
-> F(加低圧力警報,T)
もし、加圧器の圧力 (KG/CM2)が0(KG/CM2)以上69(KG/CM2)以下であるならば、低圧力(加圧器) 警報装置が発信する

図5-3 ルールの自動日本語文章説明例

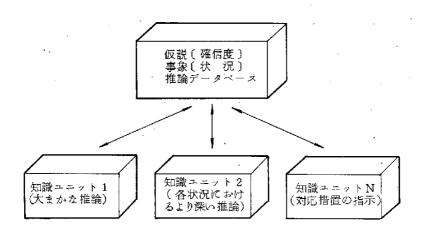


図5-4 知識ユニットの適用

5.4 構成と機能

BRAINSは推論実行機能と知識ペース管理機能からなるが、ソフトウェア構成もこれに対応している。図5-5と図5-6に機能構成と機能関連を示す。

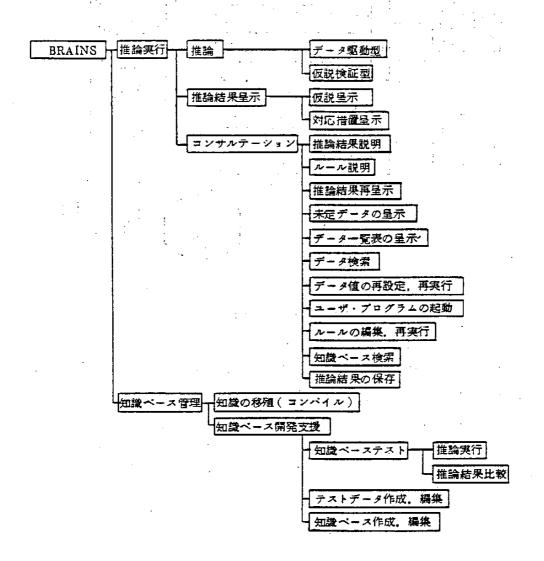


図5-5 ソフトウェア構成

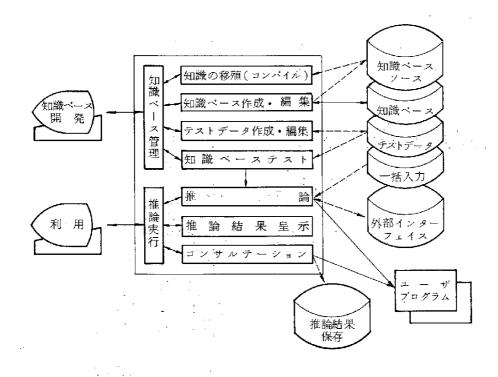


図5-6 機能関連図

図5-2で、BRAINSは推論の実行方法として、入力データに基づいてどのような仮説が成立しているかどうかを推論する前向き推論(データ駆動型推論)と、指定された仮説が成立しているかどうかを検証する後向を推論(仮説検証型推論)の二つの方法を自由に選択することができる。この他に、推論の実行途中で、中間目標が達成されたならばその時点で出力したり、ユーザの指示や追加データを受け入れるために一時停止する機能を付加している。(V2.0版)

コンサルテーション機能では、何故そのような推論を行ったのかを説明する推論結果の説明機能や、関係したルールを呈示、説明し、その場でルールを修正したり、データ値を再設定することにより、推論を再実行する機能がある。この他に、未定データの呈示や推論実行時に適用された矛盾したルールまたはデータの呈示機能など多様な機能がある。図5-7は推論結果の説明機能の一

つとして,推論の道筋を樹状構造に展開し,グラフ表示した例である。

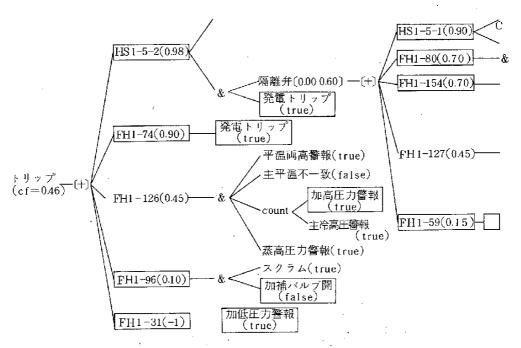


図5-7 知識ベースのトリー構造グラフ表示例(発電機トリップが 事故原因であることを決定するツリー。枠で囲まれたもの は条件を満足しなかったことを表す。)

5.5 知識表現

BRAINSの知識表現は

- ① 推論の前提となる事象や仮説の種類,内容を定義するデータ定義
- ② 推論実行中に呼び出すユーザープログラム等の処理を記述する処理定義
- ③ 事象と仮説, 処理間の関係を定義する知識ベース定義から構成されている。(表5-3)

この中で、知識ベースを構成するルールは IF-THEN 型のプロダクション 形式で表現され

- ① 事象と事象とを結びつけるFFルール
- 事象と仮説とを結びつけるFHルール (2)
- 仮説と仮説、仮説と対応指置を結びつけるHHルール (3)
- ④ 事象,仮説と処理を結びつけるFIルール の四つのタイプがある。表5-4と図5-8にルール例とルール相互間の関係 を示す。

表5-4 知識ベース・ルール記述例

FFN-N

: F(加圧圧力,0:69) -> F(加氏圧力・しょ) -> F(加低圧力・製工) もし、加圧器の圧力 (KG/CM2)が0 (KG/CM2)以上69 (KG/CM2)以下であるならば、低圧力(加圧器) 翌報装置が発信する

`FHルール

F(蒸高水位替根,T)

→ N(主給水系 - 1] もし、高水位(蒸気発生器)替報装置が発信したならば、蒸気発生器の 主給水系が破断したとは考えられない(確信度 - 1)。

 $HH\nu-\nu$

*IF H(復水墨,0.30:0.99) *THEN F(真空低下碧報.T)

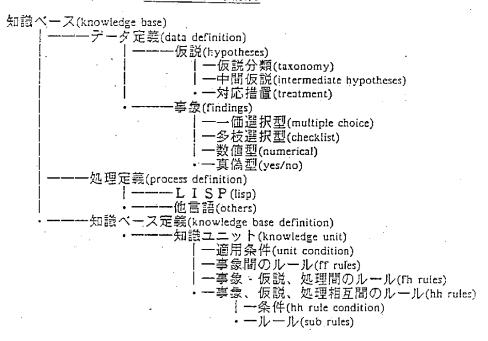
→ H(復水器,0.94)

もし、復水器の事故が(確信度 0.30)以上(確信度 0.99) 以下であるとき、更に

もし、主技水器真空度低下野報装置が発信したならば、復水器の事故である可能性が極めて強い(確信度 0,94)。

表5-3 知識ベースの構成と記述形式

知識ペースの精成



知識ペースの記述形式

	•
ネネネデータ定義	***DATA DEFINITION
未 未仮説定義	**HYPOTHESES
未仮說分類	*TAXONOMY
	:
水中間仮説	*INTERMEDIATE HYPOTHESES
:	
米 対応措置	*TREATMENTS
:	:
宋 宋事象定義	**FINDINGS
:	<i>'</i> :
未未未処理定義	***PROCESS DEFINITION
	ï
水水水知識ペース定義	****KB
キ米知識ユニット ・	**KU
米適用条件	*COND
水FF RULES	*FF RULES
	:
*FH RULES	*FH RULES
diving DILL DC	<u> </u>
*HH RULES	*HH RULES
: atasta for \$20 b	
米米知節ユニット	**KU
:	:
日本語版	英語版
- ·	·

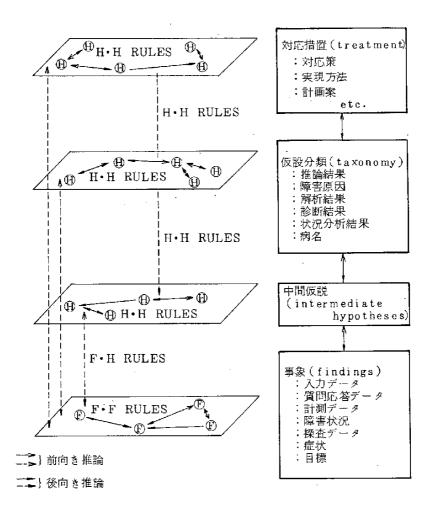


図5-8 ルール適用概念図

ところで、専門家の知識は事実や教科書的知識の集合だけでなく、断片的経験的知識や勘というものを含んでいる。これらの知識を知識ベース化するには、不確実性を記述できるタームが必要である。BRAINSは推論に伴う不確かさの尺度として、MYCINと同様の確信度を採用した。これは、

- ① 人間の確信の度合を近似的に表現できる
- ② 極限の存在, 絶体的肯定, 否定の取扱い, 計算の互換性, 未定情報の取扱いが定義され, 知識ベースシステムにおける不確かさの取扱いの諸条件を満

たしている

- ③ 確信度の合成計算が簡単で処理負荷が小さい。
- ④ この手法を採用している知識ベースシステムが多い 等を総合的に評価した結果である。

5.6 適用分野

BRAINSは1984年3月に第1版を発行以来,定期的に知識ベース構築技法を中心としたセミナーを開催,ユーザの知識ベースシステム作りをサポートしている。それとともにユーザ会を通して,ユーザ相互の情報交換や今後の機能UPの方向を決定している。ユーザは製鉄,機械,自動車などのメーカをはじめ、電力,ガス,石油会社や建設産業,健康管理センター等多岐に亘っており、知識ベース・プロトタイプ作りが活発化してきた。

BRAINSのユーザが現在知識ベース化を進めている分野は、プラントや変電所等の設備診断、送電線の障害や高炉の電気計装故障診断など診断分野をはじめとして、プラント設計時の検器選定や工法のアドバイスなどのエンジニアリングシステム、建築法規のコンサルテーションシステムの構築等幅広い領域に亘っている。この他、CADシステムとの連動やレイアウト設計、プロセス制御への応用も考えられている。以上のシステムは実験レベルのものから実用システムを目指しているものまで様々であるが、これらの知識ベースシステムが本格的な実用レベルに達するには少なくとも1年程度は要するであろう。

しかし、専任の知識工学技術者がいるわけではない一般のユーザが知識ベースシステムの開発に取り組み始めていることは注目に値する。BRAINSの出力例を図5-9,10に示す。

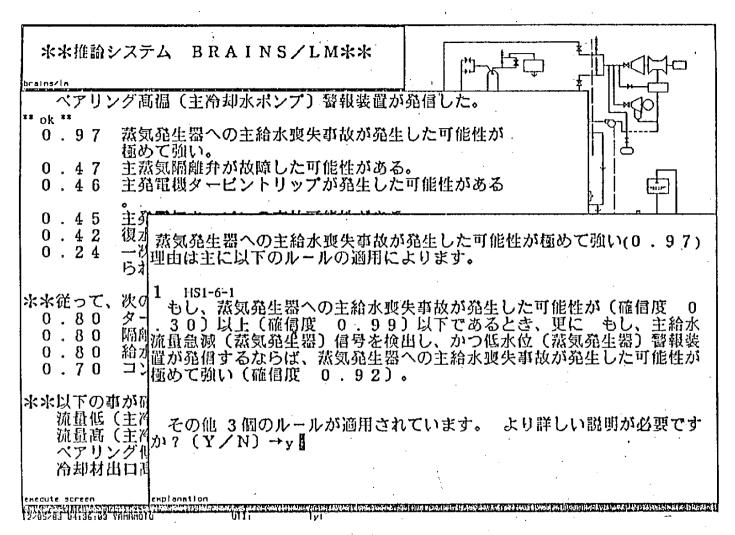


図 5 - 9 原子炉事故診断知識ベース(日本原子力研究所作成) (シンボリック 3 6 0 0 でディスプレイ表示)

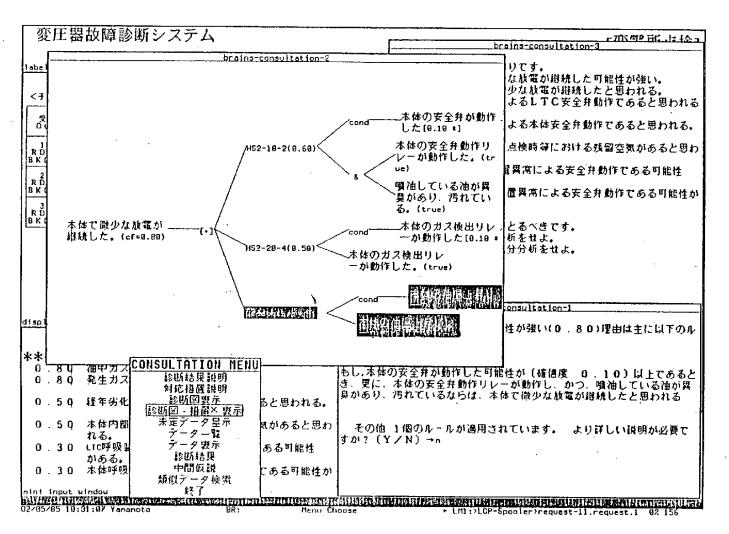


図 5 - 10 変圧器故障診断知識ベース・プロトタイプ (東京電力技術開発研究所作成)

5.7 今後の方向

我が国では産業界主導の知識ベースシステム作りが、ようやく始まりつつある。現場のニーズに充分こたえられるよう経験と不断の改良が積み重ねられたときに、はじめて知識工学が社会のシステム化の技術として定着し、開花す不であろう。BRAINSは知識ベース構築ツールの一つとして多くのユーザに使用され、ダイナミックに発展しつつある。BRAINSの将来動向を図5-11に示す。

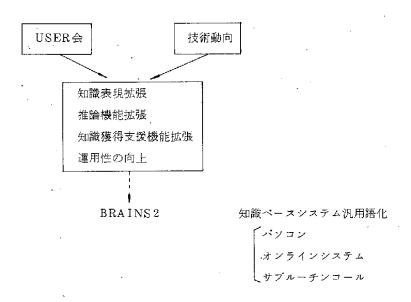


図5-11 BRAINSの将来動向

禁無断転載 ———

昭和 60 年 3 月 発 行

発行所 財団法人 日本情報処理開発協会 東京都港区芝公園3丁目5番8号 機 械 振 興 会 館 内 TEL (434) 8211 (代表)

印刷所 株式会社 タケミ 印 刷 東京都千代田区神田司町 2 - 1 6 TEL (254) 5840(代表)

