

1981年
第5世代コンピュータ国際会議
会議録



昭和57年3月

財団法人 日本情報処理開発協会

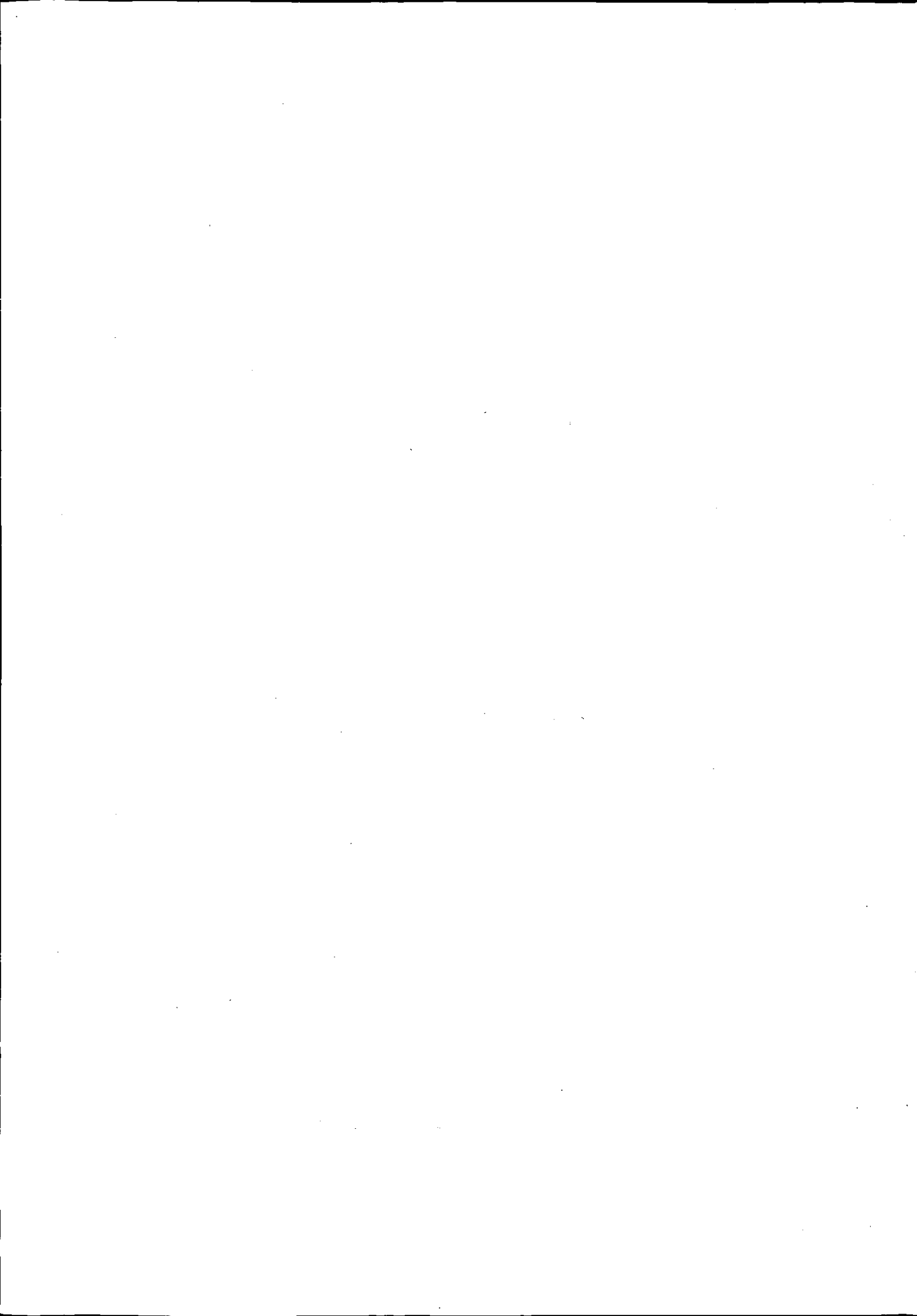
INFOED

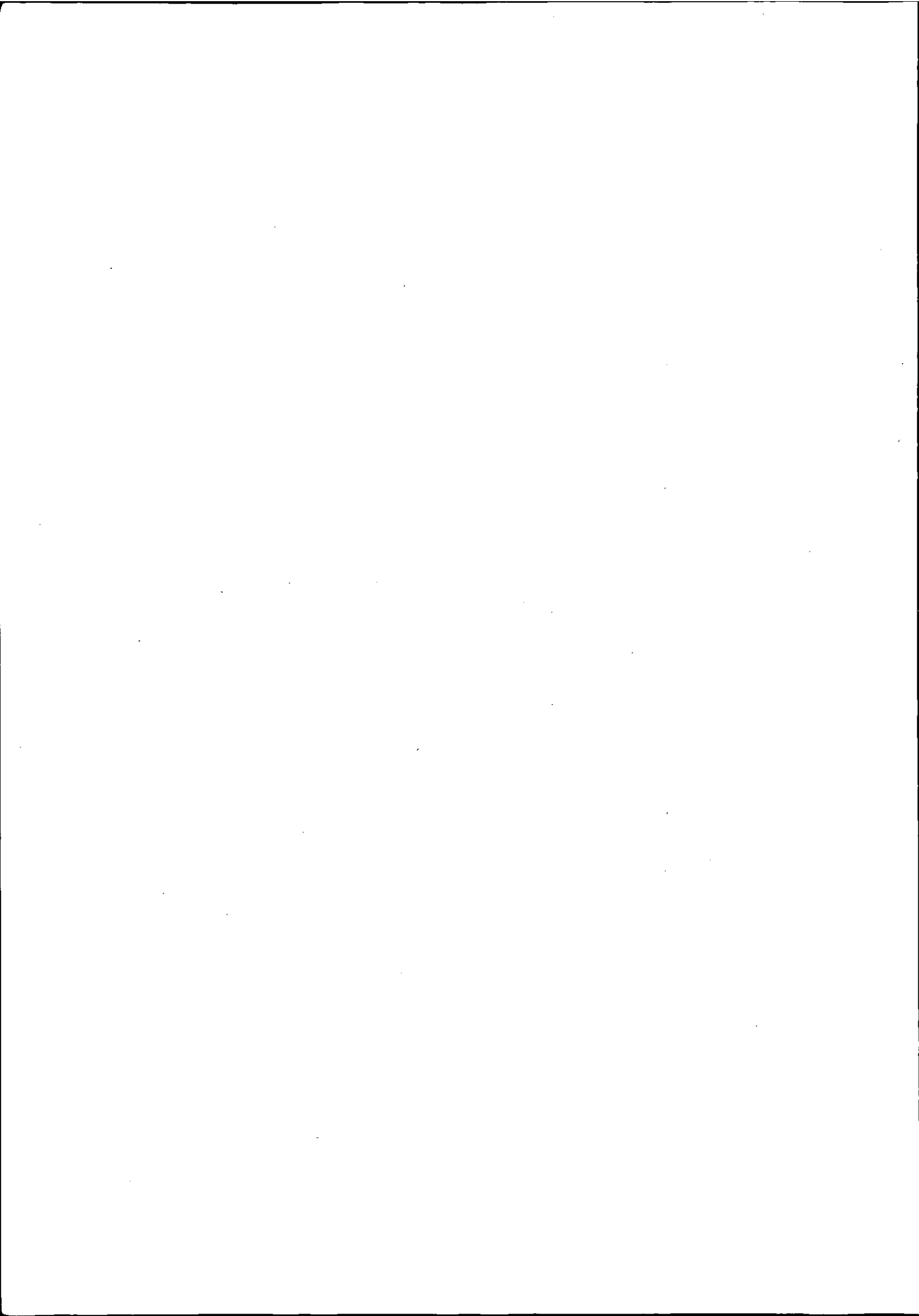
56

R008

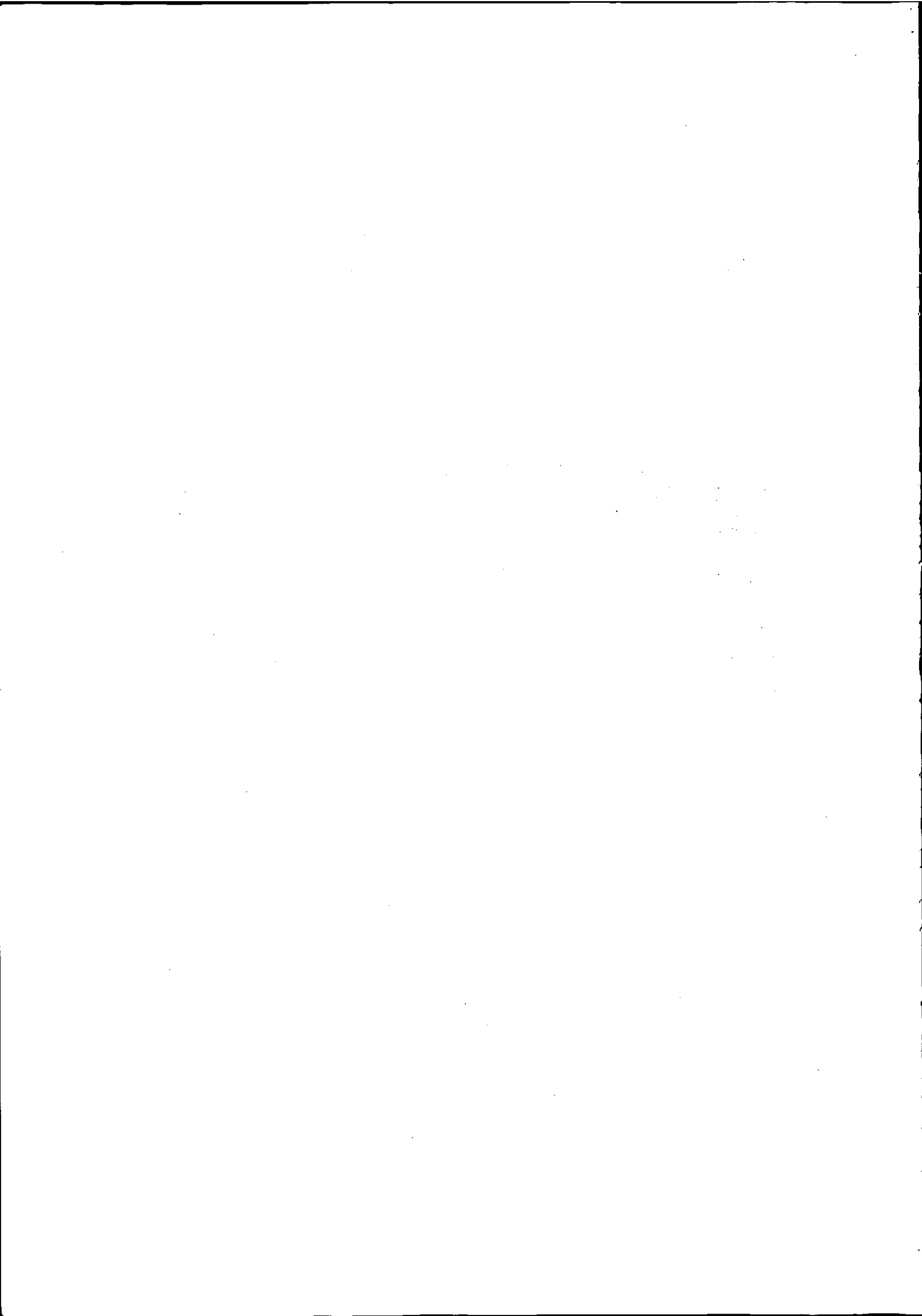
資7.3

この資料は、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56年度に実施した「第5世代の電子計算機に関する内外技術動向調査」の一環として、開催した第5世代コンピュータ国際会議の成果をとりまとめたものである。









は　じ　め　に

第5世代のコンピュータは、1990年代の実用化を目標に、革新的理論と技術にもとづく知識情報処理を指向するものであります。

昭和54年通商産業省の発意により、当協会に第5世代コンピュータ調査研究委員会（委員長 元岡東京大学教授）が設けられ、以来2年半にわたって、このようなコンピュータの実現に必要な研究開発課題および計画についての調査研究が行われました。

ところで、第5世代のコンピュータの研究開発には、わが国のほか諸外国のこの分野におけるすぐれた学者、研究者、技術者の英知が集められ、先進的な専門家の協力がえられることが必要であり、かつ望ましいと考えられます。

そこでこの際、調査研究委員会の研究成果を内外に公表し、これについての率直なご意見、ご批判、ご教示をいただいて、今後の計画推進に資したいという念願から、通商産業省のご支援と関係各方面のご協力のもとに10月19日から22日まで、第5世代コンピュータ国際会議を開催いたしました。

さいわい、この会議には、世界15ヶ国から300名をこえる参加者の出席をえて、日本側から委員長、各主査の研究計画および内容の説明と外国からの招待講師の講演が行われ、また内外の専門家のパネルディスカッションにおいて、活発かつ有益な意見の交換が行われ、多大の成果を収めることができました。しかし、欧米諸国、特にコンピュータの開発国であり先進国である米国がこの日本側のプロジェクトに対してどのような反応と行動を示すかは、これからの重要な課題であろう。

本資料は、国際会議における、各講演者との質疑応答、およびパネルディスカッションの討議内容を取りまとめて会議録としたものでありますが、国際会議の参加者におくばりした「Proceedings of International Conference on Fifth Generation Computer Systems」をご参照いただければ、会議録の理解に役立つと思います。

本資料を取りまとめるに当たり、国際会議にご協力いただいた内外の講演者、パネリストをはじめ、参加者および第5世代コンピュータ委員会関係者に深く感謝の意を表する次第であります。

昭和57年3月

財団法人 日本情報処理開発協会
会 長 島 田 喜 仁

1981年 第5世代コンピュータ国際会議

日 時：昭和56年10月19日（月）～22日（木）

会 場：経団連会館

主 催：（財）日本情報処理開発協会

後 援：通商産業省

組織委員会：委員長 元 岡 達 東京大学工学部電気工学科教授

プログラム委員長

相 磯 秀 夫 慶応義塾大学理工学部
電気工学科教授

淵 一 博 電子技術総合研究所
パターン情報部長

唐 津 一 技術評論家

事務局長

山 本 欣 子 （財）日本情報処理開発協会
開発部長

会議録目次

I 10月19日(月) 13:30~17:30

1. 開会式	1
(1) 開会宣言	(財)日本情報処理開発協会 専務理事 手島篤二
(2) 歓迎あいさつ	(財)日本情報処理開発協会 会長 島田喜仁
(3) 祝辞	通商産業大臣 田中六助
2. 基調講演	2
— 知識情報処理システムへの挑戦 —	東京大学 元岡 達
3. 第5世代コンピュータ調査研究の概要	11
(1) 社会環境条件と第5世代コンピュータの波及効果	11
	技術評論家 唐津 一
(2) 知識情報処理を旨として	14
	電子技術総合研究所 淵 一博
(3) 第5世代コンピュータ・アーキテクチャ	16
	慶応義塾大学 相磯 秀夫
・質疑応答	19

II 10月20日(火) 9:30~17:30

1. 知識情報処理・研究計画	23
(1) 問題解決・推論機構	23
	電子技術総合研究所 古川 康 一 他
(2) 知識ベース機構	26
	電子技術総合研究所 諏訪 基 他
(3) 知的マンマシンインタフェース	29
	電子技術総合研究所 田中 穂 積 他
(4) ロジック・プログラミングと専用高性能パーソナル・コンピュータ	33
	電子技術総合研究所 横井 俊 夫 他
2. 招待講演	36
(1) 第5世代コンピュータシステムにおける記号操作の革新	36
	スタンフォード大学 エドワード・A. ファイゲンバウム
(2) 論理的プログラム合成	40
	ミュンヘン工科大学 ウルフガング・ビーベル
(3) 記号演算の展望	42
	国立情報処理オートメーション工学研究所 (INRIA) ギルス・カーン

III 10月21日(水) 9:30~17:30

1. アーキテクチャ研究計画	47
(1) 推論機構のための新アーキテクチャ	47
	電子技術総合研究所 内田 俊 一 他

I. 10月19日(月) 13:30~17:30

1 開会式

- (1) 開会宣言 (財)日本情報処理開発協会
専務理事 手島篤二
- (2) 歓迎あいさつ (財)日本情報処理開発協会
会長 島田喜仁
- (3) 祝辞 通商産業大臣
田中六助

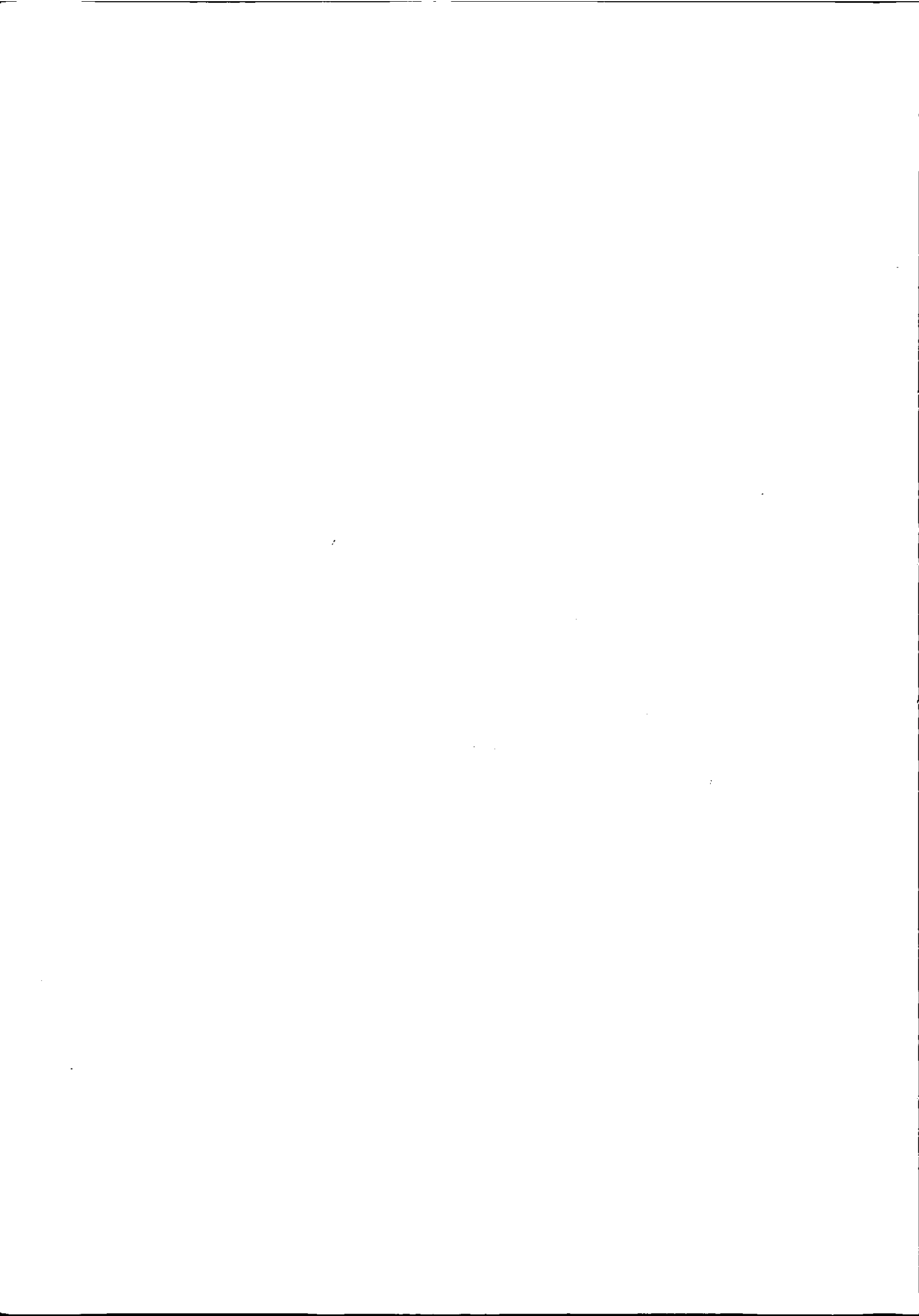
2. 基調講演

— 知識情報処理システムへの挑戦 —

東京大学 元岡 達

3. 第5世代コンピュータ調査研究の概要

- (1) 社会環境条件と第5世代コンピュータの波及効果
技術評論家 唐津 一
- (2) 知識情報処理を旨として
電子技術総合研究所 淵 一博
- (3) 第5世代コンピュータ・アーキテクチャ
慶応義塾大学 相磯秀夫



1 開会式

(1) 開会宣言



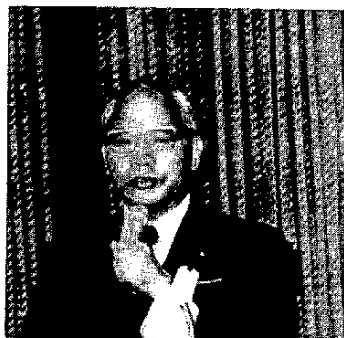
(財)日本情報処理開発協会
専務理事 手 島 篤 二

(2) 歓迎あいさつ



(財)日本情報処理開発協会
会 長 島 田 喜 仁

(3) 祝 辞



通商産業大臣
田 中 六 助



I.2 基調講演

知識情報処理システムへの挑戦

東京大学工学部電気工学科教授
元 岡 達

ただいまご紹介いただきました元岡でございます。今日は第5世代のコンピュータについて我々が過去2年半調査して参ったことをご報告するわけですが、これは、これからナショナルプロジェクトとして始めようとしている第5世代コンピュータに関する構想をお話するわけでありまして、この会議の意義といたしましては、我々が考えた構想に対する皆さんのご意見を伺い、これを、より良いプロジェクトにするために役立てたい、というのが我々の1つの希望でございます。それと同時に、このような類似のプロジェクトが諸外国、あるいは各方面の機関において同時に発足し、相互に刺激し合い、協同して次の世代のコンピュータが開発されるということを期待して、この会議をもちたいと思っております。

第5世代コンピュータという言葉自身、第4世代のコンピュータがまだ世の中に現われていないときに、第5世代のコンピュータという言葉を使うことにいろいろ抵抗があるかもしれませんが、ここで使いますこの言葉の意味は、ここに書きましたように、1990年代に実用されるべきコンピュータ・システムのために、1980年代に国として研究開発する必要がある課題を明らかにしてその研究開発を進めることが、我々が計画しているプロジェクトの目的であります。

当然、こういうプロジェクトを開始するにあたりまして、VLSIをどう利用したらよいかあるいはアーキテクチャの見直しをどういう立場からしたらよいかといったようなハードウェアに対する問題意識。あるいは要求機能の高度化、ソフトウェアの巨大化・複雑化。あるいはソフトウェアの生産性がなかなかあがらないといったような、ソフトウェアに関するいろいろ

な問題意識もこの背後にあります。それから、人工知能、コンピュータ・サイエンス、あるいはソフトウェア工学といったいろいろな研究が成果をあげてきているわけですが、これが必ずしも実際のコンピュータの技術としっかり結びついておらず、その間にはかなりのギャップがあるといった基礎理論に関する問題意識もございます。

現在急速な変化を遂げつつある社会、そしてコンピュータの新しい応用分野がどんどん開けて行く、そういった社会ニーズをどうとらえて、それをコンピュータの技術にどう結びつけていくか、といったことが基本的な問題意識としてあり、今から2年半ほど前に調査委員会がスタートしたわけでありまして、図1が最初の2年間に行われた委員会の構成であり、本日私が

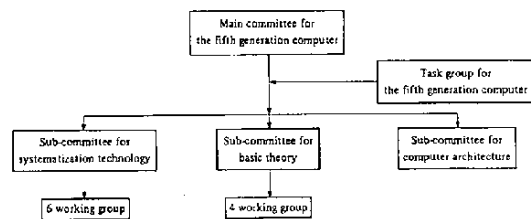


図1 Organization of the Fifth Generation Computer Committee (1980)

お話しする内容というのは、この委員会ですらまとめたものであります。システム化技術研究分科会、アーキテクチャ研究分科会、基礎理論研究分科会、という3つ

の分科会の下に、更に数多くのワーキング・グループができ、約100人以上の方々（コンピュータのユーザ、実際の製造に携わるエンジニア、大学や研究所の研究者など）にご参加いただきましてまとめたものであります。

今年度は少し組織が変わり、図2にあるとおりですが、通産省の中に設けられ、JIPDEC というのがあ

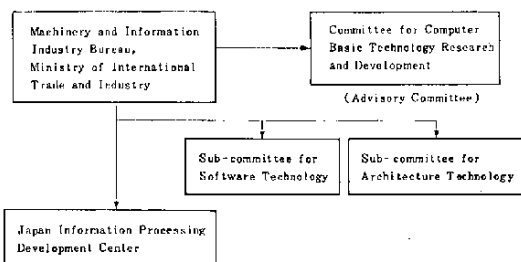


図2. Organization for Research into the R & D of Fifth Generation Computer Systems (1981)

るのですが、ここの中に更に5つぐらいの別の委員会があります。そういう構成でなお続けているわけで、今年度中に構想をまとめて来年度から正式にこのプロジェクトがスタートすることを我々は期待しているわけでありませう。

まず、社会的な要請としてどういう問題があるかということですが、我々としては1990年代の理想的な社会像というものをお考えして、そしてそういう社会にもって行くために生ずるボトルネックをコンピュータが解決するためにコンピュータが何をしなければいけないか、という観点から、こういう4つの問題を取り上げたわけでありませう。

- (1) To increase productivity in low-productivity areas
- (2) To meet international competition and contribute toward international corporation
- (3) To assist in saving energy and resources
- (4) To cope with an aged society

図3. Social Requirements of computers in the 1990's

まず第1は、低生産性分野の生産性向上であります。ご存知のように、コンピュータ制御であるとか産業用ロボットといったものの利用により、我が国における第2次産業の生産性はかなりの向上を見ることができたわけでありませうが、第1次産業あるいは第3次産業

の生産性の分野では相変わらず低いものがあり、そういう分野にこれからのコンピュータが大いに使われなければならないと考えているわけでありませう。代表的なものとしては、オフィス・オートメーションであるとか非定型業務の処理といった問題があげられます。

2番目の問題としては、資源の少い我が国の場合は国際競争力を確保して原料を輸入することができなければ国の経済を維持して行くことができないわけでありませう。そういう国際競争力を確保し、一方では国際的な協力関係を維持し諸外国の文化の向上に貢献するためには、コンピュータ産業のような知識集約型の技術を我が国で大いに広める必要があるわけでありませう。そして、データベースであるとか機械翻訳であるとか、知能ロボットであるとかCADの技術といったようなものを伸ばして、それらを通して国際的な世界の文化の向上に寄与することが我が国に求められていると考えるわけでありませう。

3番目としては、エネルギー問題、省資源の問題が我が国としては非常に大きな問題であります。我が国のエネルギーの95%近く、あるいはそれ以上を輸入に頼って行くわけであり、省エネルギー、省資源という問題にコンピュータが協力する必要があります。そのためには、こういった分野の研究開発のためのシミュレーション、あるいはエネルギー関係装置の制御、あるいはComputer Aided Design, Computer Aided Manufacturingの進歩による省エネルギー・省資源の問題に協力するということでありませう。

4番目の問題は、高齢化社会への移行の問題であります。現在の我が国は、それほど社会が高齢化しておりませうが、これから10年の間に急速な高齢化が進んで、10年後にはヨーロッパを追い抜くような高齢化社会になると予想されています。そういう急速な社会の変化に対応して行くために、老人の健康管理であるとか、生涯教育や老人のいろいろな活動を援助するためのシステム、あるいは老人がオフィスに出かけて行かなくても自分の家で作業ができるような環境をつくり出す……といったことが必要だと考えているわけでありませう。そういうものをつくるのにコンピュータが貢献するでありませう。

次に、今日のコンピュータがかかえている問題点。これについては、皆さん専門の方々ですので詳しくご説明するまでもないと思ひますが、まず第1に、現在のコンピュータは数値データの処理を主目的としたものであり、図形や画像や文字や音声といったような非数値データの処理のための能力が不足しており、これが新しい応用分野の拡張に大きなボトルネックになつ

ていると言えると思います。

2番目に、現在のコンピュータが逐次制御であり、ハードウェアの機能というものが必要最小限に限られているということでもあります。これは今日までのハードウェアの技術の信頼性、あるいはコストといったようなものが、こういったアーキテクチャを生み出しているわけですが、VLSIその他の技術によってそういう前提条件はかなり変わったと考えることができます。したがって、並列制御した新しいタイプのコンピュータというものを考える必要があります。それから、価格対性能比その他の点から大容量、大型のコンピュータを使うことが有利だと考えられていたわけですが、これも大量生産に適したLSI技術を使うことによって分散処理の方向へ移行して行くということが可能になってきたわけでありです。

それから、もう1つ大きな問題といたしまして、ソフトウェアの開発コストに膨大な費用がかかり、将来は全開発コストのうちの80%近くがソフトウェアの開発に使われるということすら言われています。それは、コンピュータ産業全体として考えた場合には、けっして健全な姿ではありません。やはり開発のコストがハードウェアとソフトウェアにバランスして投じられるような技術にもって行くことが望ましい姿だろうと思います。高級言語向きのアーキテクチャを考えると、ソフトウェアの生産性を向上できるようなアーキテクチャを考えるというようなことが必要になってくると思います。

以上のようないろいろな問題を解決するために我々が使うことのできる技術として何が考えられるか……というのを次に列挙してみます。

まず第1に、VLSI技術というものが、既に記憶その他ではかなり使われているわけですが、これを論理の面に使うのには研究しなければならない多くの問題があります。しかし、これをとにかく使いこなすということが、新しいものを作って行くための原動力になると期待するわけでありです。

2番目に、Josephson JunctionであるとかGaAsであるとかいった高速素子の技術、あるいは光通信の技術といったものも、今後使っていくことのできる技術として考えられます。

3番目は、通信技術とコンピュータ技術の融合であります。これによって、新しいコンピュータの利用法というものが考えられます。

4番目は、並列処理の技術であります。並列処理については、過去に長い研究が行われてきたわけであり

ますが、やはりハードウェアのコストが大きな比重を占めている時代には、なかなかこれを有効に使って行くことはできなかったわけでありです。しかし、VLSI技術により、この並列処理の技術が本当に生かせる時代がやってきたというのが我々の認識であります。

5番目に、ソフトウェア工学についても、モジュール化とかデータ抽象化、あるいは関数型の言語、非手続型の言語、あるいは論理プログラミングといった新しいいろいろな技術が開発され、使われるのを待っている状態であります。こういうものを使いこなすためには、新しいアーキテクチャの下で使う必要があるということには言うまでもありません。

16番目に、人工知能とかパターン認識の技術、特に人工知能技術は、この10年あまりの間にかかなりの基礎技術が行われました。しかし、これを実用的な技術にまで高めるためには、かなりの知識の集積、多くの人々の協力が必要な時代になっていると思います。いままでに突ってきた基礎的な研究を土台にして、これを実用化に結びつけるための努力というのが今後必要とされているわけでありです。

以上のような前提条件の下に、第5世代のコンピュータに対して要求される機能として、ここにあげた4つの機能があると思います。

まず第1には、コンピュータの知能レベルを高めて、人間の良き協力者としての親和性を高めることです。これは、コンピュータの中に推論の機構であるとか、連想また知識を獲得するために必要な機能といったようなものを持たせることであります。

2番目は、人間の代替をする能力や人間にとって未知の分野を開拓する場合の支援能力をもつことが望まれます。感覚器官1つをとってみても、人間が持っている感覚器官の能力以上のものをコンピュータに持たせることが可能であります。パターン処理を考えてみても、人間がやるのとまた違った意味のパターン処理をコンピュータに期待することができるかと思えます。もちろん、人間との協力の上に成り立つ話であります。人間が持っているそういった能力を伸ばすことにコンピュータが貢献できるということを期待したいと思えます。

3番目として知識ベース、あるいはデータベースに關係した問題ではありますが、いろいろな形態の情報を必要に応じて簡単に即時に検索することが可能なような情報処理システムを築きあげることが、技術的には可能になってきていると思います。こういうことが意思決定の支援システムをつくるとか、Com-

puter Aided Design システムをつくるといったことのベースとして非常に重要な機能となるわけであります。

4番目といたしましては、科学技術をはじめ、経営とか行政とかいろいろな社会問題、そういった問題についての大規模なシミュレーションをコンピュータを通してやることによって、新しい知見を我々人類が持つことができるようになるという期待があるわけであります。そういうことのできるようなコンピュータをつくらう、ということであります。

今は少し大上段にふりかぶった議論になったわけですが、もう少し現実的にユーザの観点から、新しいコンピュータに要求される機能というのを考えてみますと、ここにあげたように5つの項目をあげることができるかと思えます。

- (1) Easy to use functions capable of being utilized even without professional knowledge
- (2) Human substitute functions capable of judgement and decision making
- (3) Functions capable of flexible configurations applicable to a wide range of jobs
- (4) Functions for facilitating programming
- (5) System functions which are reliable and can be used expediently

図4. Requirements from user's viewpoints

まず第1は、専門知識がなくても利用できるような使いやすい機能を実現することです。マルチプル・インプット・メディアを使いこなすとか、自然言語を使うようにするとか、コンピュータにコモンセンスをもたせるとか、場合によっては専門知識をもたせるといったようなことであります。

2番目は、判断とか意思決定において人間を支援できるような機能であります。

3番目としては、多様な業務に適用できるような柔軟な構成を可能にするような機能というのが必要であります。

4番目は、プログラムの生産性の向上と同じですが、プログラムをある程度自動的に作製したり修正したりするような機能、あるいは常識的な判断は書かなくてもやってくれるといったようなことを通して

プログラミングを容易にする機能というのが望まれます。

5番目に、そういったシステムが当然信頼性が高く、コスト・パフォーマンスの点でもすぐれ、好きなところで自由に使え、故障に対しても自動修復の機能を持ち、機密保護についても十分な配慮がなされたシステム構成でなければいけないわけです。

以上のことをひと口にまとめて言えば、第5世代のコンピュータ・システムというのは、従来のコンピュータの技術的な制約を克服して、1990年代に要求されるであろう高度の機能に対応し得る革新的な理論と技術に基づく知識情報処理指向のコンピュータと行うことができると思います。そういったコンピュータ・システムの基本的な機能としては、問題解決と推論をすることのできる機能。2番目としては、知識ベースを管理する機能。3番目として、人間に近い知的インターフェースを実現する機能。そういった3つの機能をあげることができます。

これが、今申しました第5世代コンピュータ・システム概念を図で表わしたものであります。(図5) ちょっと小さくて見にくいのですが、3つのシステムからなり、1番左が人間に対応するアプリケーション・システムであります。右側はハードウェア・システムで、その間にモデリングのソフトウェアのシステムがあると考えられます。現在のコンピュータの利用では、大ざっぱに言って、この辺は人間が分担しこの辺はソフトウェアが担当し、このところはハードウェアでなされる。

現在のコンピュータはまだこの辺のところにハードウェアとソフトウェアのインターフェースがある。しかしそれが第4世代の間にこの辺までくるのであります。それまでに実現するものとして、データベース・マシンとか数値計算用のマシン、あるいは記号処理のマシンといったものがあげられているわけです。

それに対して第5世代のコンピュータでは、推論や問題解決の機能というものがハードウェアで実現されます。それから、知識ベースを管理するためのハードウェアもできます。知識ベースの中味というのがここにあります。そして、その種類というのは入出力に関連した言語に関係したものとか、イメージに関係したような知識、こちら側のハードウェアではマシン・モデルについての知識、そういうものから実際の問題についての知識、そういうものから知識ベースは成り立って行きます。そしてその知識ベースを使っていろいろな問題の理解であるとか、要求仕様の理解をする

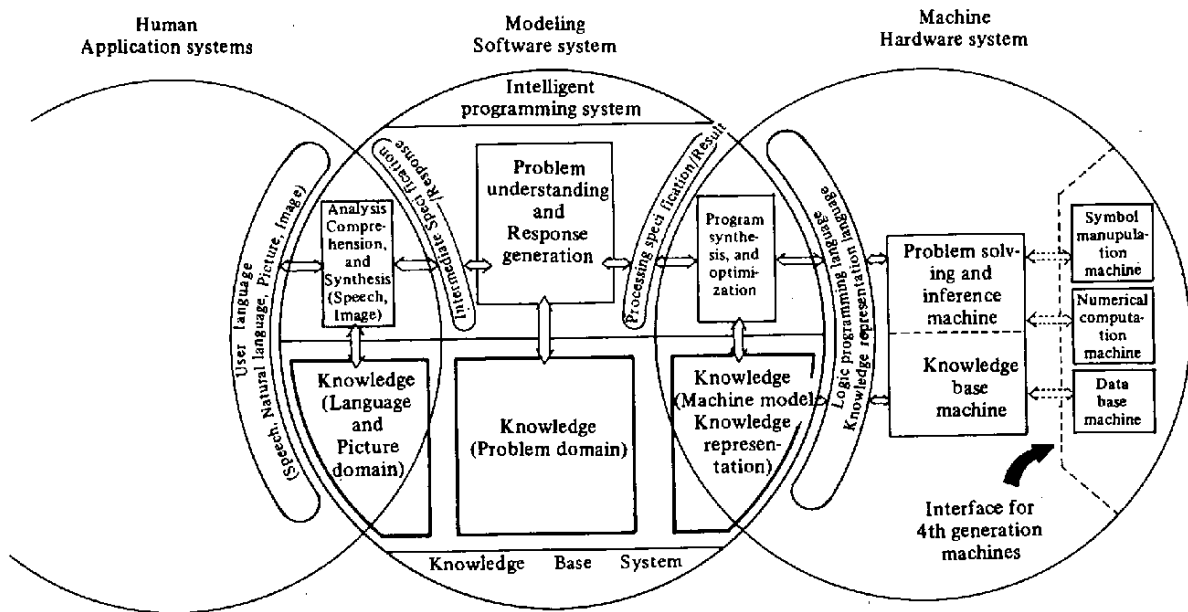


図5 Conceptual diagram of a fifth generation computer system as viewed from the standpoint of programming

ようなソフトウェアが準備されます。そして、人間とソフトウェアのインターフェースとしましては、音声、画像、自然言語といったものが使われます。そういうものを介してコンピュータ・システムと人間とが情報を交換し、コンピュータの方でそういう言語とか図形といったものが理解され、それがこの問題の理解を通して処理仕様となります。そしてその処理仕様のプログラムの最適化が行われて、こちら側の論理型の言語として、ハードウェアが理解し取り扱うことのできる言語に転換されてハードウェアに渡されます。その答えが逆の経路を通り、人間にわかりやすい格好にされて人間の方に渡されるといったイメージを描いているわけでありませう。

このプロジェクトで考えております1番アプリケーション側のシステムの例といたしましては、ここに書きましたような機械翻訳システム、質問応答システム、音声応用システム、図形画像応用システム、応用問題解決システムといったものを考えています。例えばここで機械翻訳システムと言いましても、これはいろいろな国語の間の変換に対して共通に必要な道具を提供するというので、ある特定の2カ国語の間の翻訳システムというのは、これの上に構築される実際のアプリケーション・システムになるという考え方でありませう。

右の図は、この1番外側の点線がアプリケーション

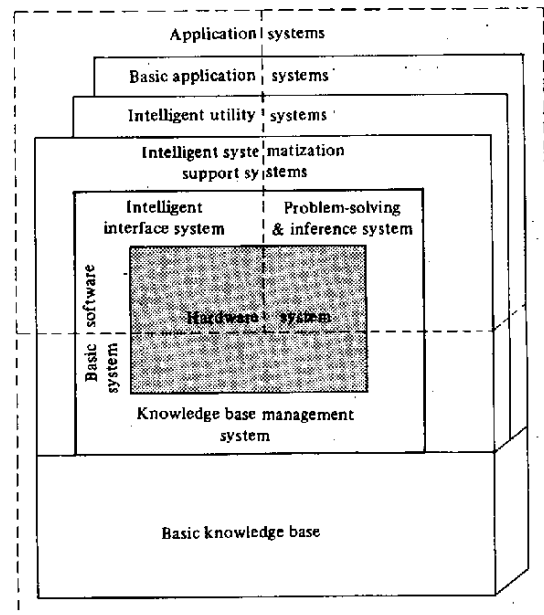


図6. Conceptual diagram of the composition of a fifth generation computer software system

・システムを指しています。例えば、英仏間の翻訳システムをつくるというのは、このアプリケーション・システムの部分につくられるわけで、それを実現するためのベーシックなアプリケーション・システムというのが先ほどあげました5つのシステムであります。これを使いこなすためのいろいろなサポートシ

テムといったものも、必要になります。現在のコンピュータのオペレーティング・システムに相当するようなものとしては、このベーシック・ソフトウェア・システムと書かれた部分であります。このベーシック・ソフトウェア・システムの3つの機能として、入出力に相当する知的インターフェース・システムと、データベースに相当する知識ベース管理システム、それか

ら今の数値計算用のプロセッサに代わるものとして問題解決・推論システムというのがあります。そこで、ソフトウェアとハードウェアが協同してそういうものを提供するというを表したのが、下の図であります。今言いました、この3つの機能が、中心的な機能としてあげられるわけであります。

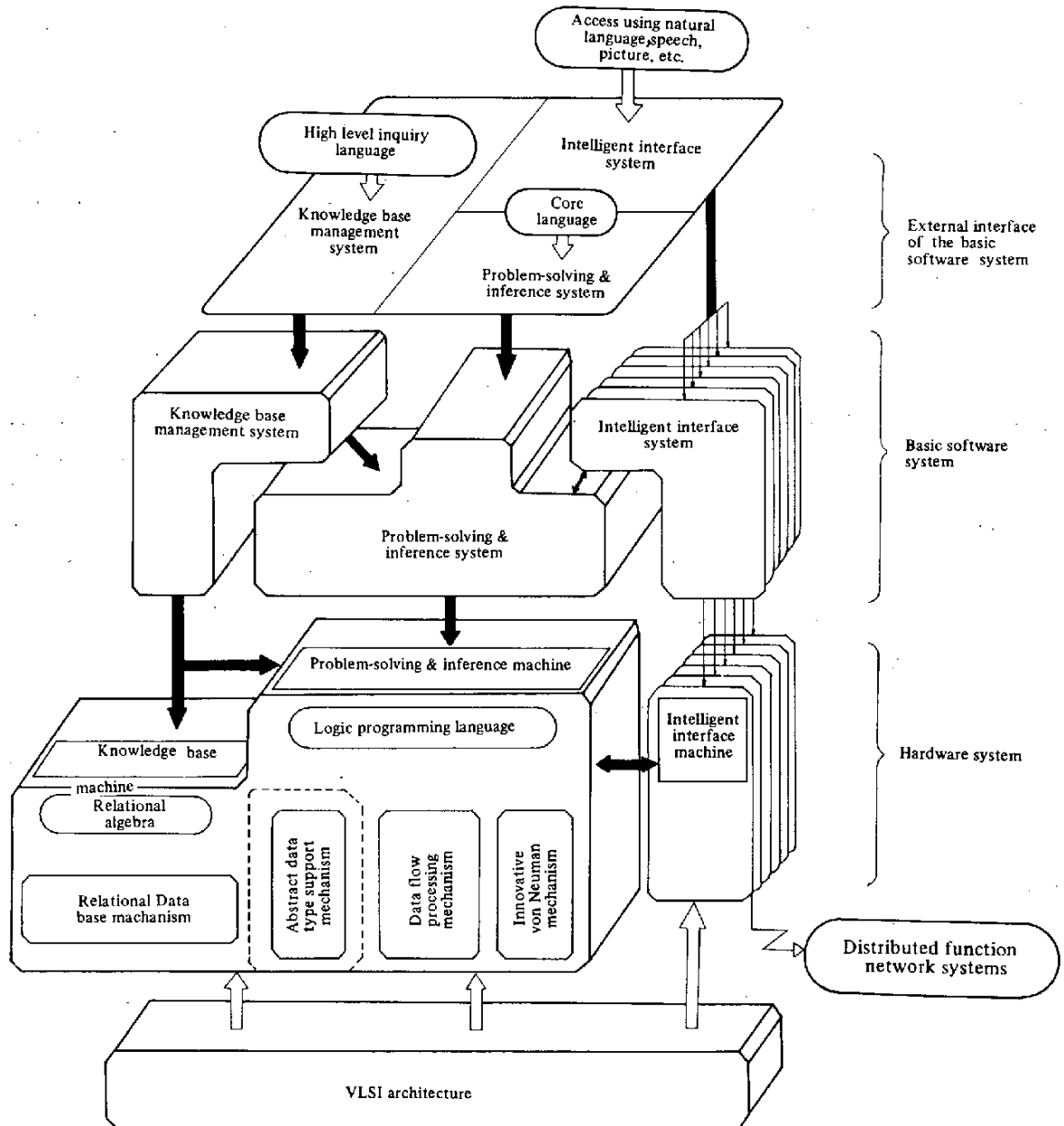


図7 Basic configuration image of the fifth generation computer systems

これは今のことを、もうちょっとハードウェアの部分を中心にして書いた図面ですが、ここから上がソフト

ウェアでベーシックなソフトウェア・システムですが、それを支えるハードウェアとして問題解決と推論のマシンがあり、それから知識ベースのマシンがある。

その他に、いろいろな入出力に対応した知的なインターフェース・マシンがある。いろいろなイメージ・プロセッサとか音声の処理プロセッサというのはこういうところに入るわけであり。その中心をなす推論のメカニズムとして論理型プログラミング言語をソフトウェアとハードウェアのところのインターフェースの言語としてもつて来たい。それを実現するためのコントロール・メカニズムとしてデータ・フローのメカニズム、あるいは現在のフォン・ノイマンを改良したようなメカニズム、また入出力に近いものとしては抽象データ型や先ほどのオブジェクト指向マシンといったような考え方を通して、そしてこちら側のデータベース・マシン、これについては関係データ・モデルを考えて、その関係代数といったレベルのものは当然ハードウェアとして実現することを考えているわけであり。そういったものを実現するのに当然VLSIを使って実現する必要があるわけで、VLSIを使いこなすのに向いたアーキテクチャという観点からこういう問題をもう一度見直して、こちら側からの要求と上側からの要求をうまく考慮して、ここ間のコントロール・メカニズムをつくって行こう……それを実現することが第5世代のコンピュータ・システムの基本的な構想であると考えているわけであり。

今まで申し上げたことをいくつかの研究テーマとして取り上げたのが、表1であります。これについては資料がありますのでそれを参照していただければ、今申し上げたことがこの辺の項目としてすべて含まれています。これが、我々が今後進めて行く研究課題であります。これを全部取り上げると非常に膨大なものになりますので、これの中から吟味して緊急度の高いものから研究して行くという方法をとらざるを得なくなると思っています。これは小さく見にくく恐縮なのですが、このプロジェクトとして一応10年ぐらいの研究期間を考えておまして、それを前期・中期・後期とだいたい3分の1ぐらいずつに分けて、それぞれの段階で何を研究してその成果をその段階で評価し、そして次のステップにつないで行き、最後に今申し上げたような第5世代コンピュータ・システムをつくり上げて行こうと考えているわけであり。

大ざっぱに言って、前期では基礎的な研究をベースに基本的な機能の実現方法というものを研究し、中期ではそういう基本的な機能を使って重要な部分のサブシステムの構築を研究し実現する。そして後期は、そういったサブ・システムを結合し本格的な性能をもったシステムを実現するというのが最後の段階でありま

す。以上で、だいたいのプロジェクトのイメージの説明を終らせていただきます。このプロジェクトは非常に高い目標を掲げているわけでありますが、従来我が国で行われてきたナショナル・プロジェクトというのは、目標もかなりはつきりしており達成の可能性もかなり高いものが目標として選ばれてきたと思われ。しかし、現在我が国がおかれている国際的な立場を考えてみましても、そういう達成が可能であるようなプロジェクトをナショナル・プロジェクトとして取り上げることは必ずしも適当ではなく、達成に多くのリスクがある問題こそナショナル・プロジェクトとして協同してチャレンジして行く価値があり、またその必要があると考えているわけです。

先ほども申し上げましたように、素人にもわかりやすいようなコンピュータをつくって、皆が安心してコンピュータを使えるようにするということが、これからのコンピュータが社会に受け入れられるための必要条件であると考えているわけです。これからの社会に対するコンピュータのインパクトというのは非常に大きなものがあると考えられます。しかし、そういう非常に大きなインパクトがあるわけですから、使い方を誤れば非常に大きな社会問題を起こす恐れもあります。従いまして、正しい利用法というのを考えながらこういうプロジェクトを進めて行かなければいけないことはもちろんありますが、そういう正しい利用法を推進することに注意を払いながら、このプロジェクトを今後進めて行きたいと考えているわけであり。ご清聴ありがとうございます。

表1. Items in research and development of the fifth generation computer systems

Basic application systems	1-1) Machine translation system 1-2) Question answering system 1-3) Applied speech understanding system 1-4) Applied picture and image understanding system 1-5) Applied problem solving system
Basic software systems	2-1) Knowledge base management system 2-2) Problem solving and inference system 2-3) Intelligent interface system
New advanced architecture	3-1) Logic programming machine 3-2) Functional machine 3-3) Relational algebra machine 3-4) Abstract data type support machine 3-5) Data flow machine 3-6) Innovative von Neumann machine
Distributed function architecture	4-1) Distributed function architecture 4-2) Network architecture 4-3) Data base machine 4-4) High-speed numerical computation machine 4-5) High-level man-machine communication system
VLSI technology	5-1) VLSI architecture 5-2) Intelligent VLSI CAD system
Systematization technology	6-1) Intelligent programming system 6-2) Knowledge base design system 6-3) Systematization technology for computer architecture 6-4) Data base and distributed data base system
Development supporting technology	7-1) Development support system

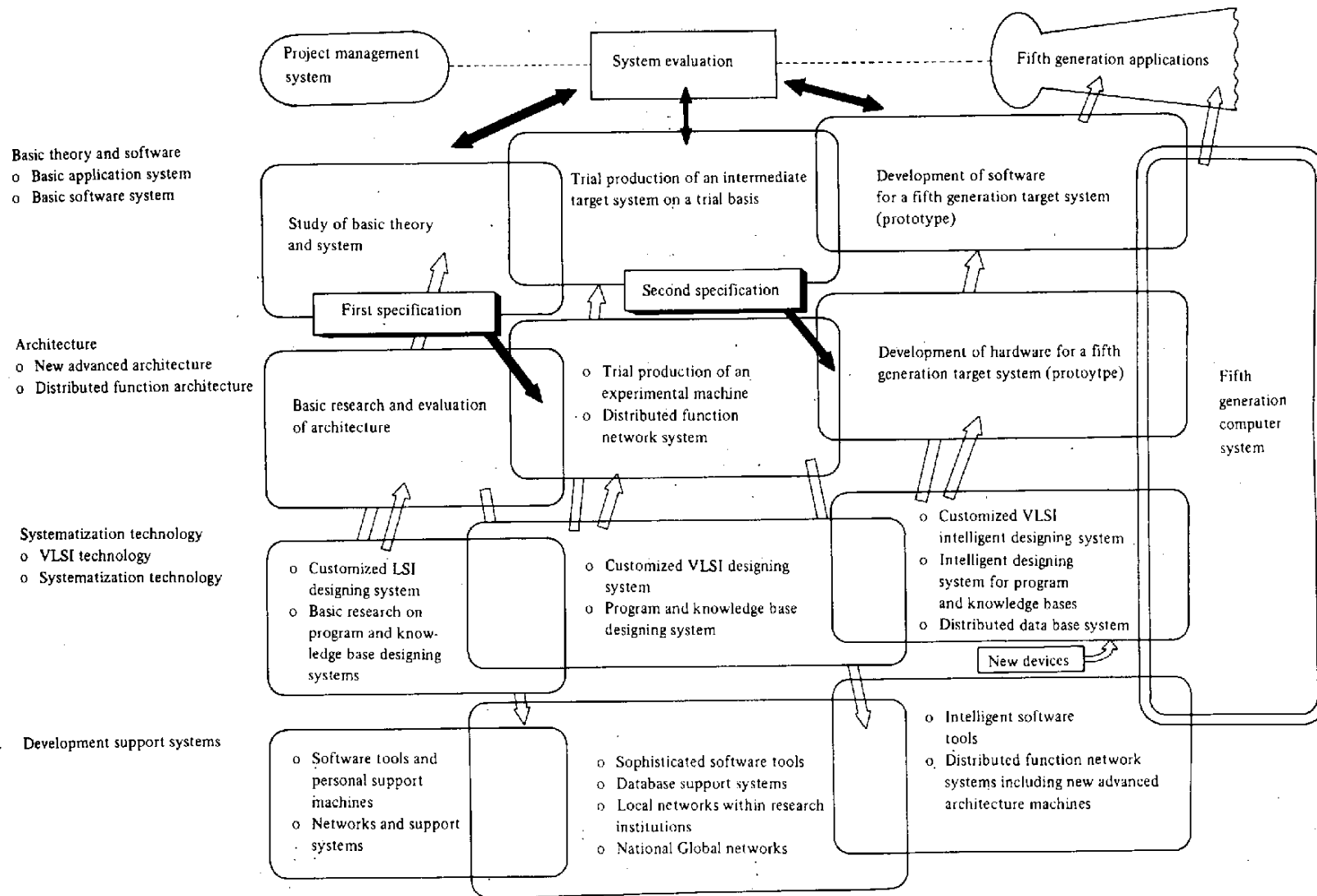


图 8. 概念图显示研究和开发如何进展



講演要旨

1 接近の方法

1990年代の初頭に実現を予想される第5世代コンピュータに対する要求をまとめるために、普通に考えられることは、その時代における社会環境の姿を予想することである。

しかしながら、一般に行なわれている予測の方法はこの場合適切でない。未来は過去の延長ではなくつくり出されるものだからである。

そこで、ここでは別の方法論をとることとした。まず10年先の日本の望ましい姿を考える。そして現在からその姿へアプローチするには多くのボトルネックがある。その中には既に明らかなものもある。たとえば人口の高齢化である。中には今はボトルネックでも将来解消しているものもあるかも知れない。たとえば政府の赤字である。しかし、いづれにせよ、これらのボトルネックを解消すれば、その望ましい姿は実現するであろう。

このボトルネックの解決のために、コンピュータはいかなる役割が果せるか？ここから次の世代のコンピュータの期待が具体的に生まれてくる。このような方法で第5世代への要求をまとめた。

2 5つのボトルネック

ここで示されたボトルネックは次の5つに要約される。

① 低生産性分野への適用

日本の製造業の生産性の高いことは有名であるが、それはGNPの33%を占めるに過ぎない。それ以外の分野は置き去りにされていて、それが物価を始め多くの社会的な歪みを生んでいる。ホテルで夕食をとるとその費用で電卓が4台も買えるというのは異

I.3 第5世代コンピュータ調査研究の概要

(1) 社会環境条件と第5世代コンピュータの波及効果

技術評論家

唐津 一

常である。これが今後10年の最大のボトルネックになる恐れがある。製造業の中においても生産性の低い分野がある。事務、設計といった手作業の仕事である。農業、流通、教育、医療、行政などもその代表である。これらの分野からのコンピュータへの期待は大きい。

② 国際化の時代

日本のGNPが世界の10%を越した。これは日本の行動は大小にかかわらず、世界に影響を与えるということである。また中進国の日本への追上げもきびしい。このようなボトルネックの解決のために求められるものは、機械翻訳、CAIといったことだが、更には国際競争力の維持のためのCAE/CAD、ロボットへの要求がある。

③ 高齢化と高教育化

日本の高齢化は急速に進み若年労働力は減少する。しかも教育水準は上る一方である。これはダーク・ワーク離れを促し、生涯教育への要求が出てくる。トップヘビーな企業の体質を更に改善し、競争力を維持するためには知識ベースの発展も重要な条件となるだろう。

④ 省エネルギーと省資源

エネルギーと資源問題は、常に目の離せないテーマである。新資源の開発、エネルギー利用の効率化から来る要求は大きい。

⑤ 個人の問題

情報化時代における個人の問題は、我々にとって全く新しい経験である。そして、そこからの問題をいかに解決するかはコンピュータ利用の方向を直接左右するほどの問題となる可能性がある。

3 第5世代コンピュータへの要求

このような分析から、第5世代に対して求められるコンピュータの姿が整理されてくる。それは次のようなことである。

① 親しみ易いコンピュータ

これまでのコンピュータはやはり専門家のものであった。誰もが自然な言葉で語りかけることで利用できる機械がほしい。これはマンマシン・インタフェースの問題から始まって、言語、推論機能、知識ベースといった形で具体的な要求へとブレークダウンされる。

② 創造的、知的活動へのサポート

既にCADが活躍はしているが、更に高度な計画や設計へのサポートをしてくれるコンピュータである。

③ 未知の分野へ踏み込むためのもの

超高速コンピュータはその一つであるが、新しい概念や知識の開発を容易にしてくれるコンピュータである。

④ 新知識抽出

膨大なデータの山から新しい知識を抽出してくれるようなコンピュータ、たとえば、スタンフォード大学のヒューリスティック・プログラミングのような発想に基づくもの。

このようなことから、①使い易い、②不確実なジョブ、③高速シミュレータ、④知識処理、⑤巨大データベース、といった目標が得られた。

4 社会的インパクト

第5世代が実現した時の社会的インパクトを全て占うことは難しい。もともと情報技術は絵を画くキャンバスと絵具のようなものである。そこで画く画題の選定を誤り、また腕が悪ければ、出来上がった絵に金を払う人はないだろう。従って、第5世代コンピュータのインパクトは、それをどのようなやり方で社会に組込むかによって、著しい差が出てくる。それには組込まれる社会の構造、情報技術利用についての各種の制度など、多くの条件によって、インパクトは異なってくる。

コンピュータは、もともと計算のための道具として開発された。それが個人のプライバシーの問題を呼び起こすなど、当初は誰も考えていなかったはずである。ロボットは日本の若年労働力不足を補ない、ダーティ・ワークからの解放をもたらすものとして、歓迎されているが、欧米では失業をもたらす悪魔ではないか、という形で受けとめられている。

ここでのインパクトは、社会、産業、国際及び個人

の項目にわけて記述されているが、これらはいずれもが密接に関係しており、独立のものではない。

第5世代の方向づけの一つは、誰にも容易に使える、ということである。従って、これを使う人口は爆発的に増加し、用途も多岐にわたるだろうということである。従って量の増大による質の問題をも考えなくてはならない。

① 社会システムの中で

これまでの社会システムは、各々の専門家にゆだねられていた。そして、それは効率的であったかも知れないが、他方ではシステムの相互の整合性において問題を生じることがあった。これらは、更に巨大化するだろうが第5世代によって、巧みな調和を保つことが期待される。だが、システムの巨大化、複雑化からくるバベルの塔のリスクを回避することは、必ずしも容易でない。また個人が直接アクセスすることが実現するだろうが、これからの予想したい状態をどのように受けとめるかの問題もある。しかも、これらは産業システムの場合は対象が主としてモノであるに対して、ヒトであるということからくる不一致性をどのように扱うかという難しい問題がある。つまり、一意的な解が得にくいものを、コンピュータがどう扱うかという課題である。

② 産業へのインパクト

これまでの低生産性分野は急速に改善され、それが、社会の豊かさの原動力となるとともに、進歩の速度を早めることになるであろう。特に1次産業、3次産業の姿が大幅に変化することが予想される。これらの成果は行政の革命の原動力ともなり、その姿は一変するに違いない。話者認識の実現は、印鑑証明といった古典的なシステムの存在を完全に不要にする。許認可業務についても、その多くは機械化が可能である。しかしながら、これらは制度の改変を要するものである。高度な効率化の波に引きずられることからの歪みをどのように解決していくかを考えておかねばならない。

第5世代コンピュータはやはり第2次産業の分野において最大限の利用がなされるに違いない。そして、それは豊かで健康な社会の実現のための資源をつくり出すものとして最も有力な牽引車となるであろう。

③ 国際化について

社会システム、産業システムは共に国際的拡がりを見せる。情報技術は国境という壁の高さと厚みとを徹底的に薄く、また低くしてしまう。しかし、や

はりそこには民族性の差があり、また、国による所得水準のギャップなどがある。それは依然として残るだろう。従って、それからくる摩擦はむしろ増えるはずである。そして、その解決のためにコンピュータの活躍が期待される。

④ 個人とのかかわり合い

すべての人々がコンピュータを意識せずに利用可能になった時、個人の生活にはいろいろなインパクトがやってくるに違いない。そして、それは次の3つに集約される。

- 1) プライバシーの問題
- 2) 新しい社会慣習の形成
- 3) 新しい秩序への移行途中における問題

現在のところ個人の立場からみる時、情報技術は発展したナマの素材のまま我々の目の前に放り出されているというのが現実である。いまのところ、これらの技術は与えられたものであり、人々は特別な知識も訓練なしのままにおかれている、というのが正しい。従って、これからの課題は新しい事態を正しく理解し、そしてこれを扱うことについての開発をすることである。人々は今、新しい可能性という未知の分野の入口に立っている。この巨大な象の鼻にさわっている人もいるし、足にふれている人もいる。また、尻ごみをしている人もいる。これは全く不幸なことである。第5世代の一つの目標は、すべての個人とのかかわり合いについて今までのような専門家とのつき合いとは違った新しい関係をつくり出すことにある。これまでとは違って、第5世代では機械の方から人の方に歩み寄ってくるものだからである。



(2) 知識情報処理システムを目ざして

電子技術総合研究所パターン情報部長

淵 一 博

講演要旨

ソフトウェア基礎論の研究、人工知能の研究、パターン情報処理の研究を包含した広義の基礎理論のサーベイから、第5世代のコンピュータ・システムについて、確定した像を導き出せるわけではないにしても、それに寄与できる段階には来ているように思われる。第5世代コンピュータ・システムについて前もっての定義はない。ただ、現行のシステムの漸進的改良という線上でなく、その先に飛躍した段階を想定したいという問題意識がある。

ここ十年程をふりかえてみると、比較的基礎的な研究分野でも、いろいろと新しい展開があったと見られる。それぞれは単独にも魅力ある話題であるが、それとともに、独自の動機から出発したと思われるいくつかのことが、実は相互に深く関連しあっているということが私にはとくに興味深く思われる。それは「新しい世代」への期待を抱かせる。

例えば、Dennisらのデータフローマシンという話題がある。これは並列処理のアーキテクチャであるが、「非ノイマン型」の有力候補としてこのところ注目されている。構成法としての面白さの他に、この提案が魅力的なのは、それがソフトウェア工学における「関数的プログラミング」と密接に結びつくことである。プログラミングとアーキテクチャの深い結びつきは、当のノイマン型以来、久しくなかったことである。

Backusらが提案している関数的プログラミングは、Dijkstraの「構造的プログラミング」に始まるプログラミング・スタイルへの反省と結びついている。それとともに、これはScottなどによる「プログラムの意味論」の問題とも関連し合う。プログラミング・

スタイルから望ましくないとされる要素が、意味論構成上の難点とほぼ一致していることは興味深い。これは、「意味論」の自然さの傍証であろうし、意味論のスッキリしている関教的言語や「述語論理的プログラミング」言語を支持する一要素であろう。

この二つの「論理的プログラミング」は、一方で進められている形式的仕様の問題やプログラムの検証ないし合成(変換・導出)の問題と深くかかわるものである。また、これらのルーツの一つに「人工知能用語」があることも指摘しておくべきだろう。

プログラミングとともに、データベースが計算機技術の大きな核である。Coddらの関係データベースの理論は、独自の発展をしてきたとはいえ、上のようなプログラミングの側の動きと、いまや大きく重なり合うであろう。データベース用語とプログラミング言語は、これまでの所、別系統として存在しているが、両者に統一的な言語の基礎ができつつある。

データベースの問題はまた、人工知能における「知識ベース」構成と結びつけて考えられるようになってきている。人工知能研究は長い間、計算機技術の辺境とみられてきたが、いまや、その中枢部と深く関連づけて把える時期が来ていると見られる。

ここで視点を変えて「自然言語」をとり上げる。それをプロパーに扱うのは言語学であるが、ここでも興味深い現象が見られる。一つにはChomsky言語学の転回がある。それとともにMontague理論の出現がある。後者は、言語哲学から生れたものであるが、その構成法には実は、プログラム言語の意味論と強い近親性がある。それは、自然言語の「関教的」理論なのである。Montagueが発展させた「内包論理」は、プ

プログラム言語の意味論やデータベースの意味論に応用され始めている。「自然言語によるデータベース検索」や「自然言語プログラミング」を考えると、共通の理論構成は望ましいことである。

このような相互関連は、部分的にはすでに指摘されている。いまここで、その全体を考えてみると、どのようなストーリーが浮び上るであろうか。

かりに「自然言語マシン」を考えてみる。それは、「専用特殊」とどまるものではないであろう。それは、データベース、プログラミングを経て、データフローマシンにまで接続する。それは「論理マシン」であり(新しい構成の)コンピュータそのものである。

こういう観点に立って「第5世代」の問題意識を整理してみよう。

第1の問題意識は、

◎ 使いやすいコンピュータ・システム

ということになる。

日本語は、コンピュータとの自然な会話形態であるが、自然な会話(使いやすさ)ということからは、図形的会話ということも挙げておかなければならない。またコトバの入出力機能としては音声入出力も望まれる。これらの実現には高い機能を必要とする。そこで、

◎ 高機能化・多機能化・高性能化

が使いやすさの一つの条件であり、また問題意識として考えられるべきであろう。

高機能化や多機能化という観点からは、数値計算だけでなく、もっと広く「記号処理」の高度化と高能率化が必要になってくる。それはコトバの処理、ソフトウェア作製などにとって必要なことである。

高機能化にとって今後重要になる観点は、

◎ 知識情報処理

である。現在のコンピュータ技術では、問題解決のほとんどの所「プログラム」にしてやらなければならない。これがコンピュータの使い難さにも通じている。プログラムの自動作成は研究テーマの一つであるが、視野を広くとらなければ進歩は望めまい。第1には、ソフトウェア基礎論的検討がもっと必要である。第2には「知識」の組み込みが必要になってくる。問題の対象領域についての情報、さらにはそこでの法則性の情報を組み込む方向で、コンピュータの問題解決能力を一段向上させる必要がある。

これは、人間とコンピュータの相互了解の範囲を拡大することに相当し、コンピュータの問題解決能力を向上させるだけでなく、自然な会話にも必須の能力であり、「使いやすさ」を実現させる基礎である。

この方向の高機能化の観点からも、新しいコンピュータのつくり(アーキテクチャ)への要望やそれへのヒントが出てきている。

そこで、

◎ 新しいコンピュータ・アーキテクチャ

の問題意識が生じる。これは、

◎ 伝統的な体系からの脱却

という問題意識につながるといえよう。そこで、

◎ 新しい情報処理技術体系

ということが、使いやすさから出発した、総合的な目標といえよう。

そして、第5世代コンピュータ・システムとしては、全体として

◎ 知識情報処理システム

というイメージが浮かび上がってくる。

次に、このイメージを実現するために、その過程において、

◎ システムを作りやすい高度の環境

が不可欠であるという問題意識が生じる。

これはゴール実現のための手段であると同時に、単なる支援システムでなく、進化するシステムとしての知識情報処理システムの要素として重要視されなくてはならない。それ自身、知識情報処理のプロトタイプ役割も果たすのである。これはまた、人材育成(我々国では、先進的な研究者、技術者の数が米国等に比してあまりにも少ない)の環境として、層の厚い技術として知識情報処理を育てる土壌となる。

以上のような設定は先進的にすぎるように見えるかもしれない。しかし、従来技術の漸進的改良の線上に難問の解決策が見えず、それが新世代への悲観論を生んでもいるが、それは現行技術のある種の成熟現象と見ることができ、むしろ、新世代への機が潜在的に熟していると読むこともできよう。一方これは、これまでの先進的研究の延長上にあり情報処理技術の必然的な方向と見こともできる。タイミングの問題は別にして、要は立ち止まるか、前進するかにあると思われる。



③ 「第5世代コンピュータ・アーキテクチャ」

慶応義塾大学理工学部電気工学科教授 |
相磯秀夫

講演要旨

第5世代コンピュータの開発に際して、アーキテクチャ研究の必要性、新しいアーキテクチャへの期待、技術的観点からのとらえ方、およびそのための主要な研究課題について概要を述べる。

◎ アーキテクチャ研究の必要性

本格的なコンピュータが誕生してから30有余年、コンピュータをとりまく技術もようやく定着し、応用分野も急速に展開しつつある。その結果、種々の応用に対して適応する新しい機能を備えたコンピュータの開発を求める声が強くなってきた。しかしながら、新しい機能の実現を実用レベルで達成するためには、アーキテクチャの抜本的な変革が不可避に思われる。

新しいコンピュータを開発する主な理由が従来のように、単に価格性能比の改善にあるならば、ソフトウェア開発の困難さ、膨大なソフトウェア資産の有効利用、LSI化に伴うハードウェア設計、開発の制約などを考えれば、新しいアーキテクチャの大幅な導入など考えずに、ハードウェアのVLSI化に徹した方が得策なことは明らかである。それにも拘らず、最近新しいコンピュータ・アーキテクチャの考案が必要とされる理由は、

- (1) 大量データに対する高速処理の実現
- (2) ユーザとコンピュータの間に存在するセマンティック・ギャップの解消
- (3) 高インテリジェンスをもつマンマシン・インタフェースの完備
- (4) 新しい応用分野で要求される効果的な支援機能の実現
- (5) 人間による問題解決手法に見られる非決定性デー

タ処理機能の実用化

- (6) 高信頼性システムの実現
 - (7) ハードウェア、特にVLSI技術の有効利用
 - (8) 高機能パーソナル・コンピュータを核とする新しいスタイルのデータ処理システムの実用化
- などの必要性にあるものと考えられる。このうち(1)~(6)はソフトウェア開発の生産性改善あるいは新しい応用からアーキテクチャに期待するニーズであり、(7)及び(8)は技術進歩から出てくるニーズと見ることが出来る。

◎ 現在のコンピュータの問題点

現在のコンピュータがかかえる主な問題点をまとめれば、

- (1) 機能ならびに構造が単純で、固定的であり、種々の応用に対して適応性が乏しい
- (2) 数値データを対象にした決定性データ処理方式に基づいており、新しい応用に十分対処できない
- (3) ユーザが使える言語の抽象化機能が貧弱で、問題を解くための意味変換が容易でない
- (4) 人間の経験あるいは知識を収集・蓄積し、それを活用しながら学習する機能をほとんど持たないことなどであろう。見方を変えれば、今までのコンピュータの開発は比較的量的な改善に主眼があったが、これからは質的な改善が重要視されるものと思われる。特に(4)に関連して言えることは、現在のコンピュータは最大の問題点は論理的な能力(logical power)が極めて弱いことである。この問題の解決は“コンピュータはどこまで人間に近づきうるか”という究極的な研究課題に帰着することになる。

◎ 期待される機能

第5世代コンピュータの応用分野は、事務処理・科学計算・公共事業・産業・制御・宇宙航空・民生など広範にわたるが、これらの分野において今後重要な問題になると考えられる課題は結局のところ、知識を取扱う基本的な知的機能をアーキテクチャ・レベルでいかに実現するかにある。このような知識を扱えるコンピュータ・システムを知識情報処理システム(KIPS)と総称している。

KIPSに共通的に期待される機能は、

- (1) 問題解決・推論機能：100～1G推論操作/秒
(1推論操作=100～1,000機械命令)
- (2) 知識ベース管理機能：100～1,000GBのデータベース上に構築、数秒内で問合せ可能
- (3) 知的インタフェース機能：音声・画像・自然言語などのためのマンマシン・インタフェース

と見ている。これらの機能が有効に働く具体的な基本応用システムは機械翻訳・質問応答システム・音声応用システム・図形画像応用システム・意志決定支援システム・知的CADシステムなどであるが、これらは上述の代表的応用分野において重要な位置を占めると考えられる。

◎ 新しいアーキテクチャのとりえ方

コンピュータ・アーキテクチャにとって重要な課題は、

- (1) VLSIで代表されるハードウェア素子を積極的に活かすVLSIアーキテクチャ
- (2) 新しい応用から要求される知識ベースを効率よく処理する新アーキテクチャ
- (3) アーキテクトの要求である実現手法から望まれる分散処理アーキテクチャ
- (4) 応用分野にそくして、一つのシステムにまとめあげるシステム技術、特に仮想化のためのアーキテクチャ技術

であろう。

KIPSを具体的に実現するためのアプローチをトップ・ダウン的な考え方で示せば、

- (i) 応用に対応した仮想マシンの設定
- (ii) 仮想マシンの構成要素(仮想プロセッサ、仮想データベース・マシン、仮想ネットワーク・マシン)の設定
- (iii) 各構成要素を具体化するための専用マシン群の設計
- (iv) 専用マシン群による機能分散型実マシンの構成
- (v) VLSI素子による実マシンの実現

というステップになろう。

第5世代コンピュータ・システムを別のとらえ方をすれば、次に一例を示すように、3つの機能レベル・コンピュータから成る分散処理システムと見ることができる。すなわち、

- (a) レベル1(パーソナル・マシン)：現在の大型機なみの性能を備え、プログラム生産性向上ならびに図形・画像・音声などの使い易いマンマシン・インタフェースの支援などを配慮した個人用の高機能コンピュータ、処理能力は3～5MIPS、主記憶1～16MB、ファイル記憶100MBをもち、自然言語も使える。
- (b) レベル2(ユーザ・マシン)：パーソナル・コンピュータの能力を越える処理あるいは他のマシンやコンピュータ・ネットワークを管理する大型機能分散型コンピュータ
- (c) レベル3(専用マシン)：多数のユーザが共用する問題向き専用コンピュータ、ユーザの共通の仕事場となるデータベースあるいは知識ベース・システムを管理する専用マシン

これらのマシン群は統一された高レベル論理的インタフェースによって接続され、ユーザには一つの論理システムに見せる仮想化機能、システム内の資源を効率的に利用するための機能、あるいはユーザに便利な種々のサポート機能を含んだ制御機構の下で管理されると予想している。

◎ 主な研究課題

第5世代コンピュータの開発に際して、研究すべき主な課題を示せば次のとおりである。

- (1) 第5世代コンピュータ核言語

KIPSの概念構造は知識ベース・マシン、問題解決・推論マシン、知的インタフェース・マシンとから成るが、KIPSそれ自体は高度なソフトウェア体系と見ることができる。したがって、現在のソフトウェアに関連する諸問題を解決する必要があるが、その解決策の代表的なアプローチとして新しいプログラミング言語の研究がある。第5世代コンピュータ核言語はハードウェアとソフトウェアのインタフェースであり、アーキテクチャのマクロな仕様になると同時に、ソフトウェア作りのツールにも成りうる重要な位置を占める。核言語としては、並列処理・データ抽象化・オブジェクト化の概念を含む述語論理型プログラミング言語(PROLOGなど)を考えている。

- (2) 問題解決・推論マシン

核言語をサポートする新しいアーキテクチャを研究

する必要があるが、核言語の性格から特に、並列処理及び推論機構の徹底的な追求がベースになると考えられる。そのような意味では、従来のノイマン型コンピュータの改善の上に成立つ高レベル言語マシンの開発、データフロー・マシンおよび高度なパターン・マッチング機構、VLSI 指向マシンの研究が主要課題となる。

(3) 知識ベース・マシン

知識ベースを構築するためには、先ず大規模で効率のよい関係データベース・マシンを開発する必要があると考えている。したがって、パイプライン処理・並列処理・VLSI 技術を駆使した関係データベース・マシンを問題解決推論マシンのメモリ管理システムとして開発する。加えて、構造メモリ、インテリジェント・ディスク、マルチプロセッサに関する研究を行い知識ベース・マシンの開発・実験を試みる。

問題解決推論マシンと知識ベース・マシンとは共通機能が多い。両者は演算処理能力の高速化および比較的小容量データを対象にしているが、後者は大規模な関係データベースの集合演算の高速化を主要課題としている。KIPS のプロトタイプは両者を結合した形で実現できると見れる。したがって、データフロー・マシンと関係データベース・マシンのあり方を KIPS の基礎課題として徹底的に研究する。

(4) 知的インタフェース・マシン

文字・図形・画像・音声など種々の情報メディアに対する高度なマンマシン・システムの開発と実用化を研究する。

(5) VLSI およびシステム・アーキテクチャ

第5世代コンピュータ・システムの実現にはVLSI 技術が不可避であることは言うまでもない。そのために、VLSI 技術を十分活かせる体制を構築することが大切である。先ず、アーキテクチャ・機能モジュール・論理・回路・レイアウト・テストパターン等の各設計レベルで用いる統一した階層記述言語およびシミュレータの研究開発を行う必要がある。また、VLSI システムやコンピュータ・アーキテクチャ設計のための知識ベース・システムを構築することも重要な課題と考えている。

第5世代コンピュータ・システムを研究するためには大規模なソフトウェアを開発する必要がある。そのために、強力なソフトウェア開発サポート・システムを準備することが前提になる。したがって、広域ならびに構内コンピュータ・ネットワークの設置ならびにソフトウェア開発用端末として高機能パーソナル・マ

シンの開発を行う。高機能パーソナル・マシン自体も核言語と高度なマンマシン・インタフェースを備えた問題解決推論マシンである。実際のソフトウェア開発サポート・システムの開発に際しては、分散処理方式、分散データベース、仮想化ならびにシステム評価に関する基礎研究が大切になると考えている。

質疑応答



元岡 ただいまから、以上3件の講演についてのご質問をお願いしたいと思います。質問される方はそのフロアのところのマイクで、名前、所属をおっしゃってから質問をして下さい。どうぞ遠慮なく……。

トレリーバン ニューキャッスル大学のトレリーバンです。3つの質問があります。最初の2つの質問ですが、どうして論理型言語を選んで核言語にされたのでしょうか。またどうしてデータフロー・アーキテクチャをコア・アーキテクチャとして選ばれたのでしょうか。

淵 あるレベルの言語としてロジックがあるのは当然ではないかと思えます。問題は、プログラムを表現するのに適当であるかどうか、あるいはその他のことがらを表現するのに適当であるかということだと思いますが、例えばPROLOGとかその他の言語を調べると、その辺が適切なレベルではないかと思ったわけです。

それから、ロジカルな言語として考えている1例としてPROLOGをあげたわけですがLISPもロジックの1部だと思いますし、やや広い意味でとらえたいと思います。その辺で答えになっているかどうかわかりませんが、ロジック以外のどういう言葉を提案されるかということの後でコメントしていただきたい……。

それから、データフローでなくてもよいと思いますが、ファンクションとカリレーションの中には非常にたくさんの並列性が含まれており、そういうものをマシンとして実行するような機構を考えると、データフローの考え方というのは1つの非常によい候補ですが、それ以外のものがないとは言えません。それから、相磯先生はソフトウェア工学のことに言及せよ

とおっしゃるわけですが、ロジック・プログラミングを選んだのには、結局ソフトウェア工学の技術体系というのは仕様記述も検証も、あるいは変換もこれまでのところは普通のロジックの上に立っているということがあると思います。質問の意味をもう少し詳しく説明していただくと、別のコメントができるかと思いますが。

カイファン 私、相磯先生に伺いたいのですが5Gシステムは何の略か教えていただけますか。また、なぜ5Gと呼ばれているのか教えていただけますか。

相磯 別に略語ではなく、我々の呼称でございます。先ほどちょっとお話し申し上げましたように、まずVLSIのCADがのること。もう1つは、ソフトウェアの開発をそれをもってサポートするというようなシステムをまず最初に構築する。そのシステムを5Gシステムと呼んでいます。それだけです。もちろん最終的にはKIPSの上でVLSI-CADも実行されるし、ソフトウェアの開発もするということになると思います。従いまして1番最初の5Gシステムは既存の大型コンピュータを使うということになると思います。

ローソン スウェーデンの大学から参りました。2つ質問させていただきたいと思います。1つは3人の先生方に伺いたいのですが、ジェネラルな質問として、何回かマン・マシン・コミュニケーションという言葉が出て参りましたが、それをメディアでやるというお話がございましたが、将来のコンピュータがどういう形になるかは別にして社会として人間がグループではなく個人として働くような社会が出てくるのではないかと思います。私がここで提案したいのは、我々はグループ・マシンのコミュニケーションをサポートし

ているのか、あるいはマン・マシンのコミュニケーションをサポートしているのかということを考えて行きたいと思います。つまり、将来つくって行くものが1人の人の製品ではなくて何人かの協力によって出てくるものであると思いますので、むしろマン・マシンというよりグループ・マシンのコミュニケーションと考えるべきではないかと、私は考えるわけでありまして。ここが非常に重要な目標ではないかというのが1つ目です。

2つ目の質問は、まず1つ目の質問のお答をいただいてからにしたいと思います。

相磯 それでは私の考えをお話し申し上げます。特に私が今申し上げたのは、非常に重要な基本的な研究課題であります。その成果を何か目的とするアプリケーションにのせた場合に全体のシステムになるのではないかと考えているのですが、その辺の応用システムについてはあまりお話し申し上げませんでしたので、ローソン先生のご質問が出たのだと思います。ですから、機械翻訳システムとかオフィス・オートメーション・システムとかいうのを第5世代の研究で構築するかどうかという意味でしょうか？

ローソン そういう意味ではありません。私の質問は、将来に向かってアプリケーションを開発して行く場合に、今までは仕事を分けその分業に合わせて端末を与えてプロジェクトの一部を分担してそれぞれ開発してきたわけです。ところが、こういうグループが一緒に仕事をするというためのデバイスというのは提供されてこなかったのではないかと思います。つまり、いろいろ便利な形でベーシックなアルゴリズムを使ってみるというようなことができるようなデバイス、あるいは1人の人のつくったアルゴリズムは他の人のつくったアルゴリズムとどのような相互作用を持っているかといったようなことが調べられるようなデバイスがつけられてこなかったと思います。したがって今後、2~5人のグループで研究して行く場合にそういうタイプのものが必要ではないかというのが、私の考え方です。

相磯 わかりました。我々もそういう意味ではみんな設計ができるようにして行きたいと思っております。したがって先ほどの5Gシステムなどはそっちの方向で開発をして、いろいろな研究者がまず協同で仕事ができるような環境をつくることをするのではないかと考えております。もちろん、今のローソン先生のご質問はもう少しアドバンスしたご質問ではないかと思っておりますが……。

淵 どのチャンネルが通訳のチャンネルかわからなかったもので、ご質問を正確にとらえてないかと思いますが、小さなグループとか大きなグループがいろんなレベルで協力し合う、あるいは別々に働くという、新しい情報処理体系をつくるのは当然の前提だと我々は考えております。そういう新しい使い方というのは10年後に始まるのではなくて、既に始まりつつあり、そういうものをすべて取り入れた上で更に新しいものがその上にのらないだろうか、という議論が主体になっていると思っておりますが、ご質問の意味とだいぶ違うでしょうか。

ローソン 完全に私の伺いたかったとおりというわけではありませんが、相磯先生が言われましたようにもう少し進んだところまで考えていたのですが、2つ目の質問はロジック・プログラミングをベースとしてどうして選ばれたかということなのです。つまり、アルゴリズムティックな考え方というものをしてきたわけですが、将来のマシンを考えるとこういう新しい考え方が出てくると。今までにつくってきた複雑なマシンについてもそういう考え方が出てきたわけです。ですからアルゴリズム的な考え方というものを全くやめてしまって、ファンクショナルなアプローチに切り替えるのが果していいのかわかるかということに少し疑問を持っているわけですが、これはあくまでも質問というよりコメントでありますので、何か逆のコメントでもありましたらいただきたいと思っております。

淵 アルゴリズムをとらえるのにいろんなレベルがあるかと思っております。1つは現在のマシンのストラクチャを前提にしてどういふものがアルゴリズムであってそれをどうつくるかという問題があり、それは現在非常に大きな問題になっています。2番目は、もっとアブストラクトにアルゴリズムをとらえることが可能であるわけで、それは論理的なフレームワークの中でとらえ得るのではないかと考えています。ですから、アルゴリズムにも2つの側面があって、1つは現在のテクノロジーに依存したアルゴリズムの研究と、それから少し独立したもっと本質的なアルゴリズムがあり得ると思っておりますが、いかがでしょうか。それで、ロジックを選んだ理由の1つはコンピュータ・アーキテクチャにとって大きな自由度を与え得る。要するに、新しいアーキテクチャを考えるためのかなり広いフレームワークを与え得るのではないかということなのです。

フロア外人 私はコンピュータ・アーキテクトに自由度を与えるということが、それほど意味があるとは今のところ思いませんが、この会議をずっと聞いてみ

るともう少しよくわかるかもしれませんが、その時点で伺いたいと思っています。

カステロ こういう開発は日本だけで行われるのでしょうか。あるいは国際的な協力が行われるのでしょうか。あるいは企業を入れた形で協力が行われるのでしょうか。

元岡 バネリストに対する質問ではないので、私がお答えいたします。まだ、どういう体制でやるかについてははっきりしたイメージはできておりません。この会議を開いた目的の1つは、どういう体制が望ましいか、どういう体制が国際的にとり得るかということと皆さんと一緒に議論し、その中から見つけていきたいということです。

サボリー スチュワード・サボリーと申します。ドイツのコンピュータ会社の者です。私、さまざまなソフトウェアパッケージの話の聞いたことがあります。これは関係データベースあるいはKnowledge Base というようなシステムとしては話が出ていないようです。磁気ディスクのパフォーマンスレベルというようなものもあると思うのですが、どのソフトウェアがどれに相当するということに関係がなく、ハードウェアとして今ここにある以上のものが必要であるということをはっきりしていると思います。ですから、スライドのところにいろいろ数字が出ていたことを少し説明していただけますでしょうか。

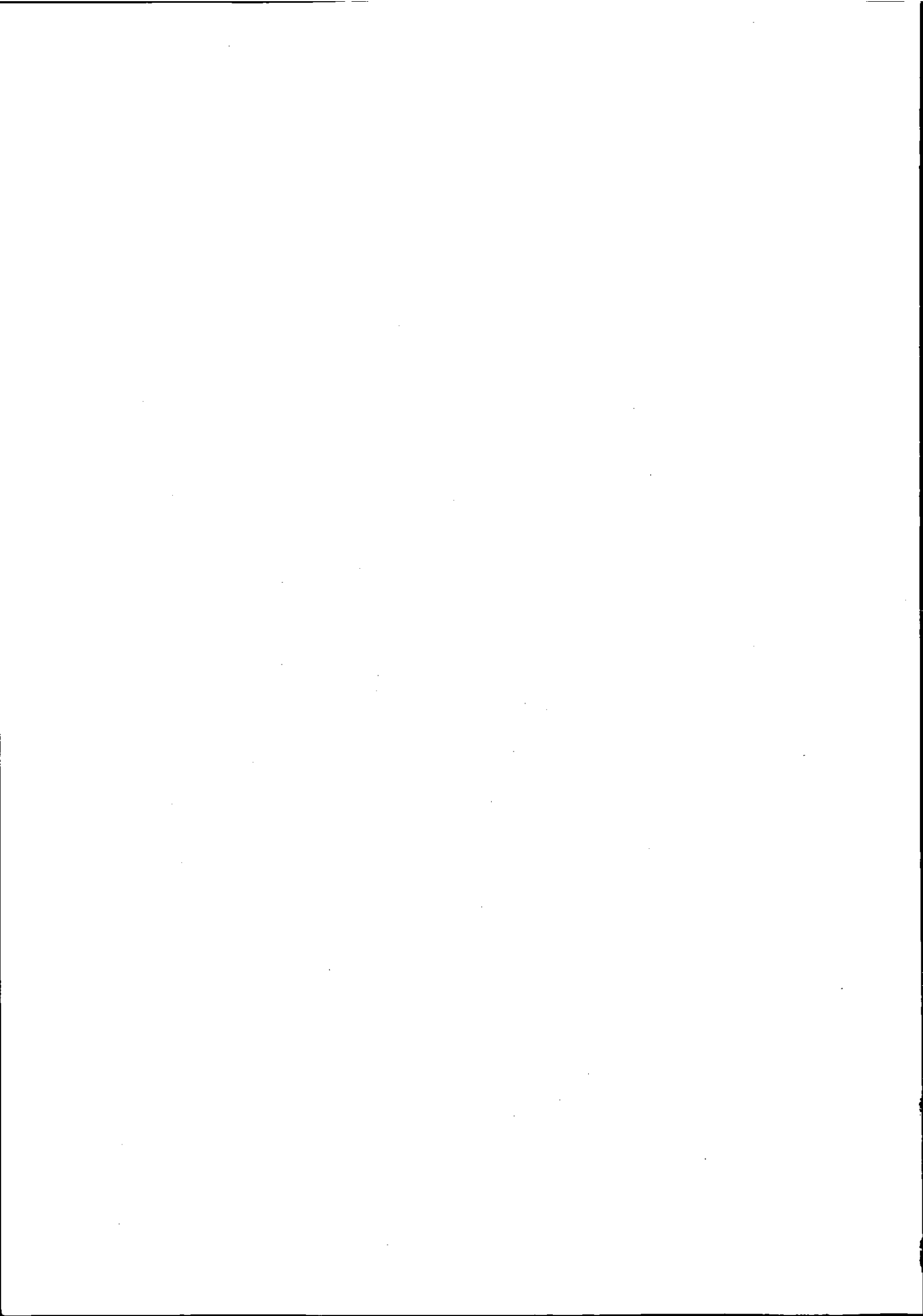
淵 たぶんLIPSという風変りな単位についてのご質問かと思いますが、出ている数字は10年後につくり得る単体のコンピュータとしてのマキシマムな数字を想定しているのです。ですから、ある意味でスーパー・マシンの数字を想定しているわけです。アプリケーションのサイドからして本当にそれだけ必要なのか、充分すぎるのか、あるいは足りないのかということについてはまだちゃんとした答えはないのですが、あの数字自体はかなりがんばった上でのハード的な目標の数字と見ていただきたいと思います。

曾和 群馬大学の曾和でございます。相磯先生のお話で、次の世代のコンピュータを研究する上でノイマン型のコンピュータをよく調べなければならないということがありましたが、私も50~60%はそういう必要があるだろうという気はしているのですが、そういうことに関してその次の世代からもう1つ次の世代かもしれませんが、そういうコンピュータに関しては何かフォン・ノイマン型のコンピュータを越えた飛躍が必要ではないかと思っています。その場合にノイマン型を調べるというのは、そのノイマン型の、例えば

どうしてもできないようなことを洗い出してその次のことを考えるのか、それとも何か現在行われている中で次の世代へのヒントになるようなことがたくさん含まれているのか、その辺のところははっきりしないのですがどうなのでしょう。

相磯 ノイマン型コンピュータの定義などはたいへん難しく、私にはなんとも申し上げることができません。ノイマン型を越えるということがどういう意味なのかよくわからないのですが、恐らく現実的には今のコンピュータがもつ問題点を解決しなければ、新しい機能はたぶんのらないのではないかと私は見えています。そういう意味で新しい推論マシンあるいは知識ベース・マシンのようなものを開発するにあたっては、そういう基本的なところもきちんと研究しなくてはいけないと、私は見ているのです。

元岡 それでは、まだご質問もあるかと思いますが、時間が参りましたのでこれでやめさせていただきます。



II. 10月20日(火) 9:30~17:30

1 知識情報処理・研究計画

(1) 問題解決・推論機構

電子技術総合研究所 古川 康一 他

(2) 知識ベース機構

電子技術総合研究所 諏訪 基他

(3) 知的マンマシン・インターフェイス

電子技術総合研究所 田中 穂積 他

(4) ロジック・プログラミングと専用高性能パーソナル・コンピュータ

電子技術総合研究所 横井 俊夫 他

2 招待講演

(1) 第5世代コンピュータ・システムにおける記号操作の革新

スタンフォード大学 エドワード A. ファイゲンバウム

(2) 論理的プログラム合成

ミュンヘン工科大学 ウルフガング・ビーベル

(3) 記号演算の展望

国立情報処理オートメーション工学研究所 (INRIA)

ギルス・カーン





講演要旨

1 はじめに

1990年代における情報処理の主たる応用分野は、知識情報処理となることが予測されているが、そこで中心的な役割を演ずるのは、これから述べる問題解決・推論機能である。これらの機能の大部分はソフトウェアによって作られ、また、第5世代コンピュータのソフトウェアの中核を占めるものとなる。

問題解決・推論メカニズムに関連した研究課題は、

- ① 第5世代コンピュータの核言語(FGKL)の設定
- ② 知的プログラミング・システム、知識表現言語、メタ推論システムなどの基礎ソフトウェア・システムの開発

の2つである。

以下に、これらの研究課題についての研究開発計画を簡単に紹介する。

2 研究開発計画

(1) FGKLの概要

FGKLは、論理型言語PROLOGに近いものである。FGKLは3つのphaseに分けて開発される。第0版は、PROLOGの拡張版を考えている。拡張は、モジュール化機構、メタ・ストラクチャおよび関係データベース・インタフェースである。

モジュール化機構は、論理式の項(term)の世界にデータ型を導入してmany sorted logicとして実現することを考えている。少なくとも関数として働く部分については、そのような扱いが可能である。一般の

II.1 知識情報処理・研究計画

(1) 問題解決・推論機構

- ・ 電子技術総合研究所 古川 康一
- 京都大学 中島 玲二
- 東京工業大学 米澤 明憲
- 日本電信電話公社 後藤 滋樹
- ㈱日立製作所 青山 明夫

述語については今後の検討が待たれる。

一方、副作用がなく、かつ構造全体をコピーしなくとも良いデータ構造およびメモリ・システムの開発が望まれる。F-treeは、そのようなメモリとして、最近提案されたものである。

メタ・レベルの推論能力は、知識の獲得や帰納的推論などのより高度な機能を実現するためにはなくてはならないものである。

我々は、2つの有用なメタ・レベル推論機能を提案する。その1つはモダリティで他の1つはメタ・コントロールである。モダリティは、コンテキスト機構、多世界表現、フレーム構造などの人工知能研究で明らかにされた諸概念を実現する基礎を与える。メタ・コントロールは、問題解決の制御や、プログラムの並行処理の制御に利用されるであろう。

関係データベースについてのインタフェースは、

- ① 言語レベルでのインタフェース、
- ② 並列実行モードでのインタフェース、
- ③ 意味レベルでのインタフェース

の3つが考えられる。これらのインタフェースは、順を追って開発される。すなわち、①は第0次核言語で、②は第1次核言語で、③は、第2次核言語で、それぞれ実現されるものと考えている。

つぎにFGKLの並列実行について述べる。FGKLを論理型言語と考えると、それはand/orグラフの探索問題と考えられる。and/orグラフの探索を並列に行うには、いくつかの方法が考えられる。それらは、基本的にはつぎの3つである。

- ① and節を流れるストリング型データのパイプ・ライン的実行。

② or node の並列実行

③ and node および or node の並行実行と、
関係代数演算の結合。

①は最も現実的で容易なアプローチである。②は、
多くの場合についての証明が同時に進行するので、そ
のメモリ管理が必要である。③は、論理型プログラム
を代数型プログラムへ変換する処理が必要である。さら
に、これらを、データフロー・メカニズムに落とす
作業が必要である。データフロー・メカニズムは、い
まのものに集合操作機能を追加しなければならない。

(2) 知的プログラミング・システム

知的プログラミング・システムの研究では、モジュ
ラー・プログラミングと自動プログラミングが2つ
の大きな研究のターゲットである。

モジュラー・プログラミングは、論理型言語上では、
未開発の技術である。そこでの新展開が必要であるが、
より強力なシステムの実現が期待できる。

自動プログラミングの研究では、プログラム自身の
機能・性能を記述するメタ言語の開発が必要である。
それから、知識工学的アプローチによるモジュールを
ベースにした自動プログラミング・システムの開発を
目指す。そして、既成のモジュールを蓄積・管理する
アルゴリズム・バンクの開発も合わせて行う。

(3) 知識表現言語

知識表現言語については、ここでは、それが核言語
上で実現されるべきものであることのみ強調しておく。
例として、PROLOG で記述したプロダクション・シ
ステムを挙げておく。

(4) メタ推論システム

メタ推論システムは、人間-機械、あるいは機械同
士の間の意味レベルでの相互理解を助けるものと考え
る。それらは、自然言語および人間の思考の状態、あ
るいは機械言語および機械の状態を理解できなければ
ならない。

我々は、メタ推論システムによって、つぎのような
問題の解決をねらっている。

- ① 知識の獲得
- ② 問題解決の制御
- ③ 信念の翻意
- ④ 会話の制御
- ⑤ プログラムの半自動修正
- ⑥ 外部データベースへのアクセス
- ⑦ 外部データベースの再構成

質疑応答

フロア外人 私、質問があります。CLUのよう
な言語では、プロセッサ・ランゲージのフレームワー
クの中でモジュール化、プログラミングとかデータ抽
象化とかあるいはシンボリック・エグゼキューション
などをうまくやることができると言われていますが、
そういう中でどうしてこういういわゆる手続き型言語
を全く捨てて、関数型プログラミングに力点をおかれ
たのでしょうか、その理由を説明していただけますか。

古川 手続き型言語でデータ抽象化とかモジュラー
・プログラミングというものが見通しが立っておりま
す。ここではもう一つのねらいとして知識情報処
理ということを非常に重要に考えており、むしろ
プログラミング言語の機能としてはそちらの方が
必要でしょう。その結果、関数型言語やあるいは
そちらの方にはまだデータ抽象化であるとかモジ
ュラー・プログラミングの研究開発というのは
進んでいないが、それは追ってやる、あるいはでき
ると考えています。

トレリーバン トレリーバンなのですが、かなりの
人がPROLOG の改善をやってきました。そのために
はコントロールマークを言語の中に入れたようです。
これは言語のパワーを減じることになるが、同じよう
なルートをたどられるつもりでしょうか、あるいは全
く別のスキームを何か考えているのでしょうか。PRO-
LOG の改善方法について伺います。

古川 今のご質問はPROLOG を拡張することによ
ってPROLOG のフレキシビリティあるいはユニバー
サル・プロパティが損なわれるというご質問ですか。
そしてわれわれはどういう方向を模索しているかとい
うのがご質問ですか。

むずかしい問題だと思いますが、一つのわれわれの
観点はPROLOG・FGKLですが、それをプログラミング
言語であると考えているところです。LISP がたと
えばアレイを導入したり、あるいはいろんなものを導
入してPure LISP からブラクティカルなLISP シス
テムになったように、PROLOG Based Language
もそういう経過をたどるであろう。今はそれはあまり
整理されてない形でいろんな拡張がなされているかも
しれないが、そういう拡張というものは行うべきであ
ると考えています。できればそれを非常にきれいなフ
レームワークで拡張したい。

トレリーバン

どうもありがとうございました。推測で結構ですが、どれぐらいのマンパワーが最終段階に到達するまでにかかると考えておられるでしょうか、あくまでも推定で結構です。大変答えにくい質問とは思いますが、だいたいの見積りで結構です。先ほど示されたスケジュールに大変関心がありますので、それを少しでも定量化したいので、だいたいかかるマンパワーを教えてくださいませんか。

古川 それはスケジュールの期間によって違ってくるというふうに考えています。初めの期間ではかなり見通しのいい計画になっています。そこで開発と併せて基礎研究もやらなければならないわけですが、正確な数字はよくわかりませんが最初は100人ぐらい、それから3年後で300名ぐらいですか。

淵 今の質問はまた最後のパネルでも議論になるかと思いますが、後一つぐらいご質問とかコメントございましたら。

それでは一つだけ一番最初のご質問に私から一言コメントしたいのですが。

データ抽象化とかそういう考え方というのは、私、本質的には非手続き的なものだと思うので、手続き型言語に埋め込むということは一番いい方法とは思いません。

それでは次の報告に移りたいと思います。



講演要旨

1. はじめに

(1) 発表の範囲

第5世代コンピュータシステムは知識情報処理システムの構築を目標としたもので、次の3つの部分から構成される。

- ① 問題解決と推論メカニズム、
- ② 知識ベースメカニズム
- ③ 知的マン・マシン・インタフェイス

本発表はこの内知識ベースメカニズムに関するものである。(プロシーディング図1の太枠内に対応)。

(2) イメージ

本プロジェクトで狙う知識ベース・メカニズムの目標像は協調型知識ベースシステムである。

2. 協調型知識ベースシステム

(1) 概要

協調型知識ベースシステムは、異なった種類の知識源を持ついくつかの分散したエージェントが協力しあいながら問題を解くというモデルに基く。

(2) エージェント

協調型知識ベースシステムの各エージェントは次の情報を持っている。

- ① 自身の受持ち分野に関する知識
- ② 誰が何を知っているか

しかしながら、各エージェント自身は、問題解決のための完全な戦略は持ち合せない。

(3) 必要性

協調型知識ベースシステムの必要性は次の3点にあ

(2) 知識ベース機構

電子技術総合研究所	諏訪	基
電子技術総合研究所	古川	康一
㈱富士通研究所	牧之内	顕文
三菱電機㈱	溝口	徹夫
東京理科大学	溝口	文雄
沖電気工業㈱	山崎	晴明

る。

- ① 扱う問題が複雑になるにつれ、多種類の知識源を利用する必要が生じる。従って、分散して存在する知識ベースを協調的に利用しなければならない。
- ② 知識が大規模になり、1個の知識ベースにまとめることは不利になる。
- ③ 知識ベースは各領域毎に個別に開発される。

3. 開発ステップ

知識ベース・メカニズムの開発ステップは次のとおり。(プロシーディングス図2参照)

- ① 前期：単一知識ベース管理システムの設計並びに開発。
- ② 中期：分散知識ベース管理システムの設計並びに開発。
- ③ 後期：協調型知識ベース管理システムの設計並びに開発。

4. 知識情報処理システムの必要条件

(1) 要請

第5世代コンピュータプロジェクトで狙う知識情報処理システムは、まず有用でなければならない。次にその性能は高度に知的でなければならない。そしてシステムを構築するのが容易であるような工夫が必要である。

(2) 開発戦略

そのようなシステムを開発するために次の3つの戦略が必要。

- ① 専門家レベルの知識を獲得し、

- ② 強力な推論メカニズムを開発し、
- ③ かつ外部あるいは既存の知識源を有効利用する技術を開発する。

(3) 課題

本プロジェクトにおける知識ベース・メカニズムの開発課題として次の4テーマを設定する。

- ① 知識表現言語と設計サポートシステムの設計。
- ② 知識獲得技法の研究。
- ③ 分散問題解決手法の開発。
- ④ 外部知識ベースのための設計手法の開発。

5. 知識

第5世代コンピュータ・システムで扱う知識には、オブジェクト知識とメタ知識とがある。

前者には、

- ① マンマシンコミュニケーション用言語に関する知識、
- ② 問題領域に関する知識、
- ③ マシンに関する知識

が含まれる。

メタ知識とは知識に関する知識、あるいは知識の使い方に関する知識である。

6. 研究課題の内容

- (1) 知識表現言語と設計サポート・システムの設計。

知識表現言語の枠組として、例えばプロダクションルール、フレーム構造、セマンティック・ネットなどがすでに考えられているが、まずそれらを十分に検討し、第5世代コンピュータ向き知識表現言語を開発する。その特徴としては、

- ① モジュラリティを有し、
- ② 宣言的記述と手続的記述の融合、
- ③ 高階論理の取扱い、
- ④ 多世界表現

が可能であることを狙う。

設計サポートシステムとして、次のサブシステムの開発を目指す。

- ① 知識ベース・エディタ、
- ② 知識ベース・デバッガ、
- ③ 知識獲得サポートシステム、
- ④ 無矛盾性チェッカ、
- ⑤ 知識ベース・コンパイラ。

- (2) 知識獲得

知識獲得システムの課題には、知識表現、知識の構

造化手法、メタ知識の獲得等がある、知識獲得はコンピュータの専門家ではなく、問題別の専門家が対象であるから、それを考慮する必要がある。

研究の進め方については、事例研究により手法を学ぶのがよい。それには2つのアプローチがある。1つは、プログラミング技術の観点から、どういう知識をどのようにプログラムは取り込む必要があるかを第一意的に考えて進める方法。もう1つは、人間の思考や推論のモデルを十分に考慮に入れながら進める方法である。

知識獲得システムの開発では、先に述べたように対象がコンピュータの専門家ではない点に留意しなければならないが、知識ベースの保守、知識獲得のためのガイドの手法の検討も行う。

- (3) 分散問題解決

協調型知識ベースシステムという目標を達成するには、

- ① 各エイジェントのふるまひは前もって十分には記述されておらず、従って計画立案を必要とし、
- ② 各エイジェントが問題解決のパワーを持たなければならない、

という点を考慮し、分散問題解決手法の開発を行う。

- (4) 外部知識ベースの利用

外部知識の構成は、関係モデルをインターフェースとした関係データベースを想定する。

関係代数をインターフェース言語に設定する理由は、

- ① 関係モデルは5G核言語(FGKL)と整合性が良い。
- ② 既存のデータベースが利用できる。
- ③ 関係代数マシンがその内に開発されるであろう。の3点を挙げることができる。

関係データベースを次の点で拡張して応用することを考える。

- ① マルチプル・ビュー、
- ② 知的問合せの最適化、
- ③ 内部構造の自動再構成。

図形・画像データベース及びテキストデータベースも開発する。

7. 研究開発のスケジュール

- (1) 知識表現言語及び設計・サポートシステム

前期：知識表現言語(FG-KRL)及びサポートシステムの実験版の開発。

プロダクションルールとフレーム構造を採用。

中期：知識表現言語(FG-KRL)の第1版開発。

分散知識源を対象。

(2) 大規模知識ベース

前期：大規模知識ベース (FG-LSKB) の実験システム開発,

関係データモデルを採用。

中期：FG-LSKB の第 1 版作成,

推論機能を持つ知識ベースの開発。

(3) 後期目標

協調型知識ベースシステムの開発。

質疑応答

シンドラ この質問がもし簡単すぎたら簡単すぎるとおっしゃっていただければ結構です。ただ1日半いろいろお話を伺ったのですが、まだ知識ベースの中身がわからないのです。この知識ベースがたとえばデータベースとどのように違うかとか、あるいは他の形で知識ベースの中身を説明していただけますか。

諏訪 知識ベースの中身を説明するのに最も適した方がたまたまこの会場にいらっしゃいまして、それはやはりスタンフォード大学のファイゲンバウム教授だろうと思います。教授はきょうの午後の Invited Lectureでそのお話をしてくださると私は確信しておりますので、その時に適切な答が返るのではないかと思います。

淵 そういう答えでよろしいでしょうか。



(3) 知的マンマシン・インタフェース

- 電子技術総合研究所 田中穂積
- 日本電気㈱ 千葉成美
- 東京芝浦電気㈱ 木戸出正継
- 電子技術総合研究所 田村秀行
- シャープ㈱ 小寺次夫

講演要旨

ファイゲンバウム教授の説明にもあった様に、知的インタフェース(Intelligent Man-Machine Interface)に対しても困難な問題は山積しており、したがってイノベーションが必要である。それにこれから10年間挑戦することは十分意味のあることだと考えている。

知的インタフェースの研究は、(1)自然言語処理、(2)音声処理、(3)図形・画像処理の3つに大別される。そして、自然言語、音声、図形・画像等の自然な入出力形式で人間との交信を可能にする知的なインタフェースを作るための基礎技術の開発が目的である。研究は前期(実験中心で中期以降の計画作成)、中期(知的インタフェース・パイロット・モデル作成)、後期(知識ベース、推論・問題解決システムを統合化したプロトタイプ作成)に分けて行われる。

知的インタフェースで得られた成果は、中期から開始する基本応用システムで利用される。自然言語処理技術、音声処理技術、図形・画像処理技術は、コンサルテーション・システムのフロントエンド・プロセッサに应用される。また自然言語処理技術は機械翻訳システムの基礎技術を提供するものである。

自然言語処理技術の前期の研究項目は以下のものである。

- (1) 基本語彙集(約5,000語)の設計
- (2) 文章データベースの作成
- (3) 柔軟なパーサの開発
- (4) 文法規則の開発(日本語、英語など数ヶ国語を対象とする)

(5) (3)と(4)を統合化する技術

(6) フォント・データベースの作成

(7) 自然言語知的インタフェースの認知科学的研究
中期は意味解析技術の開発に研究の重点をおく。意味表現形式として格文法で用いる格フレーム、モニターギョ文法で用いる論理形式等が考えられるが、それらの検討を前期に行い、その結果を用いた意味解析技術を開発する。

後期は語用論的な解析技術に研究の重点をおく。主題や焦点の抽出、省略補強、照応指示(anaphoric)関係の推定の研究を行うが、それには言語学者や心理学者との共同研究が必要だろう。

音声処理技術に関しては、この10年我が国の技術は非常に進歩した。具体的には、線形予測法の開発、ダイナミック・プログラミング手法の応用、声道形状推定技術の開発があげられる。認識技術についても、単語認識技術を基本としているが、音声認識システムがすでに商品化されている。

音声処理技術の前期の研究項目は以下のものである。

- (1) 音波波分析技術の開発
- (2) 特徴抽出技術の開発
- (3) 音素分類技術の開発
- (4) 音声合成技術の開発
- (5) 音波波データベースの作成

中期では言語情報の利用技術に重点をおいた以下の項目を研究する。

- (6) 話者適応技術の開発
- (7) 連続音声理解パイロットシステム作成
- (8) 日本語音声合成技術の開発
- (9) 音声理解システム用ハードウェア・システム作

成

後期には知的な音声理解システムの作成を目標とする。

図形・画像処理技術に関しても、通産省大型プロジェクトの一つとしてパターン情報処理がとり上げられ、我が国の技術水準は非常に高い。特に文字認識技術は、すでに各種の実用化システムを生み出している。

知的インタフェースで取り上げる図形・画像処理の研究では、人間が、図形・画像を知識源として扱える環境を作ること、2次元の図形・画像から様々な知識を抽出する技術の開発、図形・画像データを知識ベースとして蓄積、操作、検索するシステムを作成し、CAD/CAM、人工衛星画像処理、X線画像処理に應用して、画像理解システムの研究につなげる計画である。

図形・画像処理技術の前期の研究項目は次の通りである。

- (1) 図形・画像処理用高級言語の開発
- (2) 特徴抽出装置の開発
- (3) 高精度高機能ディスプレイの開発
- (4) 画像処理用データベース・マシン・パイロットモデルの作成

中期では、推論マシンと知識ベース・マシンを利用した図形・画像処理システム・パイロットモデルを作成する。

後期には、図形・画像処理システム・プロトタイプを作成するが、具体的には、手書き図面の認識、VLSI/CAD、X線写真を用いた医療診断システム等の図形・画像理解システムのプロトタイプとする。

質疑応答

フロア外人 今回の会議において非常に感銘深くこれまでマシンについてお話を受けてきましたが、この時点でどうしても人間とマシンとの間のインタフェースを考える時に、人間というものに注目しなければならないと思うのです。つまり皆様方が漢字を認識しようとする努力をされているということは、つまり人間というものが漢字を認識しているからです。また音声というものに関しても人間が音声を認識し、あるいは画像を認識しているのでそれを扱おうとしているのです。私が言いたいのはこの10年間の研究計画の中で、研究全体の一環として人間そのものの研究をどのように取り入れようとしているのでしょうか。それをしなければ持っている解というのを無視してしまうことになるのではないかと思うからです。

田中穂 私もそのように考えます。ちょっと言いそびれましたが私の個人的なことですが、応用システムを開発するというのが私の話の中心であったかもしれないませんが、それと並んで第5世代のコンピュータのマンマシンインタフェースに関する部分に関しては人間の思考過程、人間は一体どういう過程で机を認識したり、自然言語を認識したりするのかという研究もやはり重要だろうと思います。私の個人的な希望としては、こういった第5世代の研究計画を通じて、今までコンピュータと縁のなかつた心理学の人であるとか、言語学者であるとか、哲学者でこういうふうな問題に関心のある人が、集まる機会がこれであればわが国としては非常に有益ではないか。

アメリカでは実際Cognitive Science(認知科学)というのが、知識工学の裏側にあるような感じがしており、そういった方面の研究というものも、通産省は今までそうしたことをやってこなかったわけですが、少なくともこういう知的なマンマシンを作るという立場からすると、できれば取り入れられたらいいな、というふうに感じています。この計画の中でそれがどういうふう to 実現されるかということはまだはっきりしておりません。

フロア外人 はい、部分的にはお答えがあったと思います。つまり通産省にどのように働きかけをして、マシンを作るということの中にはただマシンを作るだけでなくそれ以上のものが含まれているのだということ、どのように通産省の方々に説得し理解してもらうのかという、その辺のところをお聞きしたいのですが。

田中穂 私は通産省の人間でして、実は自然言語音声、画像とかいう分野に関する調査をずっとJIPDECでやってきたわけですが、去年までは言語学者の人にも入ってもらったり、心理学者の人たちにも入ってもらいました。私としてはインテリジェントなマンマシン・インタフェースを作るためには、そういう人たちの協力が是非とも必要であると考えているので、そういう方向に進めたいと考えています。

このプロジェクトは通産省だけでなく、実は日本の大学関係の大勢の先生方も動員されているわけで、ある意味でナショナル・プロジェクトですから、できればそういう方向でやっていくきっかけにしたいというふうに考えています。

フロア外人 画像とか図形の処理のお話の中で、非常に良い図があったと思います。その図の中でサーキット・ダイアグラムが出ていましたが、VLSIに

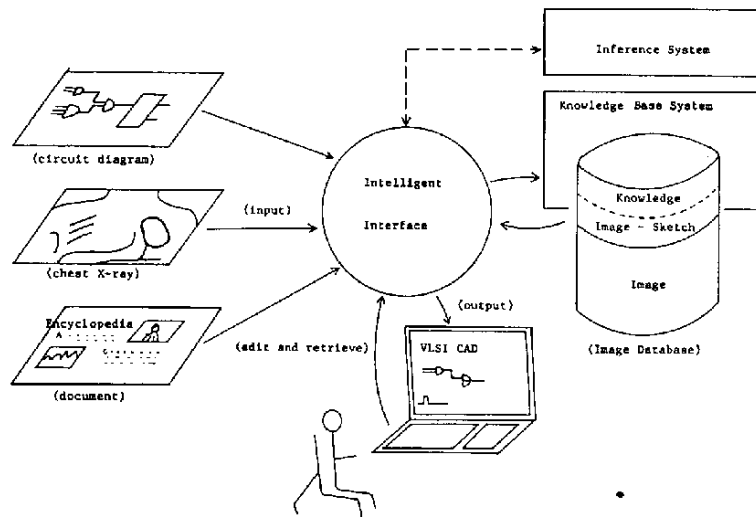


図1. Concept of pictorial functional intelligent interface

に関してのお話が含まれていました。そこでこのようなクラスの人々の場合には特別な関係を使って、作りたいたいと思っているシステムについての考えをしていくという話がありましたが、それをどういうふうにやるのか、つまりスペシャル・リレーション・シップをいろいろなロジックの中に取り入れて使っていくのかということです。

PROLOG についての話がありましたが、その中でそういった形の非手続的プログラミングをやるということですが、その中に恐らく何らかの形のスペシャル・リレーション・シップがダイアグラムの形で見つかるのではないかと思います。ただハードウェアのロジックとしてここに提出されているものがあるだけでなく、そういったものを見ていくとプログラムロジックのシステムを作っていく上で役立つのではないかと思います。

田中徳 その通りだと思います。サーキット・ダイアグラムはある種の関係を表わしているの、ロジック・プログラミングというようなものはまさにその関係を表わしているわけです。なぜロジック・プログラミングかというお話はいろいろ今までもディスカッションの中にあつたと思います。それほど毛嫌いするというよりもむしろロジック・プログラミングというのは非常にエクスプレシブ・パワーもありますが、こういったサーキット・ダイアグラムの理解であるとか、私ちょっと触れただけでしたが、自然言語の処理を行う場面などでも非常にいいものを持っているので、そういったものを取り入れてなおかつこういう実際的な問題に挑戦することによって、現在の、たとえば PROLOG というようなものをさらに拡張するという

ことも、こういった研究から出てくるものではないかと期待しています。

自然言語に関しては先ほどもちょっと触れましたが、トップダウンのパーズングの手法とロジック・プログラミングの動作がまさに一対一に対応しているわけで、非常に整合性がよろしいというふうに考えられ、われわれとしてはそういった方向を追求していきたいというふうに思っています。よろしいでしょうか。

フロア外人 幾分かお答えいただいたと思います。私が申し上げたかったのはロジック・プログラミングではなくて、他のコンベンショナルなタイプのものとか、あるいは手続的なものでもいいものがあるのではないかと、すなわちアルゴリズム的なもの、あるいはグラフィックなものも使えるのではないかと思います。

今までの問題はアルゴリズムをプログラムやコーディングの形として表わしており、アルゴリズムを構造として使っていなかったところに問題があつたのではないかと考えています。ですからアルゴリズムの構造を絵で表わすというようなことを考えるといいのではないかと思います。つまりコンピュータ・システムへの活用を考える場合に、過去はコーディングの問題としてアルゴリズムを取り過ぎていたのではないかと思います。それがまた複雑さの原因になっていたのではないかと思います。

フロア外人 私、質問があるのですがどなたにお尋ねしたらいいんでしょう。お3人の全てに伺うべきかもしれませんが。

第1世代のコンピュータで、機械言語を使っていたものから現在話し合っているものまで考えると、いろいろな研究

が行なわれてきましたが、マシンを人のオペレーションに近づけてくるといふ努力がなされてきたと思います。そのためにはいろいろなペナルティを払ってきたと思います。ペナルティとはコンピュータタイムあるいは精度が欠けているというようなものですがこうした犠牲を払ってコンピュータを人間のオペレーションに近いものにしてきたと思います。

この問題を考えると2つの点があります。一つはメモリーの問題とコンピュータタイムの問題です。一つのものを入力する場合アセンブラからコンパイラへという動きをする場合、このような仕様を書いていく時にいろいろなオペレーションがあるわけで、そのためにはメモリーもあるし、あるいはリアルタイムあるいは音声処理というようなことが行われるようになってきていますが、現在の事実から向こう10年間を展望してみると、一体どのようなことが可能だと一言で言えるのでしょうか。また第5世代コンピュータでは人間が話す自然言語が処理できるようになるのでしょうか。現在の機械の速度を遅くしないで自然言語が処理できるようになるのでしょうか。そういうオペレーションのマシンのスピードまで計算して研究の計画は立てられているのでしょうか。

田中 穂 大変むずかしい問題があると思います。トレードオフがあったのではないかとということをおっしゃったのですが、人間にコンピュータを近づけるためには犠牲があったのではないかとこのように考えておられますが、それはそれほどの犠牲ではなかったのではないかと私は思っています。スピードにしてもコンピュータはどんどん速くなっていきますし、メモリーにしても一昔前とは比べものにならないくらい安くて大量のメモリーも使えるようになってきているわけで、コンピュータというのが昔のまま進歩していないのであれば、それは確かに犠牲として処理スピードが遅くなるとかいろいろあったかと思いますが、そういうことはあまりなかったのではないのでしょうか。

たとえばインテリジェントなマシンインタフェースを作るというような場合だと、やはり一番の問題はセマンティックスをどういうふうに解釈するか、知識をどういうふうにオーガナイズするかと、そういった問題にむしろ非常に未解決の問題が多いわけで、第5世代では特にそういうことに焦点を当てて研究開発したい。

自然言語に関していうと、私、自然言語のことを少しやったので、なかなかむずかしいということは承知していますが、たとえば機械翻訳にしても完全にオート

マティックに全て話がうまくいくような機械翻訳システムというのは、10年ではなかなかむずかしいのではないかと。従ってわれわれは人間の介入ということも当然考えなくてはいけない。人間が適当な時点で介入して、あいまいであったらそれを解消してやるとか、そういうための技術というものも今後考えています。しかし願わくば多くの部分がコンピュータ化されるということを第5世代では目ざしたいわけです。

淵 いろいろ御意見をいただきありがとうございます。時間が少なくなっていますので午前中のセッションはこれで閉めさせていただきますと思います。



(4) ロジック・プログラミングと専用高性能パーソナル・コンピュータ

- 電子技術総合研究所 横井俊夫
- 日本電信電話公社 後藤滋樹
- 富士通研究所 林弘
- 富士通国際研 国藤進
- 東京芝浦電気(株) 黒川利明
- 電子技術総合研究所 元吉文男
- 東京大学 中島秀之
- 電子技術総合研究所 新田克己
- 電子技術総合研究所 佐藤泰介
- シャープ(株) 白石富勝
- 松下技研(株) 上田謙一
- 日本電気(株) 梅村護

講演要旨

1 はじめに

第5世代コンピュータ・システム(FGCS)の研究開発の踏台となる5G核言語(第0版)と研究開発支援用高機能パーソナルコンピュータを、PROLOGを題材として先行研究を行うのが我々のグループの目的である。1982年3月までに一応の結論を出すべく作業を進めて居り、現在は、必要な事項を列挙し終った段階である。本稿では我々が想定する5G核言語(第0版)と、核言語向きコンピュータのイメージを紹介する。ただし、作業の途中であるため、変更や修正が施される可能性はある。

2 言語

5G核言語として、述語論理型言語(PROLOG)を選択したのは、以下の理由による。

- ① 知識情報処理システムのプログラミングに適する。
- ② プログラミングの新しい規範を与える。
- ③ 既存言語の成果を受け継ぐ。
- ④ 新しい計算機アーキテクチャに結びつく。

次に、PROLOGの改良・拡張の要点を列挙する。第0版としては、言語の高機能化より、必要な機能をすべて取りそろえるという点に重点を置く。

(1) 抽象データ型(カプセル化)

カプセル化、副作用の局所化の機能に着目した抽象データ型を導入する。しかもPROLOGの機能を自然に延長して、この導入を行なう。これは、内部データベースという唯一の抽象データ型が存在すると見る

こともできる。したがって、多数の抽象データ型を定義できるようにするためには、多数のマイクロなデータベースを定義できるようにすればよい。マイクロデータベースの導入によって色々な利点が得られる。例えば、

- ① 副作用が局所化される。
- ② プログラムに構造が導入される。
- ③ 部分コンパイルが可能になる。

この方式の具体化には、様々のインプリメント法が考えられる。データベース内のホーン節の数、変更の仕方と頻度等によって、内部の仕組は、多種類のものを用意しなければならないであろう。

(2) 高階機能、外論理機能の整理と整備

最も基本となる機能を導入する。プログラムもデータも対(タプル)で表現するとし、分解・合成を行なえるようユニフィケーションの拡張とeval述語、あるいは述語変数の導入を行なう。さらに、カット・オペレーションを排除し、制御の構造化を行なう。

(3) プログラミング支援機能の大幅拡充

後戻り制御による実行は、デバッグを(非常に)難しくする。したがって一段と強力なプログラム作成支援ツールを用意しなければならない。

(4) データ構造の拡充

ある条件を満たす要素をまとめた集合やバグにつき、演算のための述語やユニフィケーションの拡張を行なう。

(5) 日本語処理向き機能

基本文字セットに、カナ文字、当用漢字を含ませ、述語や変数に日本語を使えるようにする。

(6) システム記述・作成機能の拡充

システム・プログラムを効率良く記述するための機能を、十分にPROLOGに取り入れねばならない。例えば、以下のような機能が考えられる。

- 効率の良い抽象データ型
- システム管理データの整理
- 並列処理記述

PROLOGは、通常のシステム記述言語に比べ、はるかに高レベルの機能を持っているため、効率を重視するシステム・プログラムの記述には適さないと思われがちである。しかし、自然な制約条件を課し、機能を縮退させれば、一般のシステム記述言語と同様の効率を保証することができる。

3 マシン

LISPに対し、単一ユーザ用の高機能のLISPマシンが開発され、商品化されている。同じように、改良・拡張したPROLOGに対しても、単一ユーザ用の高機能のPROLOGマシンを開発する。

- マイクロプログラム化されたCPU(書き込み可能なコントロール・メモリと数Mega PROLOG 命令/secの処理装置)
- 1M語(36ビット/語)のメイン・メモリ
- 200~400メガ・バイトの高速ウインチェスタ・ディスク
- 2,000×2,000BW(RGB)のビット・マップTV・ディスプレイ, etc.

全体として、できるだけコンパクトに、ビューティフルな外見となるよう設計する。アーキテクチャの基本は、高レベルなマシン語、タグ・ディスクリプタ方式等の高級言語マシン・アーキテクチャである。ただし、将来の言語仕様の変更に対応できるよう柔軟なアーキテクチャとする。

4 まとめ

言語仕様、マシン仕様の決定にあたっては、多くの例題プログラムを作成し、改良するという経験の蓄積が必要である。PROLOG(Edinburgh)は、本格的なシステムで高速処理が可能である。かなりの規模のプログラムが作成され、実験されている。PROLOG(Marseille)は、最初に開発されたシステムである。FORTRANで書かれているため、インタプリタの内部構造の学習用に用いられている。PROLOG(IBM)は、コンパクトな標準的システムである。PROLOG/KRは、PROLOGとLISPの統合を通じ新しい言語の姿を探るためのパイロット・モデルであ

る。データ構造をリストに統一、豊富な2階述語や制御構造、各種のデバッグ支援ツール等の新しい試みが盛り込まれている。DURALは、小規模なシステムながら、相対ホーン節、様相論理オペレータなどの新機構が導入されている。

質疑応答

フロア外人 プライオリティについて文句を言うつもりではありませんが。フランスの、特にPROLOGのプライオリティを言っているわけですが、つまりマルセユのコレメラウア先生が基礎的な研究をなさったために、このことを特に注目しているわけですが、ここではプライオリティのことについて私は別に反対意見を申し上げるつもりはないのですが。

横井 われわれ日本でも最初に学んだのはフランスのコレメラウアさんからのPROLOGでして、やはりコレメラウアさんのすぐれた研究が全てのPROLOGのスタートを切ったと思います。

ドワー PROLOGのセレクションに関しての基本的な問題ですが、これはたえず問われる問題です。つまりこのプロジェクトの中で非常に独特で矛盾に富んだ選択になりますが、スライドにPROLOGの選択の4つの理由が出ていました。最初は特に知識情報処理に関係あるようです。他の3つは新しいものであったように思います。非常に新しいということではこれを選ぶ理由としての説得力はないと思います。

特にこのPROLOGで書かれた場合の第5世代コンピュータの場合ですが、まずこのようなラージスケールのハードウェアの研究開発を行う前に、そういう経験を詰めていかなければならないと思います。経験を積んでラージスケールのハードウェアの方に行くべきではないかと思います。

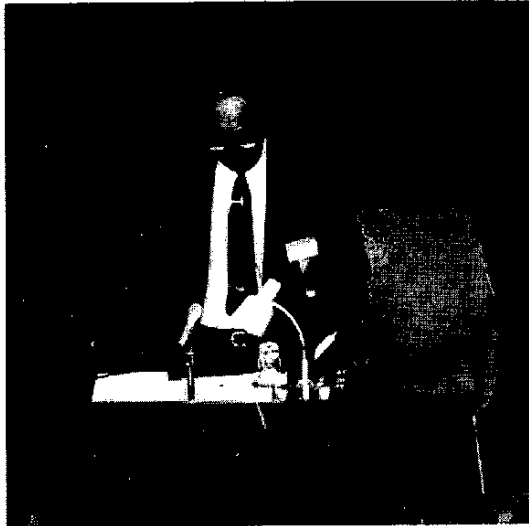
横井 PROLOGの選択の理由が適切でないという御指摘でしたが、私がもしここ2~3年以内に実用化される知識情報処理システムを作るとしたら、そのインプリメンテーション用言語にPROLOGを選択はしません。私はLISPを選択します。

ただ第5世代のプロジェクトは10年先を一つのめ途として目指しているので、明らかに10年先を考えるとLISPでは不十分です。それではLISP以外の何を選ぶか、いろいろセレクションがあります。たとえばMITのサスマン教授はLISPのいろいろな点を改良し、SCHEMEという言語を提案されて

います。それからここにいらっしゃる後藤教授も新しい機能を付け加えたLISPというものを提案しておられます。

ですから確かにわれわれにはいろいろな選択点があります。その中でなぜPROLOGを選んだかというのを、まだ十分納得いただけるだけの資料がないというのが正直なところかもしれません。ただPROLOGで行っていいんだという結論を出すためにやっているわれわれの作業、基本方針を繰返しますと、少なくともいいとわかっているシステム、LISPというシステムは十分知識情報処理システムの記述とインプリメントに使えるし、有効であることはわかっています。そこで少なくともLISPが持っているいい点はその効率までも含めて、PROLOGの中に吸収できるということをわれわれのグループは実証したい。それが実証できれば少なくともLISPがカバーしているアプリケーション・分野はPROLOGで置き換えてもよいということが言えます。

それから大規模なハードウェアを作ると言いましたが、ここで考えている高機能のパーソナルコンピュータというのは、決して大規模なコンピュータではありません。もちろんパーソナルコンピュータというには、かなり大規模な大型のコンピュータですが、決して大規模なコンピュータをPROLOG向きに作ってしまうというわけではありません。先ほどお話ししましたように当然この高機能のパーソナルコンピュータ自身も、マイクロ・プログラム方式その他で言語仕様の変更にも耐えられるようなアーキテクチャにするということは念頭において設計します。



Ⅱ.2 招待講演

(1) 第5世代コンピュータ・システムにおける記号操作の革新

スタンフォード大学(米国)

エドワード A. ファイゲンバウム

講演要旨

スタンフォード大学の計算機科学科の教授でありました、ヒューリスティックプログラミングプロジェクトの責任者としての、ファイゲンバウムは、記号処理(操作)の応用という立場に立っている。つまり、第5世代コンピュータの成否は、ソフトウェアという応用技術にあり、ハードウェアにあるのではないとしている。

記号処理の応用で使われる中心的概念が、知識であって、数学的にエレガントな論理ではない。したがって、第5世代コンピュータの課題としては、次の2つに分類できる。

第1はソフトウェアである。内容的には記号処理技術を用いたエキスパート・システム作成のための種々の道具だてを意味することである。ここで、エキスパート・システムというのは、高度の専門的知識を要する分野の、推論システムのことである。スタンフォード大学で開発した医療診断システムのMYCINなどがその代表である。こうした特定の領域のエキスパート・システムを開発するための道具だてが、ソフトウェアの基礎となる。このようなソフトウェアが、米国の研究機関で開発され、種々の応用システムが作成されていることを強調している。

第2は、知識である。エキスパート・システムの中で使われるのは、専門家の知識である。もしも、その知識が強力でないとすると、作成されたシステムの性能は、それほど高くない。ただし、ファイゲンバウムが主張する知識というのは、単なる事実ではなく、仮設、信念および経験的知識(ヒューリスティック・ルール)といったものから構成されている。こうした知

識は、長い間の経験から得られるもので、エキスパート・システムではルールとして使われる。

これらの課題は、いずれもがファイゲンバウムが提唱する知識工学のものである。この課題から考えると知識工学は人工知能の一分野であるので、全体的目標としては、人工知能マシンの実現が、第5世代コンピュータとすることができる。

ハードウェアについては、チップにどのようなソフトを入れていけばよいかというコンセプトが重要で、したがって、ソフトウェアが、第5世代コンピュータでは中心的役割を、占めている。ハードウェアの進歩からみると、いかなるマシンも作ることもできるというのがファイゲンバウムの考え方である。したがって、人的資源として考えると、必要なのは、ソフトウェア技術者ということになる。

ハードウェアへのこうしたファイゲンバウムの考え方は、1970年代の進歩、1980年代の知識処理の開発および1990年代の知識のプロセッサの応用ということが背景にある。

ソフトウェアというのは、第1の課題で述べたようにエキスパート・システムの道具だてを意味するが、これを支えているのが記号処理である。記号処理というのは、非数値の記号を扱い、数学的定式化でない分野で使われる。

そして、これらの分野は、医学、化学、生物学、経営学などがあり、物理学や工学よりも、さらに広い領域である。こうした、世の中の大部分の知識を扱うことが、また、第5世代コンピュータの対象となる。

すでに、1981年までに開発されたエキスパート・システムをファイゲンバウムはリストして種々の分野

があることを示した。例えば、医療診断のほか、故障診断、地下資源のデータ解釈、VLSI設計の支援システムなどで、いずれもが、従来の数値計算を主体とするソフトウェアでは扱いきれないものばかりと言えよう。

第1、第2の課題に対して、もっとも進歩の必要のある革新的側面は何かというと、知識獲得とソフトウェア技術者の教育の問題であるとファイゲンバウムは主張している。

知識獲得の問題は、現状で、ひとつのルール化に1時間を要することを考えると、さらに、自動化し、短縮していかなければならない。おそらく、この知識獲得の問題が、1980年代の最大の課題であるとファイゲンバウムは考えている。ひとつの手がかりとしては、人間の学習におけるアナロジーによる推論の方式を自動化していくことだと考察している。

もうひとつの教育の問題は、ソフトウェア技術者の管理者レベルにおける意識の問題である。新しいソフトウェアのコンセプトを管理者レベルが吸収し、かつ、知識工学を実践するためには、産学協同的体制と、相互のインパクトが必要であると指摘している。

その他、方法論における革新としては、①最高のパーソナルコンピュータ（例えばLISPマシン）を利用する、②プログラム環境の充実を図る、③漸近的にシステム改良をする、④プログラムコードを切り捨ててコンセプトを大切にすることなどを述べている。

第5世代コンピュータのプロジェクトに対するファイゲンバウムの個人的見解としては、次の通りである。

すなわち、コンセプトとしても正しい方向にあり、また、提案のタイミングとしても、非常に良く、スケジューリング的にもそれほど、無理がないと、そしてマーケット的にも、1990年代の個人生活を支援するエキスパート・システム（例えば、個人の貯蓄の管理など）の潜在的理由もあると述べている。もちろん、先に述べた応用領域も、第5世代コンピュータの対象となっている。

ただし、こうしたプロジェクトの課題が、非常に、むづかしいものを扱っているのも、ある種のリスクを伴っているが、分散している研究者を集め、長期間の協同研究として進めていくことが重要である。

その意味で、国際的協力により、経済的、社会的利益を得る方向でプロジェクトを推進していく必要があると結んでいる。

質疑応答

トレリーバン 今先生のおっしゃったことはほとんどその通りだと思いますが、ただAIについて少し違った考え方もあるのではないのでしょうか。第5世代コンピュータで考えているAIというものはこの会議の出席者に伺ってみると、アメリカのAIの考え方に沿ったようなものが出てきているのではないかと思います。それについてちょっと申し上げたいのですが、つまりハードウェアでなくソフトウェアが将来にとって重要なものであるという話が出ていたのですが、これについて一言申し上げたいと思います。

うまくハードウェアとソフトウェアが合成されていくということは、やはり重要であろうと思います。富士通、インテルあるいはこのようなハードウェア会社というものがVLSIの問題への解決を作ってくるといようなことは十分考えられると思います。メモリーのようなマイクロコンピュータが5年ぐらいの間には出てくると思います。そうするとKnowledge Based Systemをそういうハードウェアメーカーが作ってくるものにのせて開発していかなければならないようになってくるのではないかと思います。そうすると今開発しているようなもののインパクトが小さくなってしまふのではないかと思います。

もう一つ、先ほどおっしゃった考え方ですが、LISPとPROLOGについてももう一回取り上げて申し上げたいのです。プログラムはLISPでインプリメントされているというお話でしたが、AIコミュニティはほとんどLISP文化の集まりであるからそうなっているのではないのでしょうか。LISPのプログラムを作っているイギリスの人たちは、長期的にはプログラミングはPROLOGの方に動いていくと言っています。ですからアメリカの意見とイギリスの意見は違っていると思います。

最後にもう一点、知識工学という言葉ですが、いろいろ例を挙げられた中でエキスパート・システムについての例で、ダイクストラ先生であればクレバーなプログラムであると言われるのではないのでしょうか。エキスパート・システムとしての価値のあるようなものではないというような意見を述べられるのではないのでしょうか。

もう一つ最後にすみません。この会議で話し合われているテーマは、ほとんどがAIの人たちを対象にしているのではないかと思います。VLSIのコミュニティの人たちの話であるとすれば、第5世代コンピュー

タを考える場合にかなり違ったものが出てくるのではないかというふうに思うのですがいかがでしょうか。

ファイゲンバウム 大変長いお話だったのでお答えするのがむずかしいのですが、どの点を取り上げて私が一番異議をとないかという選択がむずかしいのです。第5世代コンピュータ会議はいろいろ考えられると思いますが、これは日本の第5世代コンピュータプロジェクトのブリリナリーな段階におけるものであって、これは1990年代のAIのマシンをねらったものですから、5.1というような第5世代コンピュータのさらに派生物を考えているわけではないので、一般的な形でいうことはむずかしいと思います。

ダイクストラやフォアがこの作業をどのように考えるか。これはおもしろいプログラミングではあるけれども、エキスパートシステムではないと言うだろうという御意見でしたが、これはかなりの専門知識を必要とするものです。シンボリックな知識、あるいは推論というものを必要とするものです。もしそれがいいプログラミングとおっしゃるならば、すばらしいプログラマであろうと思いますし、プログラマへの福音ではないかと思います。

それからLISPの話ですが、アメリカの人たちがPROLOGについて知識を持っていないというわけはありません。PROLOGを尊敬しています。イギリスではPROLOG というものがアメリカのAIワークをしのぐものであるという確信を持っており、PROLOGはスマートであると考えておられますが、私たちはそうは思わなかったわけでして、そういうふうな方法をとらなかつたというだけのことです。

トレリーバン PROLOGについての御意見ありがとうございました。私ボブ・コワルスキーに今の御意見を伝えるようにしたいと思います。

私の意見をちょっとまとめて申し上げたいのですが、私ども次世代のコンピュータというものに興味を持っていますが、このコンピュータシステムを考える場合には、今はAIマシンとして考えるのがよいと思います。ただし参加者の中にはAIを中心にやってない方もいるわけですから、第5世代コンピュータというものをもっと別の形で考えているような人もいないかということをお知らせしたいです。そういう意味で先ほど申し上げたわけで、そういうことがポイントです。この辺でやめさせていただきます。

フロア外人 ソフト対ハードの意見の違いというのは、リンクがあればいいわけで、今ここで欠けているリンクというのはコンピュータ・アーキテクトであ

ります。リクワイアメント・エンジニアリングというのがそもそもアーキテクチャのデザインの始まりであって、それはソフトウェア・エンジニアリングの知識に基づいて発生するものです。

昨日相磯先生が示された絵の中に様々な仮想化のレベルが示されていましたが、こういうものについてアーキテクトは考えていかなければならないと思います。アーキテクトがまさにこれらを設計していかなければならないわけです。ですからチップだけを設計するわけではありません。また設計を考えていく場合様々な仮想化のレベルがあると思います。まずリクワイアメント・エンジニアリングがあって、その次の段階があるわけです。ですからコンピュータ・アーキテクトはソフトウェア・エンジニアリング以外のことも知っていなければならないわけです。つまりハードウェアのパフォーマンスあるいはフィジビリティといったようなことを知っていなければならないと思うので、先ほどの先生の御発言を少し修正させていただきたいと思います。

ソフトウェアとコンピュータ・アーキテクトが協力して初めて目標を達成することができるということをお知らせしたいと思います。

ファイゲンバウム まさにその通りであろうと思います。私ちょっと強調したいのですが、私が申し上げたことはソフトウェアが問題であってハードウェアが問題でないとした意味は、急務を要するのはソフトウェアの研究であると、私はあれはハードウェアの人たちに対する尊敬の念をもって申し上げたわけです。つまりハードウェアは非常にすばらしい、ハードウェアはボトルネックではないということをお知らせしたわけです。ハードウェアの人たちは欲しいと思うものは何でも提供してくれたし、これからも提供してくれるであろうということをお知らせしたかったわけです。

15年前ならばハードウェアがボトルネックだと申し上げたでしょう。しかし今やハードウェアはボトルネックではないということをお知らせしたかったわけでハードウェアの人たちに対して多大な尊敬の念をもって申し上げた意見ですので強調しておきたいと思えます。

フロア外人 先生はかなりの国際協力がこのプロジェクトの研究開発に必要であるということをお知らせされました。私JIPDECから参加しておられる方と少し話をすると、JIPDECの方から研究開発の段階においては、自由な国際的な協力あるいは交流が必要であるというふうにお考えいただけるようです。しかしなが

らJIPDEC では研究というのは簡単な段階であるけれども、プロジェクトが開発の段階に至った時に、日本側としてどのくらい具体的な開発の情報が分かちあえるかということについては、まだはっきりしていないという御意見があったようですが、あらかじめ開発の段階においても、国際協力がオープンな形で行われるべきであるということを決めておく必要があるとお思いでしょうか。

ファイゲンバウム はい、私もそうです。一番考えられる簡単な返事はイエスです。

田中穂 これでこのセッションは終了します。



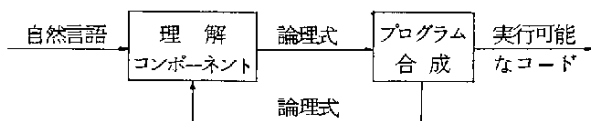
講演要旨

この会議は第5世代コンピュータを主題にしているが、私のテーマと密接に関連していると考えている。技術的な内容に入る前に第5世代コンピュータのような知的なシステムの必要性を考察してみよう。

今日、現実の世界において解かなければならない問題 (problem) の規模は膨大であり、また質的にも複雑度を増している。もちろん人間は強力な機械 (コンピュータ等) を手にしているのだが、それを十分に使いこなしているとは言えない。もう少し機械の側が知的になる必要がある。

以上の議論はコンピュータに限らずに言えることであるが、特にコンピュータについては、次のような知的な機能が期待される。すなわち、プログラミングそのものを強力にバックアップしてくれること、自然言語によるマン=マシン・コミュニケーションが実現されること、解くべき問題に対する宣言的 (descriptive) ④¹、記述的) な記述が許されること、プログラムの評価や修正の機能がシステム自身に内蔵されていること。

私の研究は、まさにこのような考え方に立脚して約7年前にスタートした。ただし、プロジェクトと言えるような体裁になったのは2年ほど前からである。私の考えているプログラムの合成法 (synthesis) は次のように図示できる。



④¹ ここでは descriptive ↔ algorithmic という対で考えている。

(2) 論理的プログラム合成

ミュンヘン工科大学 (西独)

ウルフガング・ビーベル

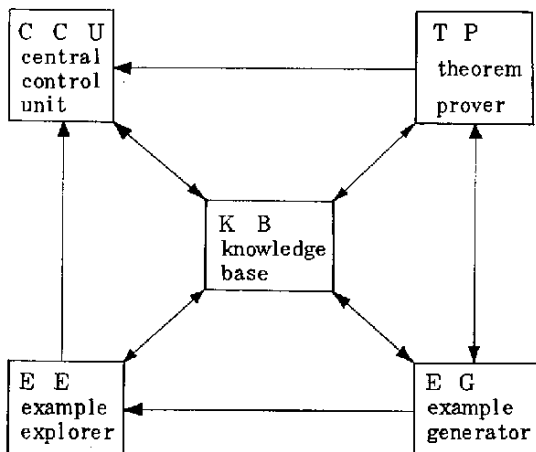
ここで幾つかの注意を述べておきたい。まず、論理式は中間言語として使われており、ユーザに直接に見えるわけではない。ちょうど論理式をアセンブラ言語のように考えてもらうと良い。ここで論理式を中間言語として選んだ理由は、自然言語に近いこと、正確で冗長性が無いことが好ましいと思ったからである。この分野ではLISPが良く使われる傾向にあるが、LISPは関係 (relation) を扱うことが出来ないので論理式の方が優れている。論理式では関数型 (functional) の記述が出来ないと誤解している人がいるが、これは正しくない。すなわち、論理式の sublanguage として term language があり、term (項) の部分は紛れもなく関数型になっている。

私がここで論理式と言う場合には full first-order logic を意味する。full というところで、私の考えでは quantifier というのは重要な働きをするので、(quantifier を取除いた) PROLOG では充分でない。また first-order にしているのは、とりあえずのスタート点として選んでいるので、将来的にはより高階の論理が必要になると考えている。

私のシステムは LOPS と名付けられているが、これは Logical Program Synthesis system の意味である。LOPS の概念図を下に示す。④²

EG、EE の部分は人間の思考法を真似したもので、簡単な例 (example) を作ってその上で考えようと言うわけである。CCU は全体の制御を司るものであり、KB はいわゆる知識ベースである。プログラムの合成にお

④² この図は Proceedings に印刷されているものと、当日 OHP で示されたものと細部 (矢印の向き等) が異なる。ここでは OHP に準拠した。



ける知識ベースの重要性は上図からも判るが、合成においては知識ベースに劣らずT Pの働きが本質的に重要である。結局、私の研究はT Pの部分を中心に扱ったことになる。定理の証明法は普通の resolution method (導出法)ではなく、connection method (連結法)と呼ばれるものを新しく考案した。

質疑応答

マコーミック 先生のシステムではグッド・ヒューリスティックというかグッド・ゲスというか、それはどのようにしてつかまえるのですか。これを先にトライした方がいいとか、いうことはどのようにしてわかるようになっているのですか。何かパターンのようになっているのですか。教育 (discipline) の中の Applied Knowledge みたいなものをどのようにしてつかまえるのでしょうか。ヒューリスティックなどというのはロジックとしてはつかまえにくいと思うのですが。

ビーベル 2つの側面があると思います。1つはシンタクティカルなものであります。たとえばこのSというエレメントを考えてみましょう。アウトプット・コンディションがあるわけで、その中には幾つかの項目が入ります。シンプルな項目を先に選ぶというのが戦略としてヒューリスティックなやり方であろうと思います。われわれの経験から考えてそういうふうな形になるわけです。この簡単なものから選ぶというのが強力な戦略です。

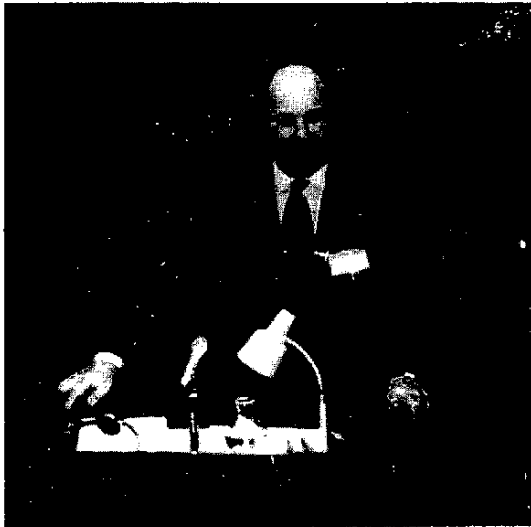
ロジカル・フォーミュラの構造から考えて、それを調べるとその後どういふふうに進んだらいいかということがかなりわかります。問題が複雑になると充分にこなせないで、それだけ Knowledge Heuristics とも呼べるものが必要になり、それを知識ベースに入れて

いかなければなりません。集合あるいは順序性のある集合のようなもの、あるいはグラフ的な構造などを考えた場合に、現在の知識ベースというのはほとんど情報が入っていないが、各サブエリアについてはまず何を最初にやったらいいとか、どういふふうにしていったらいいかということが入っているので、やはり簡単なものからやって経験を積んでいくというようなタイプの構造がいいというふう考えたわけです。

ですから意味的にも文法的にもそういうふうにしていった方が、評価をする機能を埋め込んでおいてそれを活用できるのではないかと思います。その方がオルターナティブよりも良いのではないかと思います。

シンタックスを活用すべきであろうと思います。特にここで伝えたいメッセージはシンタックスは大変多くのものを語るということです。私、ロジックをフィールドとしては大いにプロモートしているわけです。Representation Language としてはロジックを大いに押しているわけですが、それはやはりシンタックスからかなり理解できるということと、それからきょうお集まりの方はA I関係者が多いということから、私はロジック・フィールドというのを大いにプロモートしているわけです。

淵 ただいまの講演は先ほどのファイゲンバウム教授の講演と相補うような側面を指摘していただきました。



講演要旨

カーン氏はフランスの情報処理研究センターである INRIA のマネージャとして、プログラミング言語 Ada の開発にたずさわるなどソフトウェア工学の権威である。

博士はソフトウェア工学の立場から第 5 世代コンピュータ・システムの応用分野である非数値計算をとりあげて意見を述べている。

内容は記号計算の現状をまず紹介し、次にその共通の方法論や問題意識を述べ、最後に今後の新しい応用分野について意見を述べるという形式である。

標題の記号計算 (Symbolic Computation) とは耳なれない言葉である。普通は非数値計算 (Non-numeric Computation) とか記号処理 (Symbolic manipulation) とか呼ぶ。ただし記号処理は主として数式処理 (Formula manipulation) の範囲に限定され、一方、非数値演算は画像やデータベースまで含む広い概念になることが多い。

カーン氏が記号計算の例として取り上げ、紹介するのは次のようなものである。

- ① 数式処理システム — 数値計算の代わりに数式そのものを扱う。たとえば形式微分、形式積分、式の展開や因数分解を行なう。FORMAC, REDUCE などもあるが、特にシステムとしての完成度の高い MACSYMA を例として取り上げる。MACSYMA は LISP で書かれ、物理以外の多くの分野で利用されている。
- ② 構文処理システム — コンパイラ・コンパイラなど、パーサ処理システムのこと。特に構文規則その

(3) 記号演算の展望

国立情報処理オートメーション工学研究所
(INRIA:仏)

ギルス・カーン

ものを扱うメタ・パーサに注目する。

- ③ プログラム処理システム — データとして「プログラムそのもの」をとり、プログラムの変換、プログラム構造の抽出、プログラムの生成などを行なう。例として INRIA で開発された MENTOR をとりあげる。MENTOR には MENTOL という記述言語と METAL という統一化設計手法があり、言語 Ada の仕様記述が行なわれた。
- ④ 意味処理システム — 形式的な字句上のプログラム処理だけでなく、プログラム実行段階の処理を記述する意味処理/記述システムがある。例として SIS と DSL を挙げた。問題点は効率の悪いこと。
- ⑤ プログラムの証明の検討システム — データとしてプログラムの証明となっている公式を入力し、この証明が正しいかどうかを検討するシステム。例として Stanford/Edinburgh の LCF システムをとりあげる。
- ⑥ 定理コンパイラ — いわゆる定理証明システムというよりは、各種の定理をまとめてコンパイルするものである。例として FORMEL に触れている。ここに挙げられた記号計算システムは人工知能的な知識工学システムというよりは、ソフトウェア・ツールに近い実際的なものとなっている。これらの記号計算システムの特徴的方法論としてカーン氏は次のような項目を挙げている。
- ① 小グループの研究者により多年にわたって開発されている。
- ② 最初はライブラリ形式でユーザのノウハウを吸収し、要求範囲が明確化した時点で機能拡張を行ない効率を高める。

- ③ システム構成が透明でユーザが簡単に拡張を施すことができる。
- ④ 莫大な記憶域を必要とするのでガーベジ・コレクションなど記憶管理技法が必要となる。
- ⑤ 計算時間がかかる。効率的なインタプリタが不可欠。

第5世代コンピュータとの関連では新しいハードウェアにより上記④、⑤の記憶容量と計算時間の問題が解決され、効率的なシステムとなることが期待される。②、③の拡張可能性は第5世代コンピュータで考慮しなければならない。

これからの記号計算の応用分野としてカーン氏が挙げたのは、

- ① 超LSIの設計
- ② 文書作成
- ③ CAI (コンピュータによる教育)

の3分野であった。

質疑応答

淵 シンボリックコンピュテーションという考え方は非常に興味深いし、非常に重要だと私も思います。この点については全く異論がないのですが、非常に本質的でないところで一つお伺いしたいのです。

先ほどのお話の中で推論というのとコンピュテーションというの、一つはこちらで一つは別のこちらという、全く別のもののように表現されましたが本当にそうなのでしょうか。私の感じではそれは非常に密接に関係していて、コンピュテーションというのは非常に特殊な形態の推論だと思うのですが、全く別のものなのでしょうか。

カーン 絶対的に別個なものであるということではありません。つまり唯一の解決はロジカルリダクションすることだけではないということです。シンボリック・プロセッシングを行う方法は他にもあります。ですから今日リダクションによってアプローチすることは夢に見ることもできないと思います。つまりインプット・プログラムをし、そして副作用をプロシージャコールによって起こした場合、この方法はストアによるプログラミングの記述があるわけで、プログラム言語に関する全ての知識がここでは問題になるわけです。

私の考えでは今日ではこれらの知識をデータとしてオーガナイズすることはまだできないと思います。つまりプロシージャという意味ではある程度オーガナイズすることはできるが、データという意味では、私の

考えでは、どういうふうにオーガナイズしていいかということはまだわからない段階にあると思います。

ジョンパニングのシーセスで行ったチェックのタイプは、コンパイル時に行うわけで、これは非常にエフィシエントです。しかしながらどんなロジックも向こう5年間の間にこれを達成することはできないというふうに考えています。これが私の見解です。

淵 結構ですが、ロジックのコンセプトは非常にむずかしいものですから、後でまたこのことについて議論すべきではないかと思います。何かコメントありますか。

フロア外人 先ほどのアプリケーションのスライドで、特に新しいアプリケーションのところ（下図）全てのドキュメンテーションはシンボリック・フォームで書くことができるとおっしゃいますか。先生は特にシンボリックプロセスをドキュメントパーソナルな形で使うということをおっしゃっているのですか。4つのドキュメントのプリバレーションのこと言われましたが、これをシンボリックで行なった場合どのようなシンボリックランゲージをお考えになっているわけですか。このことをもう少しお話し願えませんか。

1. VLSI Design
 - Structuring designs
 - Integrating all aspects of design
 - Organizing libraries
2. Preparation of Technical Documents
 - Giving a logical structure
 - Integrating graphs and graphics
 - Heterogeneous Documents
3. Computer Aided Instruction
 - Paradigms from Program Preparation
 - Techniques from Compiler Technology
 - Techniques from Proof Checking

図1 NEW APPLICATIONS

カーン これは非常に簡単なコンセプトであって、たとえば現在のテキストのフォーマットを見た場合に、これは基本的にはつまりドキュメントの物理的なアビレンスということが問題だと思いますが、ドキュメントをたとえば処理する場合に、あるプログラムのドキュメンテーションという例をとった場合に、このドキュメンテーションをプログラムにリンクしたいとい

うふうに考えることが常です。

2番目にはその構造が非常に重要なわけでプログラムのある部分をシンボリックな形で参照するかもしれませんが。たとえばプロセスがこういう理由でやれたということをとらえるために、このインフォメーションが結局はプロセスの名前のフィジカルペアレックスからとらえなければいけないわけです。たとえばチャプターとかシンボリック・リファレンスとか、つまりコンシステンシー・チェックングをドキュメンテーションで行いたいとすれば、もっと自然現象よりも形式化しなければいけないというふうに思うわけです。これは非常に価値のある目標だと思いますが、まだ時間もかかると思います。1990年までにはテクニカルドキュメンテーションをこのような形で処理することができると思います。これが一つのアプローチであります。

黒川 フェーゲンバウム先生はいわゆるAIの分野から第5世代をとらえ、ビーベルさんはロジックの分野で第5世代をとらえ、そして今カーンさんはシンボリック・コンピューテーションの立場からとらえられたのですが、その中で今まで出てきた知識とロジックとそしてカーンさんのお話はぼくにはエフィシエンシーのお話聞こえたのですが、第5世代の中でエフィシエンシーの話は今までどちらかというところ出なかった観点だと思います。ただし私がちょっと疑問に思うのはエフィシエンシーというのは、先ほどちょっとカーンさんがもらされたような、たとえばメモリーが大きくなるとかそういう形のハードウェアの観点から解決できるものなのか、それとももっと大きな問題、たとえば私がちょっと疑問に思っているのは、エフィシエンシーというのは実はわれわれが問題をどうとらえどう解決するかという問題解決のことがら全体にかかわってくるのではないかと、そう考えたのですがそのあたりに関してまずカーンさんがエフィシエンシーの問題を、この第5世代の中でどうとらえておられるかをお伺いしたいと思います。

カーン お答えいたしますが、他の講演者からもお答えいただいた方がいいかもしれません。まずこの第5世代コンピュータを語る場合には、最初にかなりのソフトウェアを作らなければいけないわけです。そのためには、ツールが必要でして、ツールがなければどうにもならないわけです。さてエフィシエンシーの問題ですがリアルタイムでやるか、あるいは一括でやるかということを考えてみた場合、サーキットというのは対話的に使われるものであって、やはり安全な形で

やらなければいけないわけです。毎晩ルールチェックのところへ送ってトランジスタが全部うまく動いているかどうかを調べてもらうというような場合では、かなり操作形態というものが違うわけですから、それをまず理解する必要があると思います。

またニューメリカル・アプリケーションの場合には、そのコンピュータの処理量というものもずいぶん変わってくると思います。記号処理と数値計算とではコンピュータの利用度というものは変わってくると思います。ですから今われわれが使っているコンピュータよりも10倍以上の能力の高いものが必要になってくるということ、これははっきりしています。たとえば2週間ぐらいたったらだいたいこういうものをこれだけでやりたいというように要求が変わってくるだろうと思います。私の経験からして効率というのは、システムを異った形で使おうとする場合大変重要であると思います。むしろ異論のおありになる方もあろうかと思いますが、他のお説がありましたらどうぞ。

ファイゲンバウム 御質問のエフィシエンシーの問題ですが、この問題は2つの異ったレベルで言及すべきだと思います。第1のレベルというのはコンセプト的な問題解決のレベルであって、第2はメモリー、CPUスピード、コンパイラの効率といったような技術のレベルであります。第5世代コンピュータのプレリミナリープランの図を見ると、ここに高速化が必要であるというか、メモリーサイズの拡張が必要であるということが述べられています。そこは3倍、4倍の拡大ということが述べられているわけです。問題解決のタイプのプログラムを考えると、そのプロセスがもし間違っただけであれば10倍、20倍、60倍のものが必要になってくるわけです。たとえば10の125乗の数のパスのあるようなチェスゲームのようなものを考えると、そういうものは問題解決のプロシージャとしてもいいものではないわけです。ですからまず最初にわれわれが考えなければならないことは、何と言ってもコンセプトを整理することではないでしょうか。問題解決の戦略的なレベルでの整理が必要であると思います。そうしなければせっかく作ったテクノロジーも全部侵食されてしまう結果になると思います。

ビーベル 私、手短かに一言付言させていただきます。先ほどの御質問者が私の話についても言及されましたので。幾つかのボックスを示しましたが、最後のアルゴリズムのボックスを示しませんでした。このアルゴリズムのボックスというのがまさに質問者が尋ねておられる効率に関係があるものです。つま

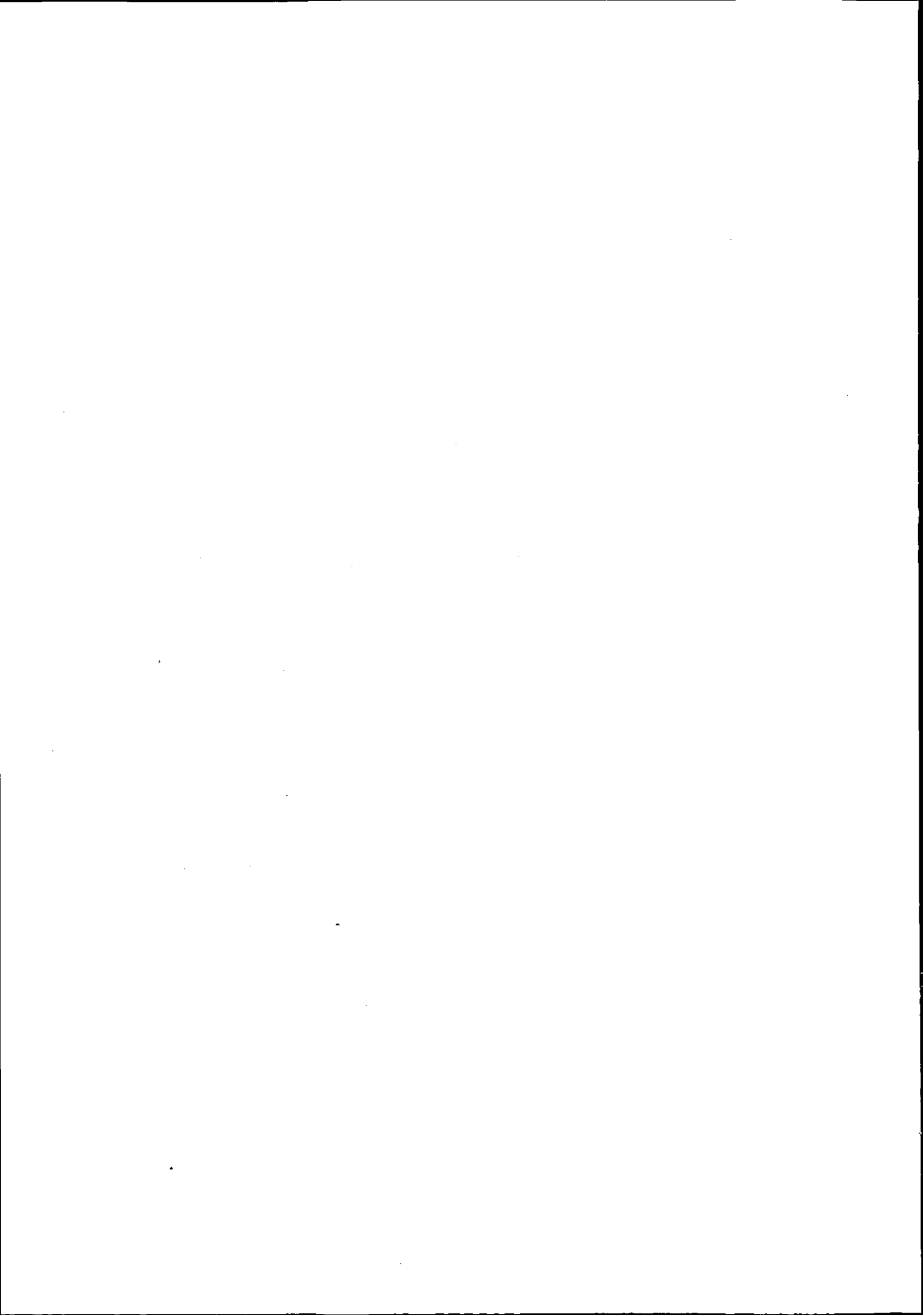
り論理的に合成されたようなアルゴリズムというのが必要なわけです。データ・ストラクチャあるいはそういうものの効率的なインプリメンテーションのために私はああいりボックスを書いたわけでした。全体のプロジェクトの組立ての中で、第5世代コンピュータを考える場合に、やはり図を見るとそういうものが入っていたと思うし、そういうものはやはり無視してはならないと思います。

プログラマはPROLOGのプログラムを合成しようじゃないかとか言っていて、それができればそれで済んだと思いがちですが、それで済んだわけではなく、さらにもう一つのレベルまで必要なわけです。ファイゲンバウム先生も言われたように効率というものは、先生が言われたように不可欠のものです。全く異った観点です。

淵 どうもありがとうございました。大分時間が長くなりましたのでこの辺で閉めたいと思いますが、私から一言。

本日3つの招待講演をしていただいたわけですが、私、それぞれに問題の非常に大事なポイントを、それぞれの観点から見てお話しいただいたと思います。私ほどのポイントも第5世代のプロジェクトにとって非常に重要であり、場合によっては考えがまだ十分ではないかもしれないが、われわれの心の中ではそれぞれの問題を考え続け、議論し続けてきたということを申し上げたいと思います。そのことについて私としては、われわれが矛盾に満ちた人種だと思っていないので、3つの話というのは、将来非常にコンシステントな意味で統合されると私は確信しています。

それではきょうは長いことありがとうございました。



Ⅲ. 10月21日(水) 9:30~17:30

1 アーキテクチャ研究計画

(1) 推論機構のための新アーキテクチャ

電子技術総合研究所 内田俊一他

(2) 知識ベース機構のための新アーキテクチャ

日本電信電話公社 雨宮真人他

(3) VLSIとシステム・アーキテクチャ

東京大学 坂村健他

(4) 第5世代コンピュータの基礎アーキテクチャとしてのデータフロー・マシン
ならびにデータベース・マシンに関する検討

東京大学 田中英彦他

2 招待講演

コンピュータ・ビジョンのための認識アーキテクチャ

イリノイ大学 ブルース H. マコーミック

3 パネル・ディスカッション I

知識情報処理

座長 電子技術総合研究所 淵一博

パネリスト スタンフォード大学 エドワード A. フェイゲンバウム

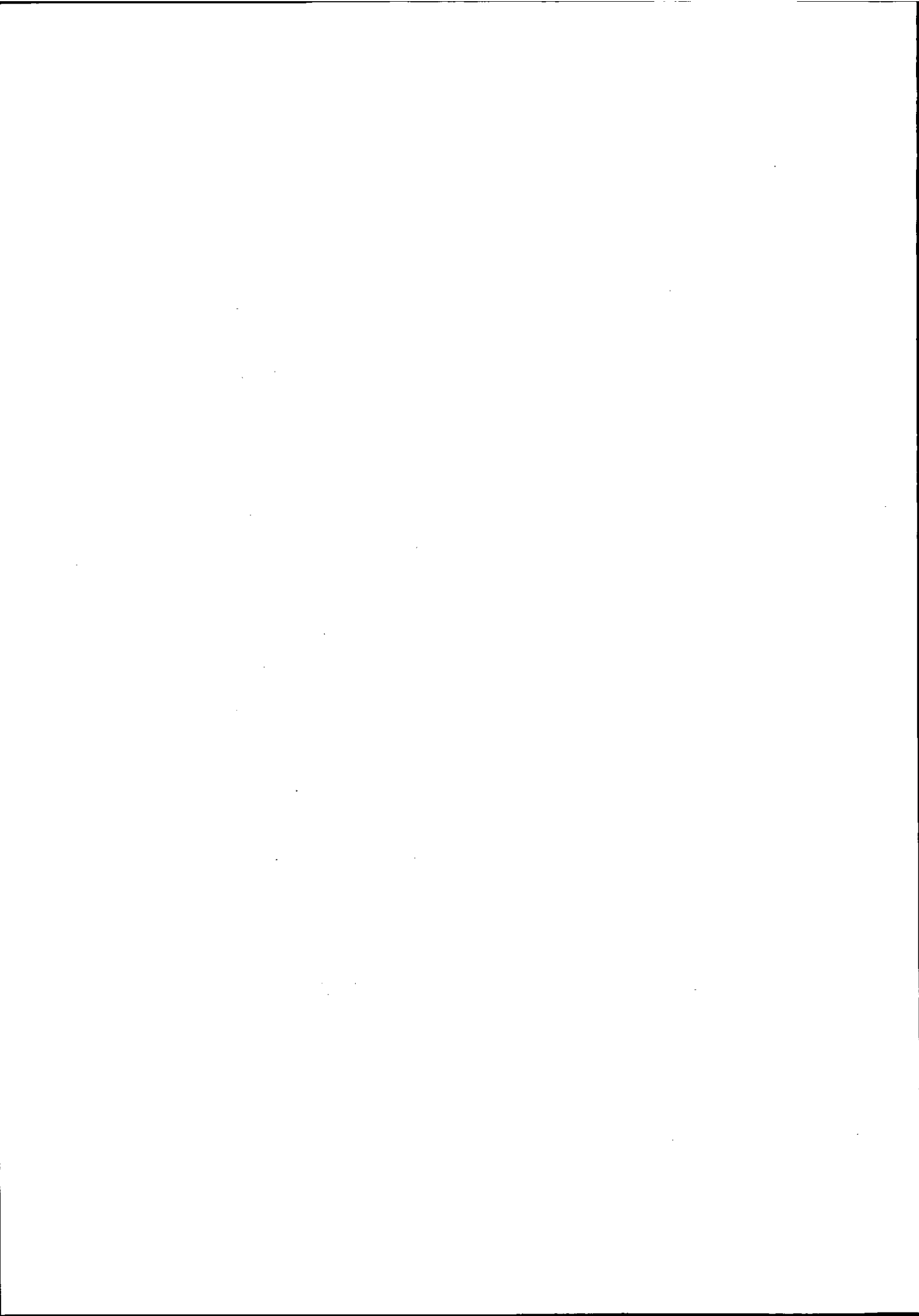
ミュンヘン工科大学 ウルフガング・ビーベル

INRIA ギルス・カーン

京都大学 長尾真

東京大学 大須賀節雄

電子技術総合研究所 古川康一





講演要旨

本論文は、第5世代コンピュータシステムの研究開発の中に含まれる一つのサブプロジェクトである推論マシンの研究開発計画について述べたものである。

全体の計画中においては、第5世代コンピュータ・システムのアーキテクチャに関連するサブプロジェクトは、この推論マシンのほかに、2つあり、それらは、知識ベースマシンおよび、知的インタフェース・マシンと呼ばれている。

本プロジェクトの最終段階では、推論マシンの研究成果は、他の2つのマシンの研究成果と融合し、第5世代コンピュータのプロトタイプを構成する。

推論マシンの研究開発の目標は、推論機構をサポートする新しいアーキテクチャを開発し、多くの知識情報処理の応用に必要とされる処理能力を備えたいくつかの実験機を作り上げることである。

実現すべき推論機構の仕様は、新しいプログラミング言語と計算モデルとして与えられる。この新言語は、第5世代コンピュータ核言語 (FGKL; Fifth Generation computer Kernel Language) と呼ばれている。

その計算モデルは、一階述語論理における機械的な定理証明の手法に基いており、最近では、PROLOGのようなプログラミング言語として与えられている。

このような言語は、知識を扱い、それを用いて種々の推論を行なうようなシステムを作成する場合の核として、優れた機能を持っている。

さらに、推論マシンに、十分な処理能力を与えるためには、VLSI技術と結合した高度の並列処理アーキテクチャが不可欠である。

Ⅲ. 1 アーキテクチャ研究計画

(1) 推論機構のための新アーキテクチャ

電子技術総合研究所 内田 俊一

東京大学 田中英彦

慶応義塾大学 所 真理雄

東京芝浦電気(株) 武井欣二

富士通(株) 杉本正勝

沖電気工業(株) 安原 宏

そして、データフロー・マシンは、このような並列処理アーキテクチャの基盤として、最も有力な候補であり、述語論理に基づく計算モデルとの整合性も優れている。

本研究開発計画は、10年間にわたるものであり、前期(3年)、中期(4年)、後期(3年)の3期に分けてある。

前期においては、データフロー・マシンの基礎研究が、まず行なわれる。これは、小規模なハードウェア・シミュレータの作成や、必要な機能モジュールの開発を含んでいる。

推論機構のハードウェア化、特に並列実行のための諸方式の検討やソフトウェア・シミュレータの試作も行なわれる。

また、ソフトウェア開発ツールとしての役割を持つ逐次型の推論マシンの開発も行なわれる。これは、中期までの間に、多数作られソフトウェアの作成の主要ツールとなる。

中期においては、前期の研究成果を融合し、並列推論マシンの実験機を開発する。これは、データフロー・アーキテクチャをベースとし、要素プロセッサ100台程度を含むものを計画している。

このほか、中期では、後期に開発するプロトタイプのために、推論マシンと知識ベース・マシン、さらには、知的インタフェース・マシンを含めた新しいアーキテクチャを構成するための基礎研究や実験を行なう。

後期では、アーキテクチャ関連の研究成果のほか、ソフトウェアの研究成果も十分取り入れ、第5世代コンピュータ・システムのプロトタイプを開発する。このプロトタイプは要素プロセッサを1000台程度含む

規模のものを計画している。

この規模のハードウェアを実現するためには、VLSI 化技術を十分に利用する必要があり、このための、VLSI-CAD システムをはじめとする支援システムも、後期の初めまでには、実用に供せられるものが開発されることを前提としている。

本計画は、多くの未知の要素を含む先端的な研究開発であることから、中期および、後期の初めに、それまでの研究開発の進行状況等を考慮した目標の精密化作業を予定しており、柔軟性を持った計画としている。

また、この計画で得られる成果は、計算機科学の進歩に大きく貢献することが期待される。

質疑応答

フロア外人 今マシンの話をされておりましたが、その時に、100～1000のプロセッシング・エレメントが入っているような、推論マシンの話をされておりましたが、どれ位のバラレリズムを、ロジック・プログラムの中に入れたらよいかとか、あるいはどのようなファクタを入れる必要があるというふうな分析は、しておられるのでしょうか。

それだけのプロセッシング・エレメントが、本当に有効に利用できるというところまで、確信した研究をしておられるのでしょうか。

ロジック・プログラミングの中に、どの位の並列性があるか、その辺は研究をしているのかどうか、従いまして、100台とか1000台というプロセッサをインプリメントした時に、どの位有効にそれが使えるか、その辺の見通しはあるだろうか、が大まかな質問です。

内田 どの位の並列性が具体的な問題についてあるか、これについては現在それらを考えるベースと言うのは、われわれは充分持ち合わせておりません。

理由はなぜかと言いますと、アルゴリズムというようなものは、やはり現在のコンピュータというものをベースに考えておりますから、それ自身いろんな意見があって、特にAIの研究される方の中には、並列性はないのだというようなことを考えている方もあります。

しかしわれわれは計算モデルの中をよく分析してみますと、やはり特にPROLOGの計算モデルというようなものを考えた時に、それ自身、並列に実行できる要素というものは、かなり多くあると考えております。

それが1000であるのか、500であるのかは、書かれる問題にデPENDするのではないかと思います。デ

ータベースのサーチのようなものというものは、本質的に並列にやっけていい部分を、多く含むわけでありませぬから、そのようなものはまず極めて多くの並列性を含むと思います。

それから知識ベースと、データベースとは違うというふうに思いますが、そのうち、今度は多くのルールを並列にチェックしていくというようなケース、そのような場合にも、多くの並列性があると思います。その時にさらに考えなければいけないのは、残念ながら多くの並列処理のうち、使える結果は、大体1つくらいだということです。

すなわちPROLOGで言いますと、ORノードというのがありますが、成功するのは多分1つ、残りのほうはどちらかと言いますと、無駄になるほうで欲しいものをさがすための手間です。われわれのように数値計算をするようなマシンを作ることに慣れた人間から言いますと、すべてのマシンの出す結果がほとんどそのまま使えることが望ましいのですが、そうではないわけですね。記号処理のようなものの並列処理では、そのような並列性というのが、極めて多くあるというふうに考えているわけですね。

ですから定量的な評価というのは、プロジェクトが始まってから、まず最初に手をつける問題というふうに考えております。

ローソン まずコメントで、あとから質問させていただきたいと思っております。

アーキテクチャを勉強している人達と言いますのは、少なくとも通訳が言ったところによると、ソフトウェア工学の要素がよくわからないところもあるということだったように思うのですが、私はそれは確かに真実があると思います。

アーキテクチャを勉強している人達は、コンピューティングの計算の部分については、よく知っているわけですね。それからアーキテクチャの会社に入る人は、少なくともコンピュータのコンストラクションのプロセスについては、特にソフトウェアのコンストラクションについては専門家でなくてもいいが、すべて知らなければなりません。

それがもし日本で、アーキテクチャを勉強している人がソフトウェア工学についてよく知らないということであれば、あまりよくないのではないかとというのが、私の1つ目の意見です。

2つ目は、これはコメントと言うより、質問ですが、私はPROLOGをあまりよく理解をし、使いこなしているわけではないのですが、PROLOGの利点は、か

なりのフレキシビリティがストラクチャー・ソリューションをつくる時にあるということではないかと思うのです。

ルールやレギュレーションに基づいてやるということだと思っておりますが、先生のご研究の話の中では、PROLOG にストラクチャリングの考え方をつけ加えていくというような側面があったと思います。これはPROLOG にフレキシビリティをつけたいと考えている人達にとっての、オルタナティブになるのでしょうか、それともストラクチャリングがついてくるということは、それとして受け入れなければならないものになるのでしょうか。

内田 最初のご質問ですが、やはりハードウェアをやっている人間が、ソフトウェアまで勉強しなければいけないというのは、今までのアーキテクチャの人間にとっては、新しい分野にアプローチしていくということですから、それ自身かなり面倒なものであると思います。

その辺は昨日、ファイゲンバウム先生がソフトウェアの専門家というのさえいないのであるということ、ヒューマンリソースが極めて限られているということと同じ意味合いのことが生じるのではないかと思います。

ですから最初の仕事と言いますのは、ソフトウェアもハードウェアもVLSIも、すべてに通じるような新しい形のアーキテクチャの専門家というのを、つくり出していくということが、1つの重要な研究の、最初の目的、ステップだというふうに考えております。

多くの研究者の方は、その必要性に充分気づいておられると思いますので、モチベーションについては、問題はないのではないかと思います。問題はそれを実際に確かめてみる研究環境というようなものを、整備することである。そちらのほうが特に日本では大事かと思っております。

例えばコンピュータ・ネットワークですとか、大体世界中の代表的な言語がすぐ使ってみれるようなコンピュータの環境というふうなものが、極めて重要ではないかと思っております。

2番目のPROLOG にストラクチャを持ち込むという話は、多分モジュール化の話だというふうに思います。現在のPROLOG の機能そのままでは、データのプログラムの構造とか、そういうのは1枚です。それ自身データベースは1つのdomainにあるわけですから、その中からルールを探そうとする時に、すべてのルールが対象になるわけです。

そういう場合のサーチスペース(Search Space)というものを、プログラムを書く人間というのはわかっているわけですから、それを色々分割して指定していきたい。それは昨日のFGKLのモジュール化の話として、横井さんの話の中に出てきたことと思います。

それは多分、私はフレキシビリティをふやす方向だと思います。但し、マシンにとっては、複雑さを増す方向ではないかと思っております。

フロア外人 パラレルリズムをVLSIを使って実現していける、すばらしい可能性があるということですが、100~1000のプロセッシングエレメントを入れていかれるということですが、ビジョン、スピーチ、タクトイル・センシング(触覚センサー)、あるいはモーダルコントロール、いわゆるロボット学とも関係のあるようなものについても、活用できるのではないかと思います。

パラレル・プロセッシングというものを、非常にむずかしいものに応用されようとしています。もっと簡単なところでも、大いに応用できるのではないのでしょうか。

内田 おっしゃる通りです。しかしながらプロジェクトの全般的な傾向というのは、極めて野心的な、より困難な問題というのを、どちらかと言いますと表面に出して、やっていきたい、そうすればそれよりもやさしい問題については、もっとむずかしい問題のアプローチの途中で、何かいいアイデアが出るのではないかと、一般的にはそのように考えているわけです。

しかしながら、まず知識情報処理というようなものが、最終的な目標として出てきた場合に、今最もむずかしいとおっしゃいましたところのシンボルマニピュレーション(記号操作)というようなものに対して、どうやって並列処理を適用するかということが、極めて基本的な問題であります。

従って、われわれはそこを避けて通るわけにはいきませんので、まずそれをいちばん、プライオリティの高いテーマとして取り上げていきたい、というふうに考えているわけでありまして。



(2) 知識ベース機構のための新アーキテクチャ

- 日本電信電話会社 雨宮 真人
- 日本電気(株) 箱崎 勝也
- 電子技術総合研究所 横井 俊夫
- 三菱電機(株) 房岡 璋
- 北海道大学 田中 譲

講演要旨

知識ベース管理を効率的にサポートするための新アーキテクチャの研究・開発に関する諸問題について論ずる。

第5世代コンピュータは知識情報処理システムとして開発される。知識情報処理システムは①問題解決・推論, ②知識ベース管理, ③知的マンマシンインタフェース, の3要素よりなり, それぞれ第5世代コンピュータ・システム開発上の重要な課題である。このうち①, ②はシステムの核部分を形成するものであり研究開発はソフトウェア, ハードウェアの両面にわたって計画される。

知識ベース・メカニズムのための新アーキテクチャの研究・開発は知識ベース管理のメカニズムを実現するのに適したハードウェア・アーキテクチャを求めるものである。知識ベース管理メカニズムのハードウェア実現をここでは知識ベース・マシンと呼ぶことにする。

知識ベース・マシンの役割と位置づけは, 問題解決・推論の機能と知識ベース管理の機能との関係において捉えられる。この関係の捉え方によって第5世代コンピュータ・システムの構造としていくつかのものが考えられるが, 現時点ではデータベース・マシン技術をベースとして開発した知識ベース・マシンと推論実行の制御を主として行なう推論マシンの2つのシステム要素を太いパイプで結合した構造が現世代コンピュータの構造とのアナロジーから考え易いものである。研究開発のステップはまずこの様なシステム・イメージを仮定して始められる。この場合最も重要なこと

は知識ベース・マシンと推論マシンとのインタフェースをどのように定めるかということである。

知識ベース・マシンの機能は問題解決・推論及び知識ベース管理についてのソフトウェア方式或はその理論モデルに従って決定されるべきものである。また推論マシンのハードウェア方式とはインタフェースの明確化という点で密接に関連しているが, ここでは知識ベース管理という面から知識ベースマシン開発上の課題を考えることにする。

知識ベースマシンに要求される基本的機能は①大量の知識を貯えるためのメモリ, ②大量知識ベースの中からの知識データの効率的な検索, ③知識データの大量知識ベースへの取り込みと融合化, の3点である。この機能を実現してゆくのに次の2つの方法が考えられる。

(1) 関係データベースマシンを基礎としてその機能を知識ベース管理向きに引き上げてゆく。

(2) 知識ベース・アクセス管理の機構を推論機構とのCo-workingという形で実現する知識ベース管理方式を確立してゆく。

このため, まず関係データベース・マシンの実用化, 及び高並列アーキテクチャ実現のための基礎研究が必要である。主な研究課題は次の様なものである。

- ① 現在或は数年後を予測しての実用的関係データベース・マシンを開発する。
- ② このマシンを知識ベース・システムの実験のための道具として用いる。
- ③ ②の研究を通して関係データベース・マシンの限界と問題点を明らかにし, これを知識ベースマシン開発の要求事項として整理する。

- ④ メモリ素子への機能要求を明確化する。
- ⑤ データフロー制御やストリーム処理の概念を取り入れ、並列型関係データベース・マシンのアーキテクチャを確立する。
- ⑥ 関係データの階層化及びそのためのメモリ階層化、構造メモリの実現法を抽象データ型サポート機構などをベースに明確化する。
- ⑦ 関係データの分散化、更新等に於て問題となる RASIS 機構などのデータベース・アクセス制御の方式を明確化する。

また、知識ベース管理方式の研究における主な研究課題には次の様なものがある。

- ① 知識データの定義とそのデータ構造の明確化、特に Extensional Data (individual facts) と Intensional Data (general facts, rules) の扱いとその管理機構を明らかにする。
- ② 知識演算の明確化、特に関係代数演算の機能拡張による知識演算の実現、パターン照合やパターン探索などを基本とする連想アクセス・メカニズム及びそのためのメモリ機構を明らかにする。
- ③ 知識データの階層化、知識の表現法やそのデータ構造、メモリ機構を明確化する際、常に知識ベースの階層構造化を前提として進める必要がある。
- ④ 並列処理、分散処理アーキテクチャの明確化、アーキテクチャの検討に当ってはデータフロー制御、ストリーム処理等の概念をベースにし、Inter-Connection Networkの実現法など実装面に於る技術も含めてアーキテクチャを明確化する。

知識ベース・マシンの研究・開発は10年計画で進められる。この10年は前期3年、中期4年、後期3年の3段階に分けられる。

前期では逐次制御に基づくプロセッサ10台程度の関係データベースマシンの開発と、関係データベース或は知識ベース管理のための並列処理機構の基礎研究を行なう。

中期では前期の基礎研究を基に並列型の知識ベースマシン実験機の開発を行なう、この知識ベースマシンはデータフロー制御やストリーム処理の概念に基づく要素モジュール100台程度からなる。この知識ベース・マシンは、同じく中期段階に別の課題で試作される推論マシン実験機とともに知識情報処理システムのソフトウェア開発のためのツールとして用いられる。中期段階の基礎研究としてはActorメカニズム、セマンティックネットワークなどの推論メカニズムと知識ベース管理メカニズムを融合させた高度分散型知識ベース

・メカニズムの明確化とそのハードウェア化の検討を行なう。

後期段階では中期段階での並列型知識ベースマシン実験機開発及び推論メカニズム・知識ベースメカニズム融合化のための基礎研究を基にし、推論マシンと知識ベースマシンとを融合化した、要素モジュール1000台程度の第5世代コンピュータのプロトタイプを開発する。

質疑応答

ドワー コメントを1つと、質問も1つあります。

このオブジェクトベースのアーキテクチャに関してですが、抽象データ型ということでしたが、INTELがこのアイデアを非常に推し進めているということを知りました。これは非常におもしろい例だと思います。

つまりハードウェアの人が、ソフトウェアの人に対して、ソフトウェアの人がユースフルだと思ふようなことをハードウェアの立場から提供するということがおもしろいと思うのですが、ハンディキャップはあると思います。

Adaのインプリメンテーションが、余りよくないマシンになってしまったということがありますから。

私の質問と言いますのは、どうしてオブジェクト指向アーキテクチャという仮説を立てて考えていらっしゃるのでしょうか。抽象データ型のためには、プログラミング言語において、なぜそれが必要だと考えていらっしゃるのでしょうか。

雨宮 多分、抽象データ型、オブジェクトというものが必要であるということが叫ばれてきたのは、いわゆるモジュラー化プログラミングからの要請だと思います。これでは、1つのデータ構造は、およびそれに対するオペレーションのarrayも含めて、1つの対象と見立てて、これをカプセルで包んでしまう、すなわち、外側からは、それに対する或る一定のアクセス基準を与えて、中については何も知らせない、それを部品として使いなさいという、いわゆるビルドアップの、ビルディング・ブロックの概念が、ソフトウェア・プログラミングにおいても、非常に重要であるということから出てきていると思います。

それをサポートするためには、一方でモジュラー化して、皮をかぶせるわけですから、当然そこには効率性というものが、逆に問題になってくるわけです。その効率性をいかにして解決してやることのできるかどうか、それは多分ハードウェア技術、今までコン

パイラとかで、ある程度オブティマイゼーション（最適化）等で、皮を一所懸命になって、薄くしていたわけですが、それも乗り越えて、もっとカプセル化とか、モジュラー化という要求が強くなってきたわけです。

そのほうがソフトウェアの生産性とか、そういうものを考えますと、非常に有用だからです。

もう1つは、当然ハードウェア技術として、そういうものをサポートしてくれるはずであるという言語屋の挑戦であるかというふうに、私は思うわけです。

それをどうやって、逆にハードウェアがサポートしてあげるかということが、非常に大きな問題です。それでINTELのマイクロプロセッサ iAPX 432 では、チップの中に、ある程度回路的なレベルまで落して、サポートしてあげようということになっているわけです。

これが逆に、ソフトウェアとハードウェアの連結が不充分ですと、言語設計者が狙っていたモジュラー化というものを、メモリー管理機構のハードウェアの束縛のために、変な形でデフォルムしてしまうという危険性があるかと思えます。その点は十分に注意して進めていかないと多分プログラミング言語を作る側にとっては、不満足な機能で、余計な機能だけが残ってしまうという結果にもなりかねないかとも思えます。

フロア外人 小さな点ですが、1つ質問があります。

雨宮先生のお話の中で、大きな問題につながるべき小さな点が、1つあったと思うのです。

知識ベースマシンを extentional（外延的）な、そして intentional（内包的）なデータを持つものとしてお話になりましたが、intentional なデータと言いますのは、もちろん種類は、推論マシンによって生成されるようなデータです。

かなり何年も前に、第0世代のコンピュータ・システムの時に、例えばラムダ・コンバージョン（lambda conversion）とかいうレベルで話していたのですが、その時に uniform なメッセージによって、extentional なものと、そして intentional なものと、両方扱うようになっておりました。

ところが第5世代のコンピュータの時には、このはっきりとした区別をして、つまり第1世代、第2世代そして第3世代において、もうこのような distinction（区別）があったのですが、今度はこういうものをなくして、結合するということは考えられないのでしょうか。

つまり一番始めのところから、結合しないで、なぜ第3番目のフェーズまで待つのでしょうか。

これは相磯先生あるいは、ほかの方に伺ってみてもいいかもしれませんが、どうして最初のフェーズのところで定義しないで、第3フェーズまで待つのでしょうか。

雨宮 基礎研究のところで、開発の最初のフェーズというのは、あしたにでも走り出さないといけないわけですから、基本検討というものを、ある意味では、得る暇がないということがあるわけです。

先ほども申し上げましたように、第1フェーズで非常に重要なのは、基礎研究であるということを言いましたが、そこでは当然、intentional データと、extentional データと結合したような、例えばセマンティック・ネットにおいても、うまく quantified variable をどういうふうに融合していくか、というような話もあるわけです。

それをハードウェア機構の中に、どう入れていくか、例えばタグ・アーキテクチャなどを持ち込んで、データを外では一見同じように見せているが、中でタグによって、variable と、atom とを区別するとか、そういうことを考え得ると思えます。

それはまだはっきりした見通しがないわけですから、当然、基礎研究として攻めていく。

この問題は最初のフェーズからの重要な問題であり、基礎研究として積極的に進めていく、というふうに我々は認識しております。

最初のフェーズの開発と言いますのは、RDBM と名づけている、それは多分商品ともなり得る、つまり実用的なマシンとしてもつながっていくものを考えています。それを中期以降の開発のための実験用のツールとして、利用するというふうに RDBM については考えております。



講演要旨

第5世代コンピュータは、知識情報処理の実現を目指す従来のコンピュータシステムとは構造理論を異にする全く新しいコンピュータシステムである。従って、その構築に当っては、部品コンポーネントから第5世代コンピュータシステムに合わせて行くという考え方が重要と思われる。部品コンポーネントには多くのものが考えられるが、現在までの研究の進み具合、並びに第5世代コンピュータシステムの実現化の目標年である1990年の技術予測に基づくと、シリコン系素子を使うのが最も適当であると考えられる。よって、シリコン系素子の部品コンポーネントであるいわゆるVLSIという百万トランジスタ/チップ以上の高集積度素子をどのように作るかが第5世代コンピュータシステムの実現を左右する鍵となる。VLSIの研究においては、VLSIの中にどのような機能を入れ、またそれをどのように作るかというVLSIアルゴリズムとVLSIを実際製作する場合必要となるデザインサポートシステム(VLSI-CAD)の研究が必要である。VLSIアルゴリズムという新しいコンピュータ設計におけるアルゴリズムの重要性をここで再確認したい。VLSIベースのコンピュータ構築においては設計における方法を変えなければならないということを主張したい。従来のマシン構築においては、設計当初よりハードウェア、ソフトウェアと分離させ研究開発が進むことが常道であったが、VLSIベースとしたコンピュータにおいては、このように分離して考えることはやめるべきであろう。VLSIの特性と、第5世代コンピュータシステムに対してのユーザの要求からアルゴリ

(3) VLSI とシステム・アーキテクチャ

- 東京大学 坂村 健
- 日本電気㈱ 関野 陽
- ㈱日立製作所 小高 俊彦
- 富士通㈱ 上原 貴夫
- 慶応義塾大学 相磯 秀夫

ズムが生まれ、その後始めてソフトウェア、ハードウェアに分けていく。即ちアーキテクチャを決定していくというような設計方法を取るべきであると主張したい。次にVLSI-CADは第5世代コンピュータシステム開発に携わる多くの研究開発者が使うということから、使い易く、標準化されたものでなければならない。そしてこの要求に答えるには、アーキテクチャの決定からVLSIマスクパターン・デザインまでが、一貫して行くことのできる階層構造記述を許す統一的な考えによって作られる設計言語を中心としたシステムの構築がまず重要である。さらに、ソフトウェア的な工夫だけでなく、高度なマン・マシンインタフェースを備えた、標準化されたターミナルを用意しなければならないであろう。ところで、一般にVLSI CADシステムが要求する計算能力は非常に大きなものであり、もちろん現在手に入る最強のコンピュータをもって十分であるとは言い難い。しかしながら、VLSI CADシステムは、10年間にわたるプロジェクトの始めから使いたいという要求が強く、よってどの時代においても最強のコンピュータシステムを使えるようなハードウェア環境が必要となるのである。当然ながら10年後には、開発される推論マシンや知識ベースマシンなどの第5世代コンピュータシステムそのものを使いたいという要求が出、その要求に答えられるようなハードウェア環境でなければならない。そこで、進化して行くハードウェア環境を吸収できるような高度な柔軟性を持ったevolutionalなシステム・アーキテクチャの研究が必要となる。またユーザの立場に立てば、このような強力なシステムを遠隔地より使いたいという要求も強い。我々はこのような考えにもとづい

て作られたネットワーク・アーキテクチャを含むシステムをシステム5Gと呼ぶ。我々は、第5世代コンピュータシステムのような従来のコンピュータとは理論を異にするような、新しいコンピュータの開発においては、その開発方法も新しいものでなければならないということに十分気がついている。そして、新しい考えに基づいて作られる開発システムの構築には非常に時間がかかるということも良く理解している。しかし、我々はすぐにも、今日からでも開発システムを使いたいのである。この要求にどう答えるのか、また恐らく新しい開発システムの要求に最も適応したコンピュータそのものが第5世代コンピュータシステムなのである。この矛盾にどう答えるのか。この一つの解が提案するシステム5Gだと我々は強く信じる。なお、本研究開発テーマは、大別すると2つに分かれ、①知的VLSI CADシステムの構築。並びにこのVLSI開発システムとソフトウェア開発支援システムを動作させる環境となる、②ソフトウェア・ハードウェア開発支援システム、システム5Gの開発の2つである。

質疑応答

フロア外人 このプロジェクトで、むこう3年間に必要なマンパワー(人力)はどの位でしょうか。何人位が必要であるかということです。

坂村 できるだけ沢山ほしいのですが、私はちょっと予算のことがわかりませんので、元岡先生にお願いいたします。

元岡 このプロジェクトに実際にかかわる人と、そのプロジェクトをサポートする人と色々いると思うのです。従ってプロジェクトで、例えば今のVLSI-CADシステムの話が出てきましたが、このプロジェクトでこのシステムを全部つくるのか、そのうちの一部をつくって、ほかの部分は、すでにあるものを使うのか、そしてすでにあるものと言っても、その時点において、いちばんいいものを使う。そういった時に、そのシステムをつくっている人達を、われわれのプロジェクトの中のメンバーと考えるのか、考えないのか、非常にむずかしい問題を含んでいます。

しかしながらここで言えますことは、このプロジェクトを成功させるか、させないかということには、いかに沢山の優秀な技術者を、このプロジェクトのもとに結集できるかということにかかっていると、考えております。

その時に日本の雇用の形態といったようなものが、

諸外国 — 特にアメリカなどとは、かなり違っているといった問題もあり、ここで私が何百人と言ってみても、余り意味のある数字にはならないと思いますので、残念ながら申し上げられません。

ただ直接かかわる人として、あるいはこのプロジェクトが経済的に直接サポートするような人としては、数百人というようなオーダーを考えております。

トレリーバン 先生は、例えばシーマス等の技術を標準化することを、お考えになっていらっしゃいますか。

坂村 そうです、すべてのものを標準化するという事はよくないと思いますが、特にある程度の標準化がないと、研究は進まないと思います。

特にVLSIの場合には、標準化は重要な項目だと、考えております。つくり方を教えるのは、やはりむずかしいですから、CADシステムにしる、教育システムにしる、つくるのにお金がかかりますから、できるだけ標準化してしまったほうが良いと考えております。

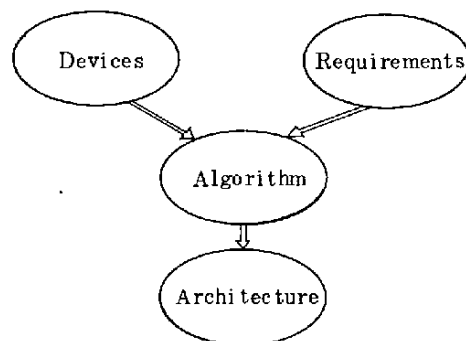
ローソン まず最初にこのすばらしいご発表に対して、心からおめでとうと申し上げたいと思います。

リアリズム、現実的な言葉が使われたのは、始めての発表だったと思います。つまり3年間で実現可能なことと言うことで、非常に興味深い点をお話しいただきたいと思います。

プレゼンテーション、非常に興味深かったと思いますが、多くが正しい方向に進んでいられると思います。

1つはシステム5Gを使うということですが、システム5Gは実際に使えるかということがあります。システム5Gを使わなくても、このプレゼンテーションの中に、非常に興味深い点があったと思います。

私は特に興味をひかれましたのは、最初のほうのご発表の中でお示しいただきました図で、デバイスからの要請(リクワイヤメント)が出ていた図がありましたが、そこからアルゴリズムが引出される、そのアルゴリズムに基づいて、アーキテクチャができていたと言われました。



これがなぜVLSIだけに使われるかということですが、つまりこのことは、このシステムの中の、どのレベルにも使うことができると思います。ソフトウェアのレベルとか、そういうところにも使われることができると思いますので、このようなコンセプトを、ソフトウェアのCADのためのバージョン(Version)それからハードウェアのCADと、こういうふうに分ける必要はないのではないかと思います。

坂村 そう思います。おほめいただき、ありがとうございました。

高橋 ネットワーク・アーキテクチャのことが、先ほど出ていましたが、グローバル・ネットワーク・アーキテクチャで既存のものを考えておられるのか、それとも例えばOSIのようなものを考えておられるのか、その辺のところをちょっと伺いたと思います。

坂村 これもネットワーク・アーキテクチャを構築するのは、やはり時間がかかりますので、現実的に対処していかなければいけないので、今お答えするのは、何ともむずかしいと思います。

しかしできれば新しいネットワーク・アーキテクチャで、知識を流通させることができるような、新しいネットワーク・アーキテクチャを考えたいと思っております。今いろいろな会社がつくっている、商業的に提供されているものは、かなり低いレベルのコンピュータ同志の通信だと思うので、高いレベルでのコンピュータ・コミュニケーションができるような新しいネットワーク・アーキテクチャ、プロトコル、そういうものを提供していかなければいけないと考えております。しかしその場合OSI, IEEEの標準化案などは十分参考にいたす所存です。

ギョア 2つ簡単な質問があります。まず第1は、100メガbit/secの光ファイバーのところ、この光通信方式は具体的に何ですか？

第2は、先生のHSLの開発において、先生はこれを国際的な標準、例えばCONLANとかそういったものがありますか。例えばシステムとか、科学者のレベルでも結構ですが、そしてまたどの程度まで、記述仕様を統一することができるかと考えてしょうか。

以上2つの質問であります。

坂村 最初の質問に対してのお答えとしては、現在研究中というところですが、物理的には光ファイバーで100メガbit/secのほうでつなげるというようなことは、明らかなわけですが、そのような素材が使えるという前提でこれから通信方式の具体案をねっていきま。残念ながら研究中なのでこれ以上お答えできま

せん。

2番目の点ですが、われわれCONLAN・プロジェクトみたいなものは、非常に興味深くみております。事実一部CONLAN・プロジェクトから影響を受けることが、非常に大きかったのです。

ただわれわれとしては、VLSIをもう少し意識したCONLAN・プロジェクトみたいなものをやりたいというようなことであるという点と、ソフトウェアの開発とつなげるというようなことをやりたいということでもあります。

ただ5G-HSLにしても、大きなシステムになると思いますので、現実的に考えて、果してどこまでできるか、もちろんソフトウェアまでやりたいのですが、やはりもう少し検討してみないとわからない事が多い。

最初に要求を整理するというところから、始めないといけない、今考えております。なお記述仕様の統一化については強く出来ると考えておまして、また逆にやらねばならない事でもあると思っております。



講演要旨

1. はじめに

第5世代コンピュータ・システムの2サブシステムである推論マシンと知識ベースシステム、それぞれの基本機構であると考えられるデータフローマシンとデータベースマシンについて、それらを実現する為に必要な検討項目を明らかにし、詳細な研究計画を立てることを目的として検討グループが今年の7月より作業を開始している。この検討グループは本年度末に報告書を出すことになっているが、ここではその中間結果について述べる。

第5世代コンピュータ・システムに於けるこれら両マシンの役割は次の通りである。推論処理は一般にかなりの計算能力を必要とするが、もともと多くの並列度を内在していると考えられ、従って優れた並列処理機構であるデータフローマシンはそれに非常によく適合すると考えられる。一方、知識ベースでは多量のデータを能率良く蓄え、高速に所要の情報を引き出す能力が必要とされる。データベースマシンはそのような処理に適したものであり、従って知識ベースシステムの中核機構となることが期待される。

2. データフローマシン

データフローマシンの検討項目は次の通りである。まず、データフロー向きの言語を開発し、オペレーティング・システムや並列処理アルゴリズムを検討すること。次にデータフローマシン内の基本実行単位であるアクティビティの発火検出やトークンのラベル付け機構、基本命令セット、入出力機構、及び全体の構造

(4) 第5世代コンピュータの基礎アーキテクチャとしてのデータフローマシンならびにデータベースマシンに関する検討

- | | |
|---------------|------|
| ・ 東京大学 | 田中英彦 |
| 日本電信電話公社 | 雨宮真人 |
| 北海道大学 | 田中譲 |
| 榊日立製作所 | 門脇吉彦 |
| 日本電気㈱ | 山本昌弘 |
| 電子技術総合研究所 | 島田俊夫 |
| 榊富士通研究所 | 相馬行雄 |
| (財)日本情報処理開発協会 | 滝沢誠 |
| 三菱電機㈱ | 竹内彰一 |
| 沖電気工業㈱ | 伊藤徳義 |
| 東京大学 | 喜連川優 |
| 東京大学 | 後藤厚弘 |

を明らかにすること。更に、VLSIを用いた実装法や高信頼化手法、デバッグ手法を開発するとともに、シミュレーション等により様々な応用に対する実効並列度を調べること等である。

研究開発のステップは次のように3期に分けて進めることを考える。まず前期は、データフローマシンの基本的な可能性を示し、基本構造を固めることを目的に進め、アクティビティ・メモリ、処理ユニット、構造メモリ、結合網等の機能モジュールの検討や、それらを組合せた実験システムを試作する。更に、ソフトウェア・シミュレーションにより様々な評価を行なう。

中期は実用化技術の開発が中心で、VLSIによる実装法、オペレーティング・システムの開発、推論マシンとの融合法について検討を行なう。後期は大規模システムの開発を目的とし、高並列実験機を試作し、第5世代コンピュータ・システムの基礎マシンとする。

前期実験システムの仕様としては、処理ユニット16台、各ユニットはマイクロプログラム制御、アクティビティ・メモリや構造メモリは16バンクで、データフロー用高級言語を開発し、従来のコンピュータをホストとして接続することが考えられる。中期では処理ユニット128台、後期では1000台程度が目標である。

3. データベースマシン

データベースマシンを開発する上で必要な検討項目は次の通りである。まず、マシンに持たせるべき機能を明確にしホスト・インタフェースを定めること、関

係演算等の並列アルゴリズムを明らかにするとともに、様々なレベルの並列性を引き出すハードウェア構造を示すこと。大容量データから必要な部分を能率良く抽出してワーク・メモリに持ってくる手法の開発。非定型データの符号化、データの一貫性維持手法、並びに分散データベース管理手法の開発等である。

研究の前期に於いては、関係データベースマシンの可能性を示しその基本アーキテクチャを固めることを目的とし、関係代数をインタフェイスとする実験システムを試作する。また、制御アルゴリズムの開発や評価をシミュレーション等により行なう。中期では、VLSIを用いて大規模システムを試作し、知識ベースシステムに仕上げる為の諸検討を行なう。後期では、高性能な知識ベース向きマシンを開発し、推論マシンとの融合を検討するとともに、分散知識ベース機能の実装研究を行なう。

前期実験システムの仕様としては、処理エレメント16台、メモリ・モジュール32台程度の関係代数マシンとし、集合演算に向けた機構を実装する。また、20GB程度の大規模メモリからのステージング機構を持つものを考えている。

4. むすび

以上、両マシンの検討項目や研究計画の案について述べた。最終報告は57年2月末迄に定める予定である。

質疑応答

曾和 群馬大学の曾和ですが、データフローの概念を定義するのが、非常にむずかしいのですが、今お伺いしましたところ、MITのデニス教授が提案しておられるものを想像していらっしゃるのだと思います。

私の定義によると、データフローというのは、データの流れを中心にして考えていくコンピュータというふうに考えておりました。現在のコンピュータというのは、コントロールの流れを中心にして考えていくコンピュータというふうに、定義しております。

そういう定義によりますと、ご提案のデータフロー・コンピュータが、プロセッサ・ユニットの点では、マイクロプログラマブルなものを使われるということですから、そこでコントロール・フローが入ってきているわけなのです。

現在のコンピュータで言いますマシンレベルで、データフローを行なうコンピュータというふうなことだ

と思うのですが、こういうコンピュータを開発される場合に、データフローとか、コントロール・フローとか、なかなか区別しにくいところがあるものですから、私は名前の呼び方として、資源フローがいいと思います。資源というのは、データもコントロールもすべてのものをひっくくめているわけです。

そういう名前で呼ぶと、矛盾なく定義できるのではないかと思います。そのように考えますと、例えばレベルの違いこそありますが、論理回路、マイクロ・プログラム、機械語レベル、ソフトウェア、そういう点にデータ・ドリブンという考え方を、どんどん入れていったデータフロー・コンピュータになってもよいと思いますが、その点いかがですか。

田中英 おっしゃる通りだと思います。先ほどお見せしました図の中で、PEというのがプロセッシング・エレメントです。これがちょっと誤解を生んだのかもしれませんが、この中には、コントロールは入っていないと思っております。

つまりプロセッシング・エレメントと書いたら、その中には従来ですと、インストラクション・カウンタがおかれて、何かゴソゴソとやっているというイメージになってしまったのですが、あれはファンクショナル・ユニット(演算処理ユニット)が複雑に入っているようなものと考えて、外から機械語が与えられる。アクティビティ・コントローラから与えられるのは、まさに機械語であると考えております。

従って、そういう意味で、アクティビティ・コントローラが行なっているのが、コントロール自体というふうに考えております。少し図面の名前のつけ方で、誤解を生むところがあったかもしれませんが、そういうふうに考えております。

いろんな意味でのデータフローというのを考える、そういう意味でデータフローの概念を使うというのは、私も賛成です。

後藤 中間機の目的は、もしハードウェアの実装を試したいならば、データフローなどではなくて、実装だけ試せばいいと思うのですが、なぜコンピュータ・シミュレーションで、きちんとアーキテクチャを組み立て、一ぺんにつくらないのか、中間機をつくられるという意味が、よくわからない。

田中英 中間機と言われているのは、第2次バージョンの話ですか。

後藤 試作されるようなことを言われておりますので。シミュレーションをまずよくやって確かめないで、なぜいきなりマシンを試作するのですか。

田中英 データフロー・マシンとして、高性能のものを開発するのは、第2期で考えております。従ってトータルとしてまとめ上げるのは、第2期として考えておりますが、ひとつのファンクショナル・エレメントの制御、アルゴリズム — 全体の構造それ自体を検討するのは、第1期に考えております。

シミュレーションだけで第1期が済むかどうか、ということがもう1つのポイントだと思いますが、こういったことはシミュレーションでは、現在のノイマン型コンピュータの上では、高並列の非常に非同期的なものを、効率よくシミュレートすることは、完全にはできないと思っておりますので、ある部分はハード的に必要ではないか、第1期ではそういうところをテストするために、第1期のマシンをつくりたいと思っております。

サル データフローマシンについてお伺いします。

今までのところでデータドリブン・デスク립ションおよび、推論のレベルなど、そういう2つの両極端があると思うのですが、その間の関係のようなものは、わかるのでしょうか。

田中英 ご質問の意味がハッキリわからないのですが、ロジック・プログラミングと、データフロー・マシンとの関係をお尋ねになっていると、解釈してよろしいのでしょうか。

サル 今のお話でデータフローマシンが、推論に使われているというふうに理解したのですが、先生はすべてデータドリブンでやっていらっしゃるように思うのです。いわゆるデータドリブン、あるいはデータフロー・アーキテクチャというのは、アーキテクチャ・レベルで考えられますが、シーケンシャル・プログラムとも、ある程度関係があるのではないのでしょうか。と言いますのは、推論の結果を使いわけですから、データフローがシーケンシャルなプログラムの場合に、どのような影響をもつのかということについて、お話しいただけますか。

田中英 データフローマシンがシーケンシャルなアプリケーションに対して、何のメリットがあるかということをお聞きになっていらっしゃるのでしょうか。

推論はシーケンシングを含むことは、確かだと思いますが、あらゆるプログラムは、もちろんシーケンシングを含んでおります。ただデータフロー・マシンは推論を実行する上で、その推論が内蔵している、非常に大きな高並列性を能率よく発揮できるアーキテクチャだろうということで、そう考えているのです。

そういう意味でシーケンシャルなプログラム自体、

データフロー・マシンで実行することは、何ら不自然でないと思っております。

ウボルディエ 私は数学者ではありませんので、パラレルイズムの度合いとして、どこまで達成できるかということ、シミュレーションでやる、データフロータイプのアーキテクチャについてやるというようなお話があり、私は実はこの点について、ドイツで頼まれて、来年ペーパーを発表することになっております。

ホフマンさんの論文を読みますと、オプティマムとして、どのようなタスクならば、クリティカルな部分をアップデートするのにかける時間とプロセス全部の実行時間との関係を、調べることができるということが述べられております。これはリレーショナル・データベース・システムでは、かなり見られます。これをシミュレーションをする場合には、プロセスを順次つけ加えながら、クイック・ソートのようなアルゴリズムを使って、ブロックでタスクをやる、そしてタスクは、タスクを増やしたり、減らしたりするようなオーバーヘッドがかかります。

プロセスの数は、これから無限にあえていく、少くとも6倍位には増えていくだろうと思います。これは実際に私だけでなく、ほかの人達も研究して、文献の中で報告している通りだと思います。

そこで具体的な質問として伺いたいのは、どれ位の程度のパラレルイズムが、実現できるとお考えですか。例えば非常にむずかしいと思うのですが、6倍とか、20倍とか、色々限界があると思うのですが、実現性についてお話し下さい。

田中英 並列性の数ということについては、少くとも10程度のような並列性ですと、意味がないと、現段階では思っております。

従って、100とか、そういう類の並列性がこないと、完成されたフォンノイマン・マシンに、それ程太刀打ちできるようなスピードを出せると思っております。これは私の個人的な意見です。

ただデータフローの概念自体を、言語の中に持ち込むというのは、まるで別の話で、それ自体は非常にいい特性を持っているのではないかと考えております。

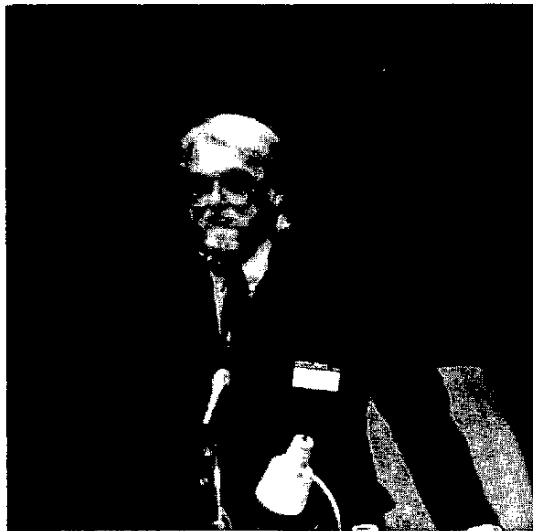
推論マシンがデータフローで実現されること自体、それは1つの証拠になっているのではないかと考えておりますが、ソフトウェアをつくる時に、大規模なものを簡単につくれる。知識の集約でつくれるといったようなことは、データフロー・マシンに非常に向いているのではないかと考えております。

従ってフォンノイマン・マシンの上で、従来のプロ

グラムを走らせた時と、データフロー・マシンの上で同じように書いて動かして、スピードがどれ位速いかということ競争することも1つあるかと思いますが、競争する場合は、並列性が少くとも100とか、そういう類の一定量以上必要ではないかと、現在考えております。

これはデータフローマシンができて、まだそんなに時間がたっておりませんので、従来のフォンノイマン・マシンが持っているような高度の、先取り機構、そういうものは、まだ色々入っておりません。ある程度 lazy evaluation ですか、色々考えられておりますが、まだ完成されておりませんので、そういう意味でまだ達成できないのではないかと、考えております。

従って2つの点が考えられ、ソフトウェア的に見れば、これは非常にいい特性があるという点、これが1つ。速度的にそういうものを除外して考える時は、これは別問題です。この2つを独立に考えております。



Ⅲ.2 招待講演

コンピュータ・ビジョンのための認識 アーキテクチャ

イリノイ大学 (米国)

ブルース・H・マコーミック

講演要旨

ここでの提案は、動画像の実時間解析のための視覚情報解析装置 (Visual Analyzer) についてである。

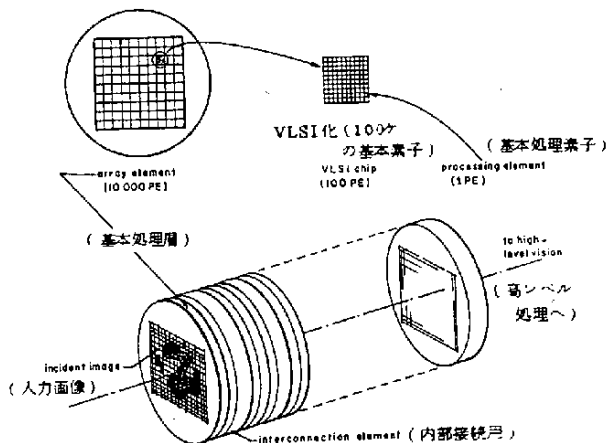


図1 視覚情報解析装置の概念構成

Visual Analyzerの概念構成を図1に示す。これは、生物の視覚系のモデル解析と最近のマイクロ・プロセス技術の発展から生まれてくるものである。特徴としては、ある瞬間毎(例えば、1フレーム間隔毎—33m秒)に入力信号を処理できるスタックから成り、全体としては連続的に変化している動画像系列を実時間処理できることにある。また、VLSI技術と生物視覚系の類似性から、視覚モジュールとして具体化でき、そして高レベルの視覚情報処理まで拡張できるものである。

このVisual Analyzerの応用は、工業用組立検査ロボット、航空機の無人操縦、深海ロボット、原子力プラント監視、リモートセンシングなどがある。

Visual Analyzer設計の基本となる考え方についてみると、まず処理対象となる画像の単位はひとつとする。時系列画像・大画像・マルチスペクトル画像・3次元画像なども、単位画像に分割(時間的、空間的)して処理すると考える。したがって、例えば動画像の実時間処理も図1のような解析装置の中を、入力画像が順に基本処理層を流れていくにつれ、処理解析が進んでいく(image-flow architecture)と考える。また、生理学的な視覚系モデル(視細胞から大脳皮質まで)は、VLSI技術とマッチする。最も具体化に近いものは、信号レベル処理回路(例えば、特徴抽出)である。そしてVLSI技術で最終的には、認知機構まで拡張できることになる。具体的なイメージは、図1にあるようにVLSIチップを2次元配列(CCD撮像素子のようにモザイク状に配列)したものを多層に結合したもの(3次元構造)と考えられる。層間の結合はファイバーオプティックスのようなもので接続され、画像データはこの中をパイプライン的に順に処理されながら流れていく。処理段数は最低5層は必要であり、各層での処理は1フレーム時間(33m秒)内に行なわれる。そして、時間的な関係も空間的な関係も保ちながら、画像を解析処理する全体構成になるべきでもある。

このVisual Analyzerの開発は、生理学的な考察と機械モデルとして実現化の点から行なわれなければならない。現在も、コンピュータを利用して研究を進めているが、それらは関係データベース、アテンション・メカニズム、分散型処理などである。図2に、時系列画像(例えば、交差点における自動車の流れをとらえた画像列)に対するアテンション・メカニズム

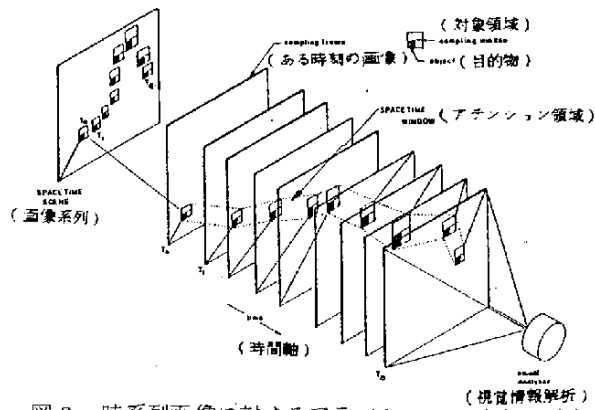


図2 時系列画像に対するアテンション・メカニズム

(attention-driven window tracking) を示す。注目すべき領域(大きさ可変)を動かしながら、その部分だけを詳細に調べていく機構である。また、画像解析・理解やその他の情報処理において知識として、2次元表現された情報を利用することは多い。例えば航空写真を解析するときの地図データ、眼底写真を理解するときの網膜内の構造などがある。この方向の研究として、空間的データ(2次元データを含む)ベースを用いた決定支援システムDIMAPがある。DIMAPのシステム構成を図3に示す。(第5世代コンピュー

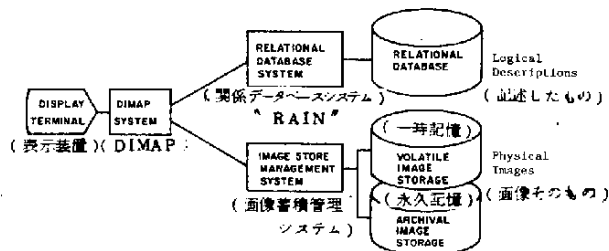


図3 空間的データベースを用いた決定支援システムDIMAPのシステム構成

タ知的インターフェイサー(図形画像処理部と類似)記号で表現されるものは関係データベース・システムRAINで取り扱われ、画像そのものは光ディスクに蓄積管理している。

以上、生理学的な視覚系の解明そして認知科学・VLSI技術による並列処理の研究開発により、ここで提案した視覚情報解析装置Visual Analyzerが一步步具体化されるであろう。

質疑応答

後藤 それはパイプライン的なものではないのですか。

マコーミック パイプラインアレープロセッサです。第3次元のジオメトリですから。

フロア外人 ペーパーの最初の部分について、質問があるのですが、先生はレイヤーの要素を、どの位タスクの内容によって変えられるかということについて、何かアイデアをお持ちでしょうか。

マコーミック インターコネクションの距離が、どれ位であればできるかとか、それがノンリニア(non-linear)なコネクションでもできるのか、あるいはリニアなコネクションだけでなければいけないのかとか、あるいはそういう変更が、ビジュアルチャンネルのhigher-structureのプロセスの結果出てきたような場合を、考えていただきたいと思います。

distributed structureから1つずつ入っていて、CRTスクリーンに何かの図を出したいというような場合には、ちょうどジェネレートバックするような形になるわけです。

私どもの経験では、ローカル・プロセッサ、このファースト・レベルのところは、ライン・オリエンテーションといったような、簡単なことをやっているわけです。

従ってグラフィック・ランゲージでも、低レベルのものが使われているわけです。次のレベルでは、むしろそれを超越するようなものをやらなくてはいけないわけです。

どういふものをやったらよいかは、まだわかりませんが、概念的には別に問題はないと思います。例えば道路と川と町の関係については、それほどむずかしいとは思っておりません。と言いますのは、そういうことをチャン博士がライン学のシステムでやったことがあるからです。

フロア外人 先生はハイパー・オブティックスをこのコネクションに使うということ、提案しておられるようですが、このコネクションをタンベリンに対して、adaptiveなものにするような案は考えておられますか。

マコーミック 私はそのようなことは考えておりません。

いわゆるインターコネクション・サーキットでのadaptationということ、提案しているのではありません。

田中 そのシステムでは、プログラミングは、どのようにお考えですか。

マコーミック プログラミングのことは、余り考えたことがないのですが、アレープロセッシングあるいは

イメージセンシング (image sensing) で行なっていることについては、少し経験があります。

色々な研究報告が行なわれておりますが、ベーシックなプロセッシングは、フォンノイマン型のマシンでも行なわれております。それらについてはかなりの文献が出ていますと思います。

イメージプロセッシングについては、1年に800件位の文献が出ております。またニューロビジョン (神経学的な視覚の研究) あるいはサイコロジカル・ビジョン (心理学的視覚の研究) については、600件位の文献が出ております。

ただ両方を対照してみる人は少ないと思います。

曾和 画像を処理する場合に、各層によって、処理を進めていこうと思うのですが、その場合プロセッシングエレメント毎に、お互いのネットワークというようなものが必要になるのではないかと思います。

並列処理の計算方法のことを考えてみますと、いちばん問題になるのは、結局プロセッシング・エレメント間のインターコネクションが非常に複雑になるということですが、人間とか、生物の場合、その辺は完全接続なのでしょう。

先ほど後藤先生の質問の答では、その辺はパイプライン処理だと言われましたが、パイプライン処理と言いますと、処理系にパイプがいっぱいあって、1つのパイプに入った原因は、ずっと最後のところに出ていく時に結果が出るというふうになっております。

そのパイプとパイプの間に、接続がいっぱい複雑にあり、それで処理していくという、そういう構造にはなっていないのでしょうか。

相磯 今の質問をもう一度言います。先生のお話ですと、色々の処理の階層があり、それぞれのレイヤーの処理は、パイプラインでされるのがいいだろうと言われておりますが、層と層の間には、複雑なネットワークが必要ではないだろうか、ということですか。

マコーミック 例えばライオンレイヤーいわゆるビジュアルコortex、視覚皮質のようなところに来ますと、そこには1つのカップリングが、隣接パイプとあるわけです。

従ってそのコントロールを分散していく必要があります。イメージは縦に入っているわけですから、そのイメージがずっと入ってきますと、イメージが層でかなり分散されていく形で、いわゆる3次元的な分散、スプレッドというものがなければいけないわけです。

私がここで言いたいのは、ビジュアル・システムをつくっていくということは、それをそのまま真似して

しまうと失敗してしまふわけです。

私がこの段階で考えておりますのは、ビジュアル・アナライザとして、1つのデザイン・ルールをまずつくって、3百年もかかって研究されてきた神経学のほうから、わかっている内容をきちんと説明してもらいその中で使えるものを人工知能の中で活用していかなければならない、というわけです。

それをまた次世代のマシンの中で、反映させていかなければならないと思うわけです。正しいエンジニアリングの背景を持った人が、どういようなものが活用できるかということ、まず見定めなければいけないというのが、私の意見です。

相磯 少し別の質問かもしれませんが、先生は生物学的なアプローチも重要だと、指摘をされておりますが、生物にはよく学習の機能、learningの機能があるといわれますが、その辺のご関係について、何かありましたら、お願いします。

マコーミック 色々なアダプトなメカニズム (adaptation mechanism) をリレーショナル・データベースの中に入れることができると思います。リレーショナル・データベースと言いますのは、数秒前のプロセッシングの内容が入っているものです。次にまたほかの点を見るわけですから……。

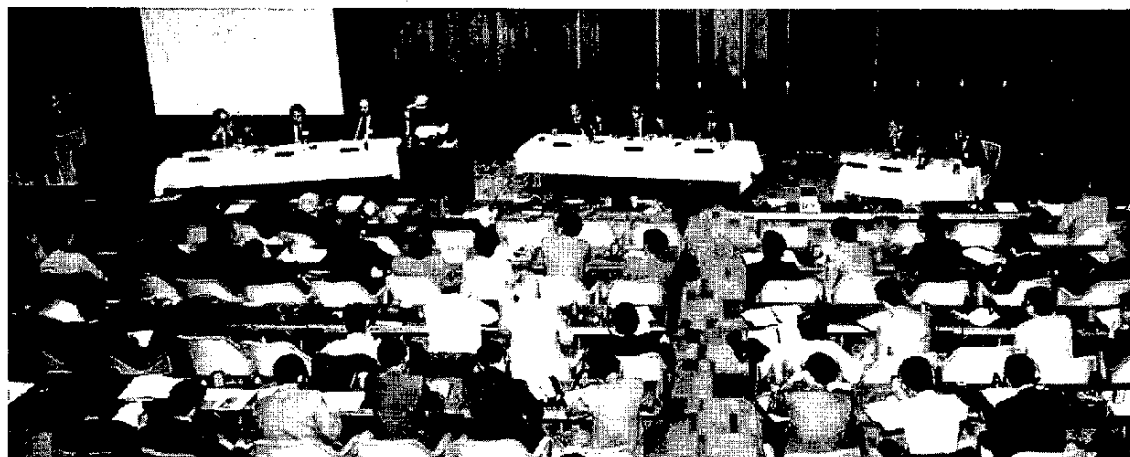
1つの adaption として考えられるのは、アテンション・ドリフトのメカニズムです。つまり何かが必要であるということがわかります。例えば、あなたが何かの動物であって、空に鷹が飛んでいるという時には、そういうリレーション・チャンネルが働いて、それに気がつくわけです。

ですからそういうアテンション・メカニズムにも、そういうフィーチャを大切にしなければならないと思います。リレーショナルテーブルの各カラムでも、そういうアテンション・メカニズムというものを考えなければならぬと思います。

アダプティブ・メカニズムが全く不要であるというのではなく、やはり問題は、アダプティブ・リトルメカニズムをどのようにつくるかということではないかと思ふます。

Ⅲ.3 パネル・ディスカッション I

知識情報処理



座長：電子技術総合研究所 淵 一博

パネリスト：スタンフォード大学 エドワード・A・ファイゲンバウム，京都大学 長尾 真
ミュンヘン工科大学 ウルフガング・ビーベル，東京大学 大須賀 節雄
INRIA ギルス・カーン，電子技術総合研究所 古川 康一

淵 知識情報処理システムについてのパネル・ディスカッションを始めたいと思います。パネラーのご紹介をします。向こうから3人は昨日の講演で皆様もご存知のファイゲンバウム教授、ビーベル教授、カーン博士です。次が京都大学の長尾教授です。長尾先生は自然言語処理、とくに機械翻訳であるとか、言語理解システムの研究では日本の第一人者であり、パターン認識、画像処理の方面でも日本の中心的な研究者です。

それから東京大学の長須賀先生です。長須賀先生は古くからオンライン・システムの研究、コンピュータ・グラフィックス、マンマシン・コミュニケーション、データベース等々の研究をされて、知識工学といえますか、知識情報処理の提唱者としては日本でもっとも早く、大部昔から提唱されてきた方です。

その次は昨日でご存知の電総研の古川さん、私は電総研の淵です。

最初に、私どもが昨日と今日にかけて、研究計画をご報告したのですが、それについてのコメントを3人の先生方をお願いしたいと思います。

昨日のご講演の時も、ある程度言及されましたが、本日の報告を含めて追加したいコメント、あるいはご意見がありましたら短かい時間をお願いしたいと思います。

それではファイゲンバウム先生、研究計画について

何かお話いただけますでしょうか。

ファイゲンバウム 私は以下の3つの点について簡単にコメントさせていただきたいと思います。一般的な意味で申し上げますが、あとでパネル・ディスカッションの時にもう少し具体的に掘り下げてお話ししたいと思います。このパネル・ディスカッションは2時間ですので、あとで詳細な話を入れないと、時間がもたなくなってしまうですし、その点が本質的なところだと思います。

まず最初のコメントとして申し上げたいのは、人工知能の実験的な研究と知識工学の応用研究についてです。

システム・デザインの基礎となるような理論はいろいろなレベルであります。アプリケーション、あるいはソフトウェア(OS等)、あるいはハードウェアのいろいろな段階においてあります。

従って、次に何をすべきであるかということを見抜くことは経験的な作業です。私どもはここ2~3日にわたって、かなりのシステムの図を見せていただきました。

こういう素晴らしい研究開発の中で、どのようなベースがあったのかということを知ってみたいところ、ベースというのは日本にはないというお返事が返ってきました。

例えば、田中穂積先生から昨日もいただきました文献の中には知識工学の研究はごく最近日本で始まったばかりであると、その文献の中で述べておられます。

そして、田中穂積先生は2つのシステムを引用しておられ、2つのシステムはいずれも小さな医療診断システムです。私の話の中で上げたリストのうち、日本で研究されているのはひとつもありませんでした。

これは悪いということではなく、これはまだ初期段階ですから。ただ私はちょっと気になるのは、もし私が通産省の企画者であれば何億ドル、あるいは何億円という資金のかかるプロジェクトをごくわずかな経験をベースにしてつくっていくということが気になります。

グラウンドでデザインをつくっていく上で、なぜどの要素を選んだかということが十分に説明できなければならぬと思います。

どういふ研究からどういふタイプのアーキテクチャが必要になったか、あるいはどういふタイプのソフトウェアが必要になったかというような話が必要であろうと思います。

このプロジェクトは芸術のプロジェクトではありません。科学、技術、工学のプロジェクトですので、それにははっきりとした理由がなければいけないわけです。機能もなければいけないわけで、すっきりとした見かけがいいだけでは充分ではないということをおひとつのプロジェクトへの警告として申し上げておきたいと思っています。

さて、2番目に申し上げたいことはハードウェア・アーキテクトの立場からいうと、このプロジェクトのニーズをよく考えないといけないと思います。単なる機械というふうに考えるのではなく、今世界に何についてもかなりの大掛りな大規模な進歩が遂げられるような機械が揃っていると思います。

今朝の説明ですと、100ギガバイトのディスクが必要であるという話が出ましたが、知識工学の研究者が100ギガバイトのメモリー、あるいは100ギガバイトのコミュニケーションレイトを必要としているということを誰がハードウェアのアーキテクトにいったのでしょうか。

エキスパート・システムはとくに専門的なものであり、従って、知識ベースが狭いものです。しかし、ハードウェアの人たちに対してなぜそれが必要だと誰がいったのか、どういふシステムがハードウェアを必要としているのか、ニーズを知る必要があるわけです。時機ではなくて、ニーズを狙ってハードウェアという

ものは開発されていかなければならないと思います。

3番目に私が申し上げたいことは、われわれは実情というものに直面しなければならないと思います。第5世代コンピュータ開発計画というのは大変素晴らしいものであると思いますが、現実を見失ってはならないと思います。

第5世代コンピュータ開発計画というものは大変創造性の高いものですから、いろいろな分野において花が咲く可能性があると思います。だからこそ、このプロジェクトはいくつかのアイデアに焦点を絞って進めていかなければいけないと思います。

すべての要素を推進していくことはできないと思います。私は3段階に分けて計画の練り直しが必要であろうと思います。計画の練り方として、ステージ1は explore、ステージ2は focus、ステージ3は perfect と、3つのステージではありますが、こういう分け方にした方がいいと思います。

ステージ1が2年、ステージ2が5年、ステージ3が3年という分け方を考えてみたいと思います。私ここで申し上げたいポイントは日本のエンジニアの方々が大変素晴らしい方であるということはおわかってはいますが、何でもできるというわけではありませんし、またそれがすべてできるだけの資金があるというわけではないと思います。

しかし、少くともコンピュータ・サイエンティストの人数がすべてをやるには少なすぎるとお思いますので、ステージ2で焦点を合わす必要があるということをお申し上げたわけです。

この素晴らしい目標を達成するためには現実的な目標といえますか、現実的な焦点の絞り方をしなければいけないというのが私の意見です。有難うございました。以上です。

淵 貴重なご意見をいただきましたし、多分お答えしなければいけない問題もあるかと思いますが、ひと通りパネルの皆様にご意見をお伺いしたいと思います。

ビーベル 今、ファイゲンバウム先生がおっしゃった最後のところをとらえて、述べたいのですが、このようなプログラムを全部やるための人数が足りないのではないかとおっしゃいました。とくにこの点について述べたいと思います。

私もそのような感じをもっていました。実際のマンパワーの面で問題が起きてくるのではないかと。このような巨大な、素晴らしいプロジェクトを解決していくだけのマンパワーがあるかどうかということです。

どのようにしたら私たちはその経済性というものを

達成することができるか、必要なマンパワーに関しての経済性の面です。具体的にいきますと、私から申し上げたいのはよりチャレンジングな価値のある問題というのは、このプロジェクトをつくる上で、昨日申し上げたことですが、セルフ・モディフィケーションが重要だと思います。

この点こそがプロジェクトについてのもっとも挑戦的な課題です。数年間に最初に実現されるもののいかんによって、もう2~3年遅くなってしまうと思います。そういうものがなければ…。

2つのことについてコメントし、提案したい。ひとつは昨日申し上げたことで、すでに解決されたことかもしれませんが、例題評価 (example evaluation) というので、例題評価、モディフィケーションというのを問題解決のところで繰り返し述べることになりますので、申し上げますが、これが第1点です。

もう1点はデータベースのところで、昨日私は余り話さなかったのですが、データベースに関して少しコメントを述べたいと思います。

昨日のファイゲンバウム先生の話に戻るのですが、知識の獲得ということが一番肝要なものであるということでした。私たちがさらに知識の獲得を高めるためのツールを得ることができるかということ、ツールを見つけることができるかどうかという問題について、人間がやってきたパラダイムについて役に立つものが考えられるのではないかと、人間がどうやって知識を獲得しているかということからひとつのパラダイムが得られるのではないかと思います。

人間はどうやって知識を得ているのでしょうか。広範囲なやり方があります。一方の端には現実的なブリベアード・ノレッジ・トレーニングコース、即ち学校での講座があります。

もう一方の端は昔から学者が部屋の中で考えたように思考するということがあります。この2つの間には広い広がりがありますので、人によっては考えること、あるいは論理的な部分が重要であるという人もいます。

勿論この点は重要になります。それによって両方を考えていくことが必要だと思います。ここで小さな図をもってきたので、それでお話しします。

パワーが演繹的な力の方ではすなわち縦軸の方は小さいとします。そうすると、面積自体は非常に小さくなってしまいます。(図2-①)

また知識の方が少して、演繹力が大きくても、面積は小さくなります。(図2-②) ここで一番いいのはバランスの

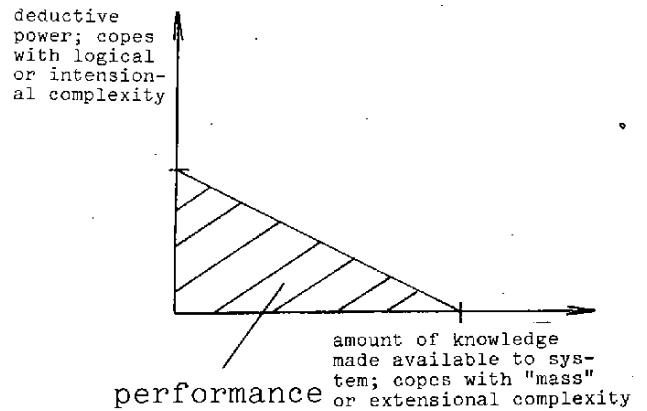


図1 Trade-off between

deductive power & required knowledge

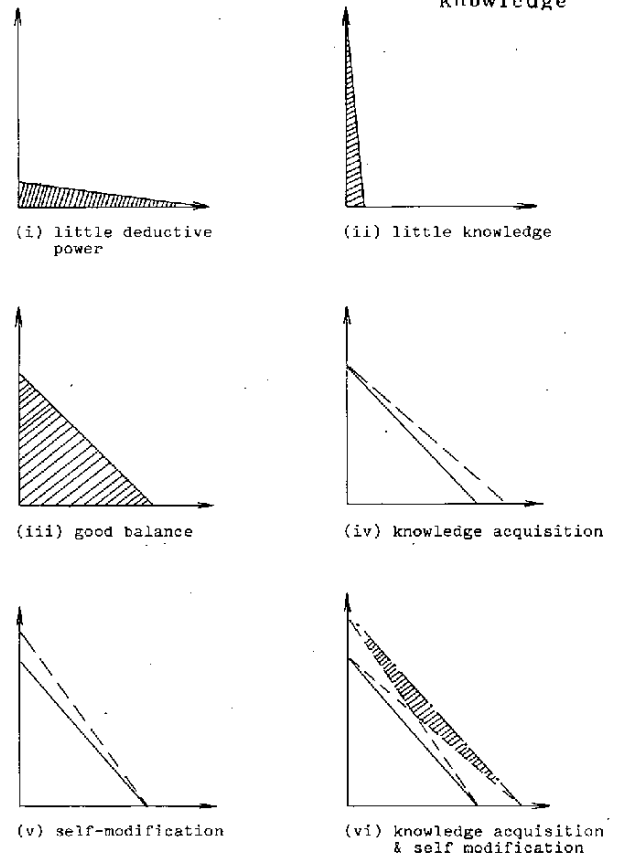


図 2

とれたやり方で、演繹的な力と知識の力をバランスのとれたものにするということによって、一番広い面積が得られます。これが広い面積だということはパフォーマンスが高くなるということです。(図2-③)

その他の図は例えば、知識の獲得のためのツールがあったということになると、システム自身がその知識自

体とツールによって増やすことができます。そうすると、点線で書きましたようにエリアが広がります。(図2-④)

また、セルフ・モディフィケーション、問題解決のツールによって、その力が加わるとこのように広がり、改良が見られるということを示しました。(図2-⑤)

私の意見ではそのような方向で、集中して研究するのがいいのではないかと思います。

私たちはロジック、論理が必要です。ロジックを知識の中に入った形でもつこと、すなわち知識ベースをロジックと結びつけることが必要です。

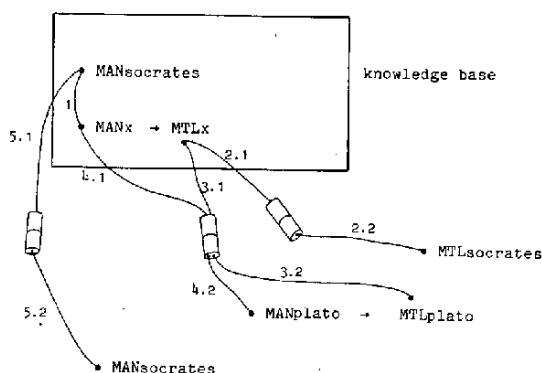


図3. Logic in the knowledge base via the connection method

ここで昨日使った小さい例、ソクラテスが人間であって、人間は誰でも死すべきものであるという例ですが、この2つのアイデアのコネクションを昨日話しましたが、このような知識ベースでこういうコネクションがありますと、ある1つの要求が与えられると、小さいノードからブラブインすると、それ以上の定理証明せずに結果を得ることができます。

準備された状態の知識ベースがあるので、ハードウェアによって、これがすべて人間の介入なしになされるということなのです。そうすると、リクエスト、つまりソクラテスが死すべきものであるというものが入る。このコネクションがここにありますので、自動的にソクラテスが死ぬべきものであるというものが出てきま

す。

左の方から入れますと、真中のところを求めると、2人のエンドがあります。オープンエンドになって、どういう英語かわかりませんが、定理証明がなされるわけです。

ロジックを知識ベースの中に取り込むということによって、昨日ではファイゲンバウム先生がおっしゃったことですが、アナロジーによる概念形成ということが非常に重要だと思います。

また他のところで出てきたことですが、リトリバル・オブ・インフォメーションということ、とくにこれらの点が演繹的推論ということ、非常に重要だと思いますし、アナロジーというのは効率的にディダクション、抽象化によってなすことができます。

例えば、ブライスタードの研究、イリノイ大学の方ですが、また概念形成の場合にはラストに近いです。こういう業績が出ていますので、いろいろなレベルの抽象化を用いて、それを使って情報の検索ができます。

これをバンドル化する、束にして抽象的なものにして、いくつかのレベルの抽象化の程度があって、それを一緒にして効率のよいデータベースにすることができるということです。

こういうことになると、ハードウェアによってそれが表現されることは可能だと思います。ご清聴有難うございました。

淵 ハードウェアとの関係については、多分第2ラウンドか第3ラウンドでコメントをお願いしようと思ったのですが、次にカーン先生お願いします。

カーン まず最初に日本側の主催者の方に対してまして、感謝を申し上げます。日本側でのR&Dの計画について、こういう形で教えていただくということはまれなことです。こういう機会に恵まれたということ有難く思っています。

ここで私の個人的な意見、いかなる組織の意見でなく、個人の意見として申し上げたいと思います。2つのグループに分けて申し上げます。

まず最初に研究計画についてです。この研究計画は目標志向型ですから、何を狙ってられるかわかりますし、トップダウンでどういうものを狙っているかわかるようになっていきます。

従って、貴重な戦略であると思います。研究項目の中には他に動機づけがあって、さらに研究されていくようなものもあると思いますが、ここでは戦略に合せて考えられていると思います。

各分野の論理的な動機づけというものを無視するこ

とはできないと思います。研究者に聞けばどういふ問題を今、現在抱えているか、そして、これを解決するためには、各研究者がどういふ順序で研究していかなければならないと考えているか、ということについては各研究者に聞くことによってわかってくると思います。

そうすることによって、貴国の計画もより鮮明なものになってくると思います。

さて、次に個人的に申しまして、今ここで取り上げられておられるトピックというのは大変難しいものであり、これを達成するためにはかなりの基礎研究が必要であると思います。

コンセプトのフォーマル化というのもまだ早期の段階にあると思います。例えば、使われる用語についても意見がくい違っているところもあると思います。

各言葉の使い方についても、もう少し形式化していくことによって、われわれが話し合っている内容が同じものであるかどうかということを確認していくことも必要であろうと思います。

経験から申して、ヒューリスティックについて具体的な興味をもっている研究者たちの中でも、数学的背景をもっている人たちはどちらかというと、ジャンプして行動するというよりも、ものをフォーマル化して考えたがるのです。そういうことで私はこういう意見を出したのかもわかりませんが、まずそういう点を考えています。

もう1点付け加えたいのですが、コンピュータ・サイエンスの研究分野の中で、私にいわせれば貴国の計画では軽視されているところがあると思います。

それはアルゴリズム・デザインの面での Complexity theoryです。信号処理におけるアルゴリズムというものは考えられています。これはよいもので、このアルゴリズムは他のアルゴリズムよりも優れているということを立証することは難しいと思います。

信号処理の場合には効率というのが大変重要で、効率というのはスマートな頭の良い、いいアルゴリズムを使うことによって達成できるものではないかと思えます。

記号処理の効率を上げていくためにもアルゴリズムの活用が必要であろうと思います。もうひとつのアプローチとして、知識ベースのフォーマライゼーションということも考える必要があると思います。

私の意見では書き換えの理論というのは難しいものであると思います。書き換えるルールというものは難しいものであると思います。非定型的な、数学的

な難しさをはらんだ問題であると思います。

大きなルール・ベースを使う場合にそれをどういふふう構成していくかということと考え、またルールをどういふふう並べていくかということを考えてつくっていかないと、複雑さが増すために、却って問題が出てくるのではないかと思います。

どれだけのハードウェアを使いこなしても複雑さのために問題を解決できないということが出てくる可能性があると思います。

ファイゲンバウム先生の先程のご発言について、付言させていただきます。とくにランド・プログラムにおけるプログラムの作成についてですが、記号処理というものが産業界の中に浸透するには時間がかかると思います。

産業界の中で、先端技術が浸透していく上で記号処理というのはより難しいものであると思います。

もうひとつ、ソフトウェアの側面について申し上げます。ソフトウェアの専門家がハードウェアの専門家についてどういふことをいいたいかといいますと、先程も指摘されたことですが、ハードウェアのデザインをする時に広範な細かいシミュレーションなしにデザインするという事は適切ではないと思います。

ハードウェアの中にはいろいろなものを入れることができますが、ソフトウェアの人たちがどういふことを希望しているかということが充分に理解されてつくられていないきらいがあると思います。

われわれにコードをつくらせてくれといわれても、コードのつくり方のわからないものをつくらせてくれといわれては困るわけです。問題の核を明らかにする必要がありとう思います。

今回の会議ではユニフィケーションというのがハードウェアのインプリメンテーションの技法としては非常によいものであるという話が出ましたけれども、私はそうは思いません。

ユニフィケーションというのは2つの部分があると思います。1つはマッチングであり、もう1つはサーキュラリティのテストです。

もしマッチングをインプリメントするならば、これは汎用のものになります。しかし、ユニフィケーションをインプリメントしますと、アプリケーションをかなり制約する可能性をもっていると思います。

また、今ここで予想されているデータ・リラクシオンと違うようなものを使おうという人にとっては使いにくいものになってしまうかもしれません。

パラレルリズムについては、私自身研究しています。

淵 ユニフィケーションについて2つの事柄があるといわれましたが、もう一度言っていただけますか。1つはパターンマッチングで、もう1つは何ですか。

カーン もう1つですか。ユニフィケーションの半分はサーキュラリティの部分です。サーキュラリティの問題です。

もし照合すればいいわけですが、照合しない場合が非常に多いのです。その点をとくに申し上げたかったのです。

はっきりいわずに申し訳ございません。

私はパラレリズムについて研究していますので、ひと言申し上げます。記号処理ではパラレリズムは余りはっきりは見えません。シンクロナイゼーションのプリミティブを観察しますと、これは非標準的なものです。denotationalなプログラミング言語PL-O-LOGのものが様々な文献で研究されていけませんが、これはレベルが低く、まだこの日本の研究計画で使えるところまではきていませんから、全くまだ未開発の分野であると思いますので、ここもさらに研究の必要な分野だろうと思います。一応ここまでにしておきます。以上です。

淵 沢山の問題提起をされているわけですが、まずはひと通りご意見をお伺いしたいと思います。次に長尾先生に。

長尾先生は第5世代の調査研究委員会のメンバーとしていろいろ助言をいただいていたのですが、しかし、それを含めて計画全体でもよろしいですし、招待講演のことに関するコメント、あるいはご質問でも結構ですし、あるいはもっと一般的なことで結構ですのでよろしく願います。

長尾 この第5世代のコンピュータシステムをつくっていく上において、一番大切なことは何かということについて、いろいろ議論がこれまでなされてきたわけですが、私はファイゲンバウム先生と同じように、知識というものを如何に活用していくかという問題を取り上げて少し考えてみたいと思います。

その場合、知識というものをどういふふうに整理してコンピュータに与えるかということが一番大事な問題になります。整理の仕方は時代によって、また学問の進み方によって変る。現時点においてはこういう見方で整理をするのがよい。しかしあと5年たてばまた別の見方で知識を整理しなければならないことになるかもしれません。

巨大な量の知識をコンピュータの中に蓄積して活用しようというわけですが、その蓄積の仕方が根本的に変

わってしまうという危険性をもつこととなりますから、それをどういふふうに解決していったらいいかということを実際に考えねばならないということに注意しておきたいと思います。

それを例えば、機械翻訳のような立場で考えてみますと、機械翻訳で一番難しい問題の一つは意味をどういふふうに扱うかという問題になります。

今考えられている1つの方法は意味素というものを考える。意味素をいくつも取り出して、例えば、ある単語はhumanであるか、non-humanであるか、あるいはmaleであるか、femaleであるか、あるいはconcreteであるか、abstractであるかというように分けていくというような考え方がありますが、そういう意味素のつくり方というのが本当にいいのかどうかというのは、恣意的な、仮設的なものでありますから、これをずっと続けていくことができるかどうかはわからないということになります。

ところが一方、人間はどういふふうにして翻訳をマスターするかということになると、日本の場合は中学校に入って英語を習う。その時にどういふ習い方をするかを考えてみると、要するに沢山の例文を与えてもらっている。

例えば、簡単なところから考えますと、He is young. John is young. Mary is young. というふうにいろいろ与えて、日本語訳の方もどんどん与えていくと、どういうことがわかってくるかということもいろいろ教えるわけです。

そうすれば多くの単語が同意語であるとか、反意語であるとか、関連語であるとかいうことが自然な形でわかってくる。それを最初から人間が意図的・恣意的に分けてしまうということがいいかどうかというところに問題が出てくるわけです。

例文とその対訳という関係の情報は何年たっても正しいのですが、知識をコンピュータに与える場合に、ある観点から解釈した知識を与えるということは、コンピュータにとって能率のいいことではあります。それが長年本当に正しいとして保たれていくかどうかということに関して疑問が出てくる。これはデータベースの世界で得られた深刻な問題でもあります。ある種の整理の仕方で何年かかかって作ったばう大なデータベースが、完成された時点で役に立たないとして捨てられた例はいくつもあるわけです。

翻訳を考える時には単語の選択の問題が重要になります。例えば「破れた」というのにはいくつもの用法が出てきますが、例えば、「大統領は投票に破れた」という時に、brokenとdefeatedのどちらの破れ

たを採用するかというのはそこで使われているいろいろな単語、たとえば大統領、投票といったものが「彼は選挙に破れた」というのと「紙袋は重みで破れた」というのと、どちらに近いかというような、ある種の連想に基づいた考え方で処理するのがいいのではないかと考えられます。

こういうことを最初から意味素を設定してきちっとやることができるかどうかは難しい問題だし、そういうシステムをつくるよりは、人間的なアナロジーによる翻訳というのを考えた方がいいかもしれないということになります。

「投票の結果、大統領の敗北が明らかとなった」という文が与えられた時に、いろいろなスタイルで訳することができるのです。そういうこともいろいろな例文的な要素からやるということが必要になってくるのではないかと考えられます。

広範囲の大きなシステムを考えていく場合に長年にわたっての信頼のできる知識とは何か、それをどういうふう to 獲得していったらいいか、あるいはどういうふう to 増やしていったらいいか、それをどういうふう to 使いこなしていったらいいかということを考える必要が出て来ます。言語処理とか、翻訳の場合には先程から出ていますようなパターン照合であるとか、連想的な機能と推論の機能をもたせるとか、検索を連想的にうまく行なうことができないかという問題とか、例文をどんどん与えることによって知識を増やしていくことができるのではないかといった問題を考えなければならぬのではないのでしょうか。

システム、あるいは知識ベースが巨大になってきますと、システムティックに知識を与えることは非常に難しくなります。それに対して例を与えるということはそれほどむづかしくない。また例は常にある範囲内においては正しいのですから、そういうものをうまく活用するというような柔軟性をもったシステムを考える必要があるのではないかということが、私の指摘したい点です。

淵 またありましたらあとで追加していただきます。それでは大須賀先生お願いします。

大須賀 一昨日以来のお話を伺っていて、大変興味深かったのですが、私のこれからの話は今までの夢のある話に多少、水をかけるといった形になるかもしれません。ただし私自身の立場は知識ベース・システムというものに対して関心があって、研究を進めているというものであるということをお断りしておきます。

昨日来から第5世代のコンピュータに対する枠組みというものが議論されてきました。これからはもっと内容をつめていかれる、具体的なものになっていく段階だろうと思います。

その間において沢山の可能な手段があると思います。問題なのはこれらの可能性の中でどのような選択を行なっていくかという問題になると思います。

この選択が任意に行なわれるものではなくて、ここに必ず必然性がなくてはいけない、その必然性というものを与えるための根拠というものをわれわれはこれから追求していかなければいけないと思います。

それにはいろいろなことがあります。私は昨日からの話で、ここで2つだけ今日までの話で出てこなかった点を指摘したいと思います。

1つは第5世代というからには現在のコンピュータ・システムにかなりの程度置換わるものであるというニュアンスを含んでいると思います。これは単に興味あるアプリケーション・システムをつくるということとは違って社会的に見て、これをつくる組織と、それを使う側という役割分担がでると思います。

知識情報処理システムは、ファイゲンバウム先生がおっしゃったようなエキスパート・システムのような、ユーザにとって便利なシステムでなくてはなりません。ここでいう汎用なものとはいうのはメーカの立場から見て特定のアプリケーションに対して手当をしなくてもすむもの、沢山のプログラムをそのためにつくらなくてもよいものという意味で、もしそうでないとしたら普及が困難であるということは当然メーカとしてお感じになると思います。

このようなことを私は汎用化とひと言で申しますがこういうことを追求することができれば、それは一種の量産化が可能になり、低価格にもなり、普及するだろうと思います。

これは単にこれをつくるというシステムの問題ではなくて、そのような形にできるためにどのようなシステムとするかという問題を多分に含んでいると思います。

もう1つはわれわれが考えておかななくていけないのは、第5世代のコンピュータは10年後の実現を目ざしていますが、その時に果してこのようなシステムがこれまでわれわれが使ってきたコンピュータとどういう関係にあるのかということです。例えば、知識情報処理システムというものがあらゆるアプリケーションに対して現存の手続型といわれているコンピュータよ

り優れているならば、それは全面的にとって変わるものになるかもしれません。

逆にいろいろな努力にもかかわらず、現存のコンピュータのスタイルの方がいいのだということになってしまうと、この計画は消滅してしまうことになるでしょう。

現実には恐らくその中間にあると思います。だとすれば、どのようなアプリケーションに対しては現存のものがよくて、どのようなアプリケーションに対して知識情報処理システムというものがより有利であるのか、恐らくこの2つは相補的なものであって、お互いに弱点をカバーするような立場で社会に広がっていくというのがもっとも自然の姿だと思います。

安定した姿になるまでには両者が競合するでしょう。あらゆるアプリケーションの中には処理速度を最高に上げたいというアプリケーションが常に存在して、それは知識型といわれているようにマッチングを主体とするような計算よりも手続さえ前もって明確に定義できれば、手続で表現した方が明らかに早いというものが当然存在すると予想されますが、知識型のシステムというのがどういう場面において現存のシステムよりも優れているのかということを確認する必要があります。

この例はいくらでも上げることができます。昨日のファイゲンバウム先生の話にありましたように、知識工学のいくつかの例を挙げるすることができます。それを含めて、汎用化という先程の立場からすると、もっと広い領域、この領域は知識ベース・システム、あるいは知識情報処理システムが優れているといった領域というものをおわれわれは設定したいと望むわけです。

恐らくその中によくいわれるのに、非定型処理、前もってその手続が100%定義できないような処理、ヒューリスティックな処理、人とマシンが協力し合わなければならないような応用というのがあると思います。

その例として、エンジニアリング・デザインであるとか、医療診断であるとか、意思決定支援であるとか、研究開発自体にコンピュータを使おうというような種類の問題が出てくるわけです。

このように2つの面を上げたのですが、これらはいずれも純粋に技術の問題ではありますが、先程申しましたように、知識システムを実現する上で考えなくてはならない多数の可能性の中で何を選ばなければならないかということに対する基準を与えるのではないかと思います。

今までいろいろ述べてきましたが、つきつめていきますと、知識ベース・システムについて私は4つの点が重要であると思います。知識を如何に表現するか、如何に知識を利用するか、如何に知識を獲得するか、最後に如何に知識を管理するかという問題です。

先程来私がいろいろ述べてきたこと、またそれ以外の話も多分にこれらの問題にかかわってきますが、ここではこの中で知識の表現と知識の管理の2つだけ取り上げてみたいと思います。

知識表現というのは最終的にすべての問題にかかわってきますが、先程あげました非定型的な処理、例えば設計という問題を取り上げてみます。この問題は単に知識があるからだけで必ずしも解決する問題ではないのです。

知識情報処理システムというものが設計という応用に対して既存のコンピュータに優れているためには、設計という仕事に対して人間が必要とする機能をより良く提供するものではなくてはいけません。

設計という仕事はある対象を素材から作り上げていくプロセスとすれば、当然それをコンピュータ化するためには複雑な構造を操作する機能をもたなくてはならないはずで

そういうような知識の表現形式というものをわれわれは果して持っているでしょうか。われわれはこういうことをもっと掘り下げて研究しなければならないと思います。この問題は、非常に重要な問題だと思います。これが1つの指摘です。

もう1つの問題は知識の管理の問題です。コンピュータを使いやすくすることによって多くの人がコンピュータを使い始めたとしましょう。しかし知識というものをに入れて使い始めたのに、その知識ベース自身の管理が不十分で、何が出てくるかわからないということがもし起こったら、その知識システムというものは信頼性を失なうでしょう。

コンピュータが使いやすくなるということによってユーザの層がどんどん専門家から離れていくという状況を考えますと、ユーザ自身にその知識を管理する、保守するという能力というのはかなり少ないと思わないといけません。

そうすると、必然的にシステム自身が知識ベースを完全に管理し、無矛盾を保ち、冗長性を減らすという機能をもたなくてはならないだろうと思います。

逆にそういうことができる知識の表現形式はどのようなものかそれに対する管理の方式は何かという問題が当然起こってくるわけです。

私は最初に2つの面を上げて、そのことからこれから問題点をいくつか出してきましたが、私はこれからこのような討論が行なわれることを期待したいと思います。

今、私が上げました以外にも沢山の観点があって、それらの観点に基づいて、可能性の中から一番いいものを選んでいくというプロセスを早急に実行する必要があるのではないのでしょうか。

淵 現在の委員会のメンバーの中から古川さんに出ています。いろいろコメント、ご意見を寄せていただいております。中には答えにくい問題もあるようですが、そのへんは適当にパスしていただいて結構です。

昨日の報告を含めて補足したいこと、最後に結構ですが、問いかえしたいことがありましたらどうぞお願いします。

古川 ファイゲンバウム先生を初め、カーンさん、ビーベル先生、大塚貴重など意見をいただきまして有難く思っています。

われわれ自身、このプロジェクトが野心的である、基礎研究を広範囲にやらなくてはいけないということは認識しているわけです。とくに日本でこの分野の研究が必ずしも進んでいるわけではないということも認識しています。

1つの問題はファイゲンバウム先生とビーベル先生がご指摘になったマンパワーの問題があるかと思いますが、それに直接答える形になるかどうかわかりませんが、それを絡めて初日、2日目を通じた議論にWhy Logic Programming ということがあったと思います。

ロジック・プログラミングを取り上げているということに対してかなりの方が唐突であるという感じをもたれているかも知れませんが、われわれはいくつかの点でこれを非常に大切なものであるというふうに考えています。

その1つは非常に性質がよろしいということです。メカニズムがシンプルであるということがいえます。そのことはとりもなおさずハードウェア化が容易であるということにつながります。

もう1つ、LISPで開発されたいろいろな技術というものを素直な形で引き継ぐことができるということです。

2番目は特定のPROLOGということですが、インプリメンテーションがよろしい。例えば、ストラクチャー・シェアリングであるとか、ローカル・スタック

を使ったりまいストレージの管理などがPROLOG自身を使える言語にしているといえます。

3番目はPROLOGカルチャー自身が急速に、これは見方によるかも知れませんが、成長している。関係データベースというのは1971年にIBMのコードさんによって発表されましたが、それが急速に発展したのと同じような成功が期待できるのではないかとというのが3番目のポイントです。

LISPではなくて、なぜPROLOGかという点について2つ程簡単にいいます。PROLOGはPure LIPSのもっている、ラムダを除いたほとんど全ての特徴を備えている。人によってはラムダが本質的だとおっしゃられるかも知れませんが……。しかし、例えば、リスト処理、あるいは再帰呼出しというものは素直に入ってきている。2番目はLIPSよりもアプリケーションに適しているのではないか。それはパターン照合とか、バック・トラッキングが、自然の形で組み込まれている。そういうことによって非決定的な処理が簡単に書けるということです。

もう1つの点はコンベンショナル・ランゲージでなくて、どうしてロジカル・ランゲージかという点ですが、これは検証ですとか、合成に必要なフォーマルな扱いがずっと簡単である。これは1例ですが、ロジック・プログラミングですと変数の概念がシンプルである。それに比べて、コンベンショナル・ランゲージでは、変数は時間によって変わる時系列になるのです。数学的な扱いが全然違ってくる。

2番目は、これはプログラムの開発をどういうふうにかえるかにもよるかも知れませんが、1つはコントロールのプリミティブが限られているので、プログラムの表現に制約が出てきて、どうしてもウェル・ストラクチャードなプログラムになるという傾向があります。

その結果、信頼性が増加するし、モデファイなども容易になるということがいえます。

最後に今までいってきたメリットに対してドローバックがあるのですが、それは効率と機能で、ある重要な機能がないとかいうのがありますが、それをどういうふうに克服したらいいか、そういう道があるのかということです。

効率については例えば、最適化をやる。最適化をやるということはシステムの中でやるということが大切で、あくまでユーザが扱える言語としては機能的な良い性質を保ったままで、最適化をやっていくということです。

例えば、知的バック・トラッキングであるとか、コンパイルというようなことが考えられます。これは一番最初に述べましたが、専用化したハードウェアによって効率が向上する。新しいアルゴリズムも開発しなくてはいけないかもしれません。

最後に、ソフトウェアの開発も含めた総合的なコストを考えますと、開発が容易になってくるということで、全体のコストはむしろ論理的プログラミング言語をベースにしたものの方がよくなるというように考えています。そこら辺が一番最初にいいました開発する時の、どれくらい人がかかってどうなるかというところに関係してくるのですが、開発を加速するために重要なものになっていると考えます。

現状の PROLOG ではいろいろな機能が不足しています。それは適切なやり方があれば、それで拡張していかなければいけないということで、ラムダであるとか、正当性、抽象データ型、モダリティ、メタ・コントロール、ここら辺の話は私の昨日の話で触れていますので、詳しい話は省略します。

淵 どうも有難うございました。いろいろな問題提起、あるいは説明があったわけですが、第2ラウンドに入る前に私も何かいう義務があると思います。

一番最初にファイゲンバウムさんがプロジェクト全体のことについてコメントをされ、これは確かに貴重な意見として受け取りたいと思います。

そのご指摘の中に、日本には知識工学についてはほとんど経験がない、今から大いにやれという激励だと思えますが、しかし、現在までのところ、第1級の業績として参照される仕事は極めて少ないかもしれませんが、量は少ないにしても、ある程度の蓄積とか、歴史が日本にもあります。

そういうものは少ないかもしれないが、そういうものをベースにいろいろな人の知恵を集めて計画を立てつつあるということもいっておきたいと思えます。

例えば、皆様の国は大人だとしますと、日本は赤ん坊であるか、あるいはその前であるかということかと思いますが、私の感じでは少年ぐらいにははいつているのではないかと。

少年の心得を私がいうのも変ですが、大人を学ばなければならぬことも確かです。その意見もよく聞いて尊重することも確かです。

しかし、自分たち自身で判断して、自分たち自身が成長していくということが基本だろうと思えます。

第5世代についても私はそうである必要があると思

うわけで、すでに成長した大人の振りをするつもりは毛頭ありませんが、そちらの方にちゃんと成長していくと、関係した人々は思っているのではないかと思います。

このことやプロジェクトの構成については、これまでのご説明の中で3段階に分かれているということは、ファイゲンバウムさんのいわれたステージ1、2、3とかなり似ていると申し上げてもいいのではないかと思います。

最初のステージはかなり基礎的な研究であって、同時にそこにおいて日本の中で知識工学、ソフトウェア・エンジニアリングのヒューマン・リソースをもう一度育て直すというか、もっと数を増やしていく、若い世代を増やしていくということも含まれていると思います。

この点は日本で過去に行なわれてきた、大規模のプロジェクトとは違ってくるのではないかと思います。それは国の政策もそうなることを期待しますし、取り上げるテーマの性質からいって、そうなっていないと、本当に目ざす目標が達成できない。

ただ、漠然と、基礎研究を試みようとするのではなくて、ある程度鮮明なゴール、方向性をもって進めていくということが大事で、これまでの委員会の作業は枠組とそれに伴ういろいろな検討を進めてきたと私は了解しています。

これはビーベルさんがセルフ・モディフィケーションが必要であるとおっしゃったことにも関係がありますし、カーンさんがゴールオリエントであることは大事であるけれども、また各分野での意見を聞くべきだとおっしゃいましたが、それはこれからも勿論必要ですが、これまでも各研究分野での意見をできる限り吸い上げて、それをできる限り統一的な形でまとめようと努力してきたつもりです。まだ不十分かもしれません。

私が初日に下手な講演をしたのですが、そこでの気持はそういう意見は一見バラバラに見える、研究者というのはオリジナリティというか、個性を大事にしますので、俺は反対だとか喧嘩するのですが、その中をさらに探ってみれば、非常に共通の要素をもっている、共通のゴールをもっている。それが第5世代のプロジェクトの基礎であるということを一昨日申し上げたかったのです。

いろいろなポイントから意見が出たのですが、パネリスト同志のディスカッションもあるかと思うのですが、昨日来、もっとフロアの皆様のご意見なり、ご指

摘を受けるべきだという声もありましたので、お願いして、またそれを受けてパネリストでそれぞれ議論していただこうと思います。

お話になる前に所属とお名前を各自お願いします。早速ですが、東京大学の後藤先生をお願いします。

後藤 2つ質問があります。1つはテクニカルなのですが、ラムダとおっしゃったのは何のことか。もう1つはPROLOGを使われるとおっしゃいますが、PROLOGの使用経験は具体的に何ラインぐらいのどのぐらいのプログラムを書いておりますか。

プログラムというのはかなり使い込んでみないと、その特徴はわからないと、今のPROLOGには十分なサポートはないということです。

古川 ラムダだけではわからないと思いますが、ファンクショナル・アーギュメントの処理です。ハイヤ・オーダになってくると、今のPROLOGには十分なサポートはないということです。

後藤 ラムダ・カリキュラスではなくて、要するに名前のない関数ですから、名前を付けなければいけないことだと私は思うのです。

本当のラムダ・カリキュラスをチャーチのラムダ・カリキュラスを使うということなのか、ただ名前がない関数は使えないよということなのか、名前のない関数であれば名前を付けばいい。

古川 アーギュメント自身が、ハイヤ・オーダにできないのです。

後藤 だけど、ファンクショナル・アーギュメントだったら名前を付けばいいのではないのですか。

古川 名前を付けられるというメカニズムは簡単に実現できるのですが、それだけでは不十分なところが出てくるのです。

後藤 LISPの使い方ではラムダはいらないです。

古川 そうなのですが、PROLOGですと、それだけではうまくいかなくて、変数のスコープといいますか、全体がグローバルといいますか、そういうふうになっていて、ラムダというのはラムダバインディングの範囲だけで有効だというメカニズムをうまく実現することができない。

後藤 これは余りテクニカルになりますから、今日はいい。もう1つ、PROLOGにどのぐらいの経験をおもちなのかということです。

古川 非常に手厳しい質問であると思います。全くないわけではありません。そういうことを目的としていろいろな人を動員して使用経験を積んでいるとこ

ろです。

例えば、私の例ですと、リスティングで10枚ぐらいですから、それぐらいのプログラム、1つと、それはルービック・キューブをとくプログラムですが、それとあとはテキスト処理の簡単なプログラムとか、簡単にいうと、LISPのプログラムをどうやって翻訳するか、対応づけができるかというぐらいの見通しはもっているということです。

カイファン 台湾の者ですが、PROLOGをデータフロー・マシンに使う適応性について、適正さについておっしゃっていただきたいと思います。PROLOGについては賛否両論であると思いますが、PROLOGをデータフローのようなアーキテクチャに使うことについての良さ、適正さについてお話しいただけますか。

淵 充分ではありませんが、まず私がお答えして、誰かさらに補足していただきたいと思います。

データフローマシンはそれなりの意義もっている。PROLOGのようなプログラム言語もそれなりの意義もっていると思います。それをなぜ結びつけるのかということでしょうか、ご質問は。

カイファン ちょっと言葉を替えてご質問させていただきます。今朝のいくつかの報告を伺っておりますと、推論マシンやデータベース、あるいは知識ベースマシンでデータフローが随分強調されておりましたし、PROLOGが強調されておりましたので、私は自動的にその2つをリンクをして関係があるのではないかと推察したわけです。

何かその2つの間に相関関係があるのでしょうか。

淵 そういうふうに推論されたのは自然だと思うのですが、これについてはもっと議論をつめなければなりません、大いに関係がある。その2つを結びつけることに可能性があるという感じと、無理に結びつけるのではなくて、自然な結びつきがありそうだということで、そういう仮説のもとに検討を進めていくということです。

検討を進めていくと、われわれの観測に誤りがあって、間違っていたということもあるかもしれませんが、これまでの議論ではその方向の可能性を大いに考えてみようということに進んでいます。

これは確実な答である。それら2つがそれぞれベストであって、全く自然に結びつくという結論を出しているわけではない。

しかし、議論の過程で有望であるか、あるいはそれ

は間違っているかもしれないという感じは出てくると
思います。

現在のところはもっと検討を進めてみるべきだとい
うような意見が強いので、そういうことをまず第
一に進めているということです。内田さんに補足して
いただきます。

内田 私の今朝の話聞いていただいて、多分そ
のように推察していただいたのだと思いますが、私と
してもそのように推察していただくことを期待して話
したわけです。

それは過去のコンピュータ・アーキテクチャの研究
の1つのもっとも大きな課題といますのは、マイク
ロ・コンピュータが世に出た時から、これを汎用につ
ないで速いコンピュータをつくれないうかというこ
とがわれわれのアーキテクチャを研究する者としての
大きな興味であったのです。

それと同時に、高級言語、つくられたマシンがプロ
グラムできるということとどう結びつけるかというこ
とが最大の問題であったと考えます。

過去においてそのような背景があって、マイク
ロ・コンピュータをそのまま使うという形では簡単に解
けるような問題ではないというのが過去の研究の1つの
結論であると思います。

それは何を意味したかといいますと、そのような高
級言語と結びつけるためには、われわれアーキテク
チャをやっている者が考えたよりもずっと多くのバ
ードウェアのメカニズムを使わないと、その高級言語
とは結びつかないという結論も同時に示唆していた
と思います。

現在、データフロー・マシンのサポートしていま
すのは純粋な関数型言語であって、必ずしもPROLOG
のようなものとはまだ一致したものではないと思
います。

これについてはわれわれのグループの中でもいろ
んな意見がありますが、私はアーキテクチャをやる
者としてはまだPROLOGは並列処理に不十分である
と考えます。並列PROLOGの意味はデノテーション
・セマンティックスの方に従えば、それ自身パラ
レルな可能性をもつわけですが、実際問題、プロ
グラムを書いてみますと、そうでもないというよ
うなことで、その間には並列にどうやるかという
計算モデルをソフトウェアの研究者の方、理論の
研究者の方がアーキテクチャを研究する者に提
案していただけるのではないかと期待していま
す。

しかしながら、その間のギャップというのは過去の

コンベンショナルなプログラミング言語から並列
処理の部分抜き出すというような問題に比べれば、
まだ手がつけやすいのではないかと考えていま
す。

コンピュータのアーキテクチャの方としてはデー
タフロー・マシンをどういうふうに行うか、完全
にエンシクロナスでやるのか、一部はシンクロナ
スな部分を入れるのか、それについて実際にど
ういうアーキテクチャをとるかはまだ可能性は
随分あると思います。

それ自身、VLSIとどう整合するかということ
を考えると、まだ代替手段は汎用にあると思
います。

しかし、われわれが今スタートする時には現在
の研究の状態を出発点としたいということなので、
慎重にデータフロー・マシンの現在のアーキ
テクチャのレベルからスタートするつもりです。

その間のギャップは新しいアーキテクチャの
研究者がソフトウェア及び理論も詳しくよく知
っているという人がその間を埋めていくのでは
ないかと期待しています。

今のご質問のようにしたいというのがわれわれ
の希望であり、それをわれわれはわれわれの
プロジェクトの中でのデータフロー・マシンとい
うふうに行いたいと思っています。

これは世界でいろいろな方がイメージとして
描いているデータフロー・マシンの範囲を逸脱
していると思いますが、われわれの希望とし
てはそのレベルまで達成したいと思
っています。

淵 内田さんから既存のデータフロー・マシ
ンの提案から逸脱したいという意見が出
ましたが、ロジック・プログラミングを
ベースにアーキテクチャを考
えるという方向はむしろ逸脱することを
狙う、そのために与えるものとして
はロジック・プログラミングは、より
高いゴールである。ロジック・プロ
grammingは是か否かという議論は
まだあるのですが、それが是か否か
というはさらに高い知識情報処理
というゴールと絡み合せて検討する
つもりです。

マコーミック まずコメントをして、質問
させていただきたいのですが、第5
世代コンピュータ・システムにつ
いての日本の提案は大変野心的な
ものでありますし、ディトロ
のような、フランスの百科辞
典のような世界の知識をすべ
て集めて、新しい発明をつ
くるといった、フランスで初
めて百科辞典が作られた時
と同じような試みであるか
のように私には思えるので
す。

フランスでは革命が行なわれた時にはそういうよ

な表現形というのはなかったわけですが、百科辞典をつくる時には印刷所へ行き、また市場へいき、ツールを使ってテクニックを使うとか、いろいろな所へ行って情報を集めたわけです。

知識ベースというのはこれからどんどん増えていくものでありますから、こういう百科辞典的な形ではとてもついていくことはできない時代が来るだろうと思います。

この百科辞典を英語で書いたり、フランス語で書いたり、ドイツ語で書いたりする場合には3ページ、10ページにするべきなのか、あるいはカラー刷りにするのか、どうするのかといったようないろいろな問題があると思います。

実際にする場合にはフィールドでの知識ベースが何であるかということを見て決めなければならないと思うのです。

各プロジェクトにおいてはどのような知識を集めるつもりなのか、それによって知識ベースマシンの性格が変わってくると思います。

どのようなエキスパート・システムを考慮されるのか、このプロジェクトの一環としてどのようなものを考慮されるのかということをお考えなければいけないと思います。

それについて1つ逸話を申し上げたいのですが、ファイゲンバウム先生と私と友人でエキスパート・システムをオーストラリアでつくった人がいます。この人はスタンフォード大学へ行きましてツールをつくった人ですが、アスレティック・ペグメント・システムをやった人で、これをツールといったものです。

そのあとどのようなことがあったかというと、例えば運動した時に発生する病気、運動家の病気がどのようなものであるかということが十分にわからずして、システムを先につくってしまって、あとからスポーツ肩とか、スポーツ肘がどのようなものであるかということを理解をしたというようなことがあったのです。

どうしてこういうことになったかといいますと、あとで知識がどのようなものであるか、どういう知識を集めなければいけないかということをおあとで知ったわけです。

どういうハードウェアが走るかということは重要なのではなくて、正しいルールをまず選ばなければいけない。まず何を把握しなければいけないか、何をつかまえておかなければいけないかということさえわかれば、あとは簡単であろうと私は思うのです。

淵 只今のはご指摘と思うか、質問と思うかですが、

第5世代のプロジェクトで計画していることで、1つのポイントは百科辞典そのものをつくるということにはなっていない。

むしろ、それに対する共通のフレームワークを抜き出して、それをサポートするようなソフトウェアとハードウェアを用意したい。

しかし、何もなくて共通のフレームワークができるかどうかですが、いくつかの適切なケース・スタディを選んでやるという議論になっています。

共通の、というところで、またいろいろご意見があるわけで、マコーミックさんは問題によって違うとおっしゃる。

それは確かにそうだと思うのですが、これは私の個人的な意見ですが、そのところはかなりの共通の部分が大きいであろう。

これについてはファイゲンバウム教授は異論があるかもしれませんが、あるいは賛成して下さるかもしれませんが、例えば、医療のシステムをつくってみると、それは他の全く違った分野に適用できるということで、すべてとはいいませんが、かなり共通のメカニズムがある。

その共通のメカニズムを間違えて捉えれば大変なことになりますが、もし適切に捉えられれば最初のステップは共通の枠組をいろいろな形で整理する。

それにはいろいろなアプリケーションとのつながりを忘れてはいけないのですが、スペシフィックなところ、他とは共通のないものを選んでやるべきではなくて、むしろ共通のところからスタートすべきだと思います。これは私のコメントで、正しい答えかどうかはご判断願います。

フロア外人 調査を2年ぐらいやってきて、この第5世代をどのような方向へもっていくかということで、1歩下がって考えてみたいのです。

知識情報処理システム、コンピュータ・アーキテクチャ、VLSIという方向を選ばれたのですが、その中で大きな部分として皆様方がこれはとらないと決めた部分はどのような部分か、それをお聞きしたいと思います。やめた部分、こういうところは研究の対象としないと決めた部分はどのようなところですか。

またなぜそれをやめたのかも説明していただきたいと思います。

淵 これは全体の話ですので、ここで議論するのが適切であるかどうかわかりませんが、とらない部分をリストアップはしておりません。残りは無限にあります。

すので。

百科辞典はつくらないとか、偏微分方程式を解くようなスーパーコンピュータの開発は別のプロジェクトがありますので、現在の計画書の中には入っていないとか、あるいはそれと関連しますが、新しい素子、ジョセフソン接合素子もそれに含まれます。

これはコンピュータの世代と素子が深く関係があると主張される方もあると思いますが、現在そういうのはこの計画には入っていません。別の計画になっています。

カーン　ここで私自身は言語の問題があるという長尾先生の御意見に賛成です。私が賛成だということとは申し上げればおわかりいただけると思います。例えば、私がそう思いますといえば、長尾さんにはおわかりになっても私にはわからないわけです。

ですから、 x, y, z のラムダが x, y, z, I, you というものと一致するとすればどうおっしゃるでしょうか。

それでは理解できないと思います。あるいは I, you 、私は、あなた、賛成しますではわからないと思います。

私はここで何をいいたいかといいますと、ある特定のマシンなり、1つの知識を表現するために使っていますマシーナリィは、あるものについては簡単に表現できますが、あるものについては難しいわけです。

先生はメカニズムの性質というものを説明されたわけですが、5年後にわれわれが使うメカニズムは今日われわれが使っているメカニズムとは違うということを示唆されたと思います。

もしマシーナリィも変わるのであればいいわけですが、そのマシーナリィを機械に入れてしまいますと難しくなってしまいます。つまり、一階述語論理、あるいはラムダ変換言語、英語、日本語で表現できないようなものについてはますます表現は難しくなると思います。先生、如何でしょうか。

実際に知識を表現するのに十分にシステムの中で表現できるようなものができるまでには数年かかるとお思いにはなりませんか。

長尾　今ご指摘になったことはすべてもったもんだことだと思います。私が例で申し上げたのはあくまでもリニア・ストリングスとしての言語を扱う範囲においていったわけであり、知識としては言語以外にいろいろと沢山あります。

そういったものの場合にどうしたらいいかというのはいちいち考えなければいけないと思います。

ただ、言いたいことは、すべての事柄を、すべての知識をコンピュータに入れることは恐らく無理だから何らかの形で連想的に物事を解釈するという能力をもたせなければいけないということです。

連想的にやれる範囲がどの程度のことかというのはなかなか難しいことです。脳の活動の素晴らしい人は深く推論していくでしょうが、多くの場合は比較的浅い推論しかしません。

その代り知識は沢山もっているということになりますから、コンピュータといましょうか、情報処理機械もそういう方向でうまくやれるようなメカニズムをもつように考えるのがいいのではないかと考えています。

これはカーン先生のご指摘にうまく合った答えにはなっていないかと思いますが、私の考え方はそうです。

淵　それではフロアからもまだ手が挙がっていますが、ちょっと待っていただいて、パネルの皆様にもう一度、順番はいつでもよろしいのですが、フロアからの質問、あるいはパネリストの意見等に関して述べてもらいます。

ファイゲンバウム　長尾先生のコメントからお話したいのですが、私は100%賛成します。長尾先生がおっしゃったコメントと申しますのは実際にビーベルさんが前にお書きになったグラフの再結合部分だったと思います。

誤解を招く可能性があると思います。グラフ自体は推論の力は知識とある場合には直線的なトレード・オフがある、2つの部分があると思います。

1つの部分というのは人間あるいは機械はどんなものでも全部のリーズニング・チェインをリ・コンピュータすることはできないのではないかと思います。

その理由というのは余りにも高価である、時間がかかるということです。それよりも知識を浅いリーズニング・チェインで沢山の知識をおいておく方がいいと思います。

第2番目のことは飽和(サチュレーション)の効果ということです。推論のメカニズムというのは短い距離しか進めません。パワーが限られているわけです。プロセスのパワーが飽和してしまうわけですから、いろいろな知識をストアしておかなければならないと思います。

もう1つのコメントを私が申し上げたい。フロアの中の3人の方からのものをまとめて、後藤先生のもの、マコーミックさん、私のもの、3つのことを違った観点から述べたものだと思いますが、1つの提案

として言いかえたいと思います。

第5世代コンピュータ研究開発計画の第1番目のステージとして、もう一度まとめて申し上げますが、私どもは人間の知識というものは経験を一般化したところから生じるものとし、いろいろなことをやって、それから学んで、そこから一般的な概念に辿りつくのです。

基本的なマシン・ランゲージ、例えば、それを第5世代に対して、どういふものを選ぶかという選択について、いろいろな議論があると思います。第5世代の中でスーパーLISPにするか、PROLOGにするかということで議論があるようですが、いろいろな経験のベースというのがあると思います。

そこからLISPはエキスパート・システムをつくるのに非常に強力な言語であるということにはわかっています。しかし、私が思うにはPROLOGが良い言語である、複雑なエキスパート・システムをつくるのによい言語であるということを経験するベースはまだないと思います。ほとんどないと思います。

私は実際に、このプロシーディングの中のものをざっと見たのですが、ルービック・キューブのプログラミングも、プロダクション・システムのPROLOGを使ったものなど見たのですが、一般化するにはまだ足りないと思います。

エキスパート・システムをつくるにはエジンバラでは小さなメカ・インプリメーションのものしか出ていません。ですから、最初のプロジェクトの2年間で行なわなければいけないのは20位のエキスパート・システムをつくって、その経験をもとにして、そこからどの言語を使った方がいいのか、マシン・ランゲージとしてどれが最適であるかというのをその経験を通して、そこから見出ししていくべきではないでしょうか。

今どれがいいということをいうのは時期尚早であると思います。私自身の考え方としてはプロジェクト全体をここに傾けてしまう、かけてしまうというのはちょっと早いのではないかと思います。

私自身としてもう少し、肘掛け椅子に腰かけて考えてみる、例えば、マコーミックさんがおっしゃった20ぐらいのシステムをもう少しじっくり考えてみる必要があるのではないかと思います。

もう1つ申し上げたい意見はこれは先程も出たことですが、第5世代の研究開発が出てくるプロダクトというのが予期されないものであるという点です。

マコーミックさんがおっしゃったのですが、ディンバーク・ドクターがその知識を表現する過程で大きく

向上したということを経験した場合は取りあげておっしゃいました。このプロセスのために私たちはあるフレーズがあると思います。

これは知識のエンジニアリングです。知識工学ではなくて、知識のエンジニアリングです。マシンで使うために知識を入れるというのが知識工学ですが、その知識をもっと鋭いもの、もっと調整されたものに変えていくということ、チューニングするということが知識のエンジニアリングです。

より長い視野で見えていくと、知識情報処理というのは人間の知識をだんだんに洗練し、高度化し、より明確なものにしていく、その知識をパーズングし、オーガナイズしていくということではないでしょうか。

それによって非常に大きな影響、インパクトが与えられると思います。

最後にもう1点述べたいことはマンパワーと革新(イノベーション)ということです。昨日私の話の中で私が強調したのはイノベーションという問題です。というのは非常に重要で、とくに第5世代の問題では必要であると思ったからです。

このプロジェクトは非常に難しいので、誰もすべての人の想像力をできるところ限界まで引き伸ばしていく必要があります。引き伸ばしてギリギリのところまでもっていくためには現在、人間のいろいろな事柄を扱っていく組織(研究所)の能力をギリギリまで引き伸ばしていく必要があります。

現在の研究所、産業界の研究所、学問の大学関係の研究所といったその能力に縛られていると限界があるのではないのでしょうか。

将来においてはそのように学問と業界と、あるいは大学と業界とこれまでのように伝統的に2つをはっきりと区別してしまうということではいけないと思います。

若い人々が知識情報処理の方に入っていくようにするためにはコンピューティングに関する他のものに与する言及をする前にそうした動機づけを与える必要があります。

早い時点で正しいアイデアを大学に入っている時点で得れば、そのことによって有用なマンパワーを早く第5世代に関係のある研究所にマンパワーとして入れることができるのではないかと思います。

例えば、これから2~3年後にどういふ状況が起こるのかということを考えてみたらいいと思います。通産省が大きな研究所にお金を出して、大学の方は1人の教授、3人の生徒という形で小さいものになってき

て、業界の方では大きな研究所になっています。

そういうことになると、非常に大きな不均合いが生じますし、また大きな防害になると思いますので、大学が人間を訓練しようとしている努力となかなか噛み合わないと思います。

この問題を解決するためには例えば、合同の研究所、大学の研究所、実験室と業界の研究所とを合同のものとして、学部の子生も実際の業界の研究所と共同するような形で目標達成していったらいいのではないかと思うのです。

大学の研究者、教授というのはよい研究者なのですが、それはうまく教えることができるというよりは知識の一番最先端にいるということが重要なのです。

最先端にいるということはこれまでの伝統的な学会では最先端から遠ざかるという傾向にありますので、そうではない産学共同のシステムをつくっていく必要があるのではないかと思います。

ビーベル 私、ちょっと付言させて下さい。私の図についてご意見がありました。むしろ先生は私の図をサポートして下さいと思います。

これは直線リニアラインであるべきではないかと思えます。簡単に書きましたが、一方、バランスをとらなければいけないというつもりで申したわけで、飽和（サチュレーション）に近いものになっていくであろうと考えていますので、それほど先生と意見が違っているというものではないと思えます。

一般的なコメントとして申し上げたいのですが、第5世代プロジェクトのツールの中にいくつか気になる点があります。すべての科学、とくにコンピュータ・サイエンスにおいてはファミリーティというのが大変重要です。新しいツールがあってもコンピュータ・サイエンティストが余りよく知らないというものがあると思えます。

知らないとどうも使う気になれないということがあると思えますので、新しいツールによる解というのは別のツールによってつくる解とそれほどかけ離れたものにはならないと思えます。

従って、新しい経験を新しいツールで重ねていくことによって知識工学の本質的な性格というものを理解し、新しいツールと古いツールの間の橋渡しをしていくべきではないかと思えます。

例えば、PROLOG やその他のものについてのアイデアというのは賛成ですが、最後に一言申し上げたいのは私もソフトウェア・サイドからハードウェア・サイドにものを申せば、ひとつの希望があります。

ユニフィケーションをすることは可能であろうと思えます。メカテクニカルなコメントとしてサーキュラリティのことをいわれましたけれども、ユニフィケーション・アルゴリズムを考えた場合に、モンターギューのアルゴリズムを考えますと、このサーキュラリティをひとつの手続きで行なっているのです。

これがハードウェアにもうまくもち込まれておりますので、そういうものが実現することを是非期待したいと思えます。

これはたった1つの例でした。他にも沢山例があると思えます。例えば、コネクション・カルキュラスをチップにのせることもできればよいと思えます。

コネクション・ブーフをナチュラル・ブーフにつなぐこともよいと思えます。ひとつそういう例もあったと思えます。内部表現の形、あるいは知識サイドで先程のデータベース・ストラクチャのスライドの中で申し上げましたように、こういうようなものがハードウェアにのってくるのではないかと私は思えます。

こういうようなポイントはかなり沢山ありますので、そういうポイントを是非認識してプロジェクトを進めていただきたいと思えます。

カーン 小さな点ですが、ユニフィケーションの話をこれ以上続けるつもりはありません。またLISPとPROLOGのどちらを使うかということについても言及することは避けておきたいと思えますが、ひとここで警告を発しておきます。これは私自身の経験、あるいはプログラミング言語のデザインに参加した経験から申し上げたいと思えます。

プログラミング言語を設定する場合にはニーズに対立するものがあるのです。論理的にはクリーンなものをつくりたいという要求と、エンジニアリングの要求とが必ずしも一致しません。

例えば、PROLOGのような言語をとってみますと、これを拡張したいという場合、これはプログラミング言語を拡張するという事は決して些細なことではありません。

だからこそ最初に言語を選ばれたのだらうと思えます。そして、PROLOGの場合には如何に評そうともエンジニアリング上の要求は現在は十分に満足されていません。

従って、PROLOGを拡張されたいのであれば、この問題は過去の多くの研究者の頭を悩ませた問題ではありますが、PROLOGをよいものにして、プログラミング言語のセマンティックスがディノテーションに表現されている、非常に簡単な数学的な形で表

わされているものであるので、エンジニアリングの要求を満足させようとしますと、コンピューテーションをオーガナイズして、データストラクチャのファンクティなどももう少しよくし、毎回コードしなくていいようなものにしようとしていきますと、パンドラの箱をまた開けてしまうことになると思います。

今、皆様のプロジェクトはパンドラの箱の上にかかっていますので、その点が危険をはらんでいると思いますので、その点を強調させていただきたかったのです。

淵 コメント有難うございました。どうも議論聞いていますと、何かを決めてしまった、これが正しい結論だとわれわれが主張しているように思われて、反論を受けているような感じもしますが、そうではなくて、いくつかの新しい魅力的なものを伸ばしていこうというのが基本精神です。

言い方が悪いかもしれませんが、新しいものを見ないという精神はとるべきではない。パンドラの箱には何が入っているのでしょうか、開けてみないとわからない、あるいはすでにお開けになったということですが、しかし、人類は破滅していないと思います。新しいいろいろな要素を積極的に取り入れ、その時にできりゃばバラバラではなくて、なるべくクリーンな形で取り入れたいというのが一番の基本であるわけで、これが正しい、最終の結論だということを目指しているつもりではないので、そのへんをよろしくお願ひしたいと思うのです。

そういう意味では日本は若いのだと思います。先程使った比喩でいいますと、大人は余りにも経験が足りすぎるといえるかと思ひます。

フロアの方から村田さんお願ひします。

村田 私は2つの質問があります。1つはビーベル先生にテクニカルな質問です。もう1つはファイゲンバウム先生に質問があります。

まず第1のものはテクニカルな質問で、ビーベル先生はまず初めに大きな仮説から始めたと思ひます。昨日の午後のご講演の中で大きな仮説からお話を始めたと思ひます。

何か人間のアタランスをあるロジカルなフォーミュラにするようなシステムがあると。これはかなり仮説的なものと思ひます。これから何年かの間にかういものが出てくるのでしょうか。

現実的な視点から見ますとある意味ではこれは可能であると思ひますけど、他の意味から、あるいは視点から考えれば近い将来では無理だと思ひます。

ACLLのモンターギョ先生の文法が新しく出てきて、言語の文法についてのもが出てきて、論理的なフォーミュラに変換されるようになってきたと思ひますが、これにも自然言語全体を変換するにはまだごく一部分にすぎないと思ひます。

ですから、ビーベル先生のコメントをこの点に関して、先生がどのようにしてこのような仮説を立てられたのかということをお聞きたいと思ひます。

もう1つ、ファイゲンバウム先生には社会的な関係のご質問ですが、ファイゲンバウム先生のコメントは価値があったと思ひます。

これは先生の貴重なご経験に基づくものだと思ひます。先生はその中で丁寧な言葉でおっしゃっていましたが、あるギャップが現在のインテリジェンスのレベルが日本とアメリカでは現状が違うとおっしゃったように思ひます。

格差、ギャップというのは私には大きく響いてきました。というのは他のコンピュータ・サイエンスの分野を見てみますと、例えば、コンピュータ化されたバンキング・システムなどを見ますと、日本もかなり進んでいるのではないかと思うのです。

業界の面を見ますと、日本はうまくいっているのではないかと思ひます。産業界では成功を収めているように思ひます。ですが、人工知能という分野においてはギャップが日本とアメリカではあると思ひます。

そこでファイゲンバウム先生にできればアドバイスをいただきたい。この社会的な弱点というものがわれわれの側にあるのではないかと思うので、それを指摘していただければと思ひます。

業界と学問の方の産学間のコネクション、関係が日本とアメリカでは違う、日本の場合にはかなり大きな区別、断絶があるということをおっしゃいました。もう少しこの点についてご意見をいただければと思ひます。

というのは、先生はこの問題を答える資格がある。東大で客員教授として少しいらっしゃる経験があるし、また学界と産業界の両方に関与したご経験があるので、とくに先生に伺いたいと思うのです。

淵 ビーベルさんに対するのと、ファイゲンバウムさんに対して質問されたのですね。短かくお答えいただけますか。最初にビーベルさんの方から。

ビーベル ごく手短かくお返事します。私、仮説を立てたというわけではないのです。むしろ高いゴールを掲げたといった方が正しいと思ひます。

こういう高い目標が達成されれば、私の具体的な解というものがこの中にあてはまるということです。高い目標自身がもうひとつ落ちつかないという印象をおもちになったのかもしれませんが、仮説というつもりで申し上げたわけではありません。

ファイゲンバウム 私が今直面している大きな問題は一体どういうご質問だったかと理解することです。

村田 私の質問は先生にアドバイスをさせていただきたいのです。こういう弱さ、日本の人工知能の面における日米の研究格差を如何に埋めていったらいいかについてアドバイスしていただきたいわけです。

ファイゲンバウム 大きなものは淵先生がいわれましたように時間の問題だと思います。アメリカ合衆国の専門家はこういう研究を全般的にいつて25年間はやってきたのです。

そして知識工学については16年も研究してきているわけですが、日本の場合には数年といった研究期間です。こういう概念が広がるには時間がかかるのではないのでしょうか。

この問題に焦点を合わせるためには第5世代プロジェクトの第1ステージにおいて十分な時間とインセンティブを与えて、追いつくための大きな歩みをする必要があると思いますから、時間が1つの問題だと思います。時間が1つの要素だろうと思います。

もう1つの格差の原因となっている問題点というのは私の意見では日本でヨーロッパに見られると同じような理論的に走りすぎやすい傾向があるのではないのでしょうか。グリーンにものをやりすぎるきらいがあるのではないかと思います。エレガンスの方が実際性、実用性よりも先に立ってしまうきらいがあるのではないかと思います。

3番目の点として挙げられることはこれはすべてのコンピュータ・サイエンス、あるいはテクノロジー・ラインに共通していえることであり、人工知能の分野だけにあてはまるものではありませんが、ハードウェアが素晴らしいものであって、ソフトウェアはそれに従属するものであると考えてしまうきらいがあると思いますので、各企業でハードウェアをやっている人は一流の人である、ソフトウェアは2流の人間であるというふうに思われているきらいがあるのではないのでしょうか。

大きな声ではこういうことがどうして出てきたかということはお聞きしないことにしておりますが、1つの考えはもっています。

私は昨日も申し上げましたように、ハードウェアは今や問題ではない。ソフトウェアが問題なのです。ハードウェア・エンジニアというのは大変素晴らしい、奇蹟でももたらすことのできるようなエンジニアであります。

ですから、今こそソフトウェアに焦点を合せて、ソフトエンジニアに高い名声を与えていきましょう。

次の要素として申し上げられるのは産学の協力です。産学の協力が今ひとつ充分ではないのではないのでしょうか。

シリコン・バレーやスタンフォードは産業界からこういうものがとくに問題になっているかということをお吸い上げ、また産業界が科学の急先鋒の情報を吸収できるようにしております。

なぜかといいますと、産業界がすぐに解をほしいという状況におかれているからです。これは日本では難しいということは承知しています。

といいますのは文部省と通産省が別の省になっているので、はっきりと分業体制になっているので、なかなかそういうようにするのは難しいということはおわかってはいますが、これが理由だと思います。

ピエール フランスのピエールです。このようなプロジェクトを、これに限らず成功させるためには3つの基本的な問題点というのが出てくると思います。

ひとつは技術の現状が充分であるかどうかということです。スピーカーの先生方からはいく分かの解答が寄せられています。2番目の問題点はどれくらい能力のある専門家がそのプロジェクトに必要であるかということ、そして、果してそれだけの専門家が集められるかどうかということです。

3番目の問題点はどれだけの資金がいるかということです。そして、2次的にはその国の政府がどれだけそのプロジェクトに金を出す用意があるかという点です。

とくにあとの2つの点については今日、明日の会議の中でいろいろご指摘があるかと思いますが、そういう点を指摘させていただきかけたわけです。

溝口 東京理科大学の溝口ですが、立場的にはソフトウェアのワーキンググループに属しています。マンパワーと経験についてのコメントをしたいのですが、1つは私がこの6月からグループに参加して一番気が付いたことが、ファイゲンバウム教授がいったように日本のエンジニアは非常にアダプティブだということが感じられたのです。

それはなぜかという、ほとんど知識ベース、メタ

知識、モンテギ、文法、知識獲得といった言葉を理解していなかったのですが、この3カ月間にほとんどその言葉を吸収し、なおかつ新しいコンセプトをつくりつつある。

それはそれなりに分析してみますと、1つは淵さんの第1日目の講演にありましたように、潜在的な流れがあって、データベースにおけるロジックの問題、あるいは推論の問題というのと、人工知能あるいは知識ベースの問題というのが潜在的には同じであったという感じがするのです。

そういう未経験でありながら、そこに流れている思想というのが共通していて、それが知識ベースという形になってきたわけですから、それほど経験、あるいはマンパワーという問題は気にしなくていいのではないかと、実際にプロジェクトの計画段階で感じたわけですね。

その場合に PROLOG のプログラミングの経験がないからという問題もありますが、それに対してひとつのベーシックな LISP なり、エキスパート・システムなりという経験はかなり日本でも蓄積があるのです。

ですから、そういうものの中からひとつはロジックが推論のメカニズムを構成する時に有効であるということをもとめたのが、私の理解するところだと、ロジックだと思ふのです。

巨大なソフトをつくらなくともつくらないとかいう問題より、むしろ潜在性を顕在化したというのが非常に大きいのではないかと。

余りにも経験とマンパワーということに気にしなくともいいのではないかとというのが私のコメントです。

もうひとつはファイゲンバウム教授に聞きたいのですが、スタンフォードでも同じようなマンパワーの問題があるはずだと思うのです。多分、第1世代の知識ベースプログラムというのは完了したと思ふのです。

これからは第2世代の知識ベースシステムをスタンフォードではつくろうとしている。例えばひとつのきざしはモデファイアブル表現言語というグレイシャレイクなどがやり始めているのは完全にロジックをベースにしてデータベースを構成するというアプローチなわけですね。

その時にマンパワー、あるいはそれを教育するという問題がスタンフォード大学でも多分起こるはずなのですが、そういう問題をスタンフォード大学ではどういふふうに行っていくのかというのが私の質問です。

ファイゲンバウム 溝口先生のご質問のところ

がよくわからなかったのですが、スピーチの一番最後のところだけおっしゃっていただけますか。

溝口 私は教育のことを伺いたかったのです。知識工学をどういふふうに行っていくのかということですね。知識ベースに関する知識工学というのは大変大きなものですので、システム自体をサポートしていく、博士号をとったあとサポートしていくというのは大変だと思いますので、大きなシステムをどういふふうに行っていくのかということですね。

こういうようなツールを保持していくというのは大変だろうと思います。とくに新しい知識ベース・システムではそういうものを保持していくのは大変だろうと思います。

ファイゲンバウム 恐らくお尋ねになっているのはこういうことですか。大変大きなソフトウェア・ツールが開発されてきている、それがそれぞれ大学院生が博士号をとる間に研究する。そうすると、博士号をとって卒業したらどうなるのかというご質問ですか。

そうすると困るわけですね。要するに困るわけですね。トラブルになるわけですね。トラブルがくるということではわかっているわけですから、クリアなスタイルでインプリメンテーションをやり、またドキュメンテーションもしっかりやってもらいよう必ずリサーチ・サイエンティストが監督するわけですので、次世代の学生が入ってきた時にディテールもわかるようにしておくという方法を考えなければならぬのです。

このシステムを維持していくためには誰かおれわれの方で人を雇って専用に行ってもらわなければいけないと思います。

小さな会社が沢山つくられて、ソフトウェアの高売をやっているというのはソフトウェアのメンテナンスをある程度の期間、産業界で使われている間、メンテナンスをやるための財政的な資金面でのインセンティブがあるから行なわれているのです。

博士論文のために開発したものをそのまま産業界で使えるということまではいっていないわけですね。ドキュメンテーションも少しよくしないとイケないし、コードも効率の良いものに行なければなりませんし、システムももう少しメンテナンスのしやすいものに行なければなりませんし、ソフトウェアの評価ということ、通常コンピュータ会社ではわかっているようなことまでしないと博士論文で研究されたものはそのまま産業界にもち込むことはできません。

そのためにはある程度のインセンティブが必要であります。こういうコードはそのまま小規模の企業に渡

されて、それが商業活動に使われているというふうになるのではないかと思います。そういうお答えでよろしいでしょうか。

溝口 もう1点すでに開発されているツールと、新たに生まれてくるコンセプトの間に衝突がありますが、この衝突をどう解決されますか。既存のツールと新しいコンセプトとの間の衝突を知識システムでどのように解決されますか。

ファイゲンバウム それは2つの方法で解決することができると思います。1つの方法、現実的な、実用的な方法はもっとも的確なツールだと思われるものを選んで、壊れるまでそれを使うということです。

ツールが動かなくなるまでは使うというのが、その次に何を開発しなければならないかということをよく理解する一番良い方法だと思います。どこが問題領域なのかということを理解するもっとも良い方法だと思います。

この戦略は装置の診断プロジェクトなどでも使われています。例えば、EMYCINのツールを最初に使い、最初のインプリメンテーションをやったのです。

プロジェクトでEMYCIN ツールを使っても面白くないという段階まで使っていて、それから初めてMRSという新しいシステムをつくったわけです。それがさらに進んだものであったわけです。

私は話の中で申しましたように、われわれは捨ててしまうコードということについては全然心配しないわけです。コードはもういらなくなれば、捨ててしまってもまた別のアプローチでインプリメンテーションをやって、全く問題がないという考え方なわけです。

もう1つ、文献はよく調べ、新しいシステムについての記述がないかどうかを調べる必要があると思います。

そして、アプリケーションの必要性がないかどうかということを見極める必要があると思います。

例えば、信号理解システムが1973年に登場した時に、私はこのシステムをDENDRALプログラムのソフトウェアでインプリメントしてしまいまして、このDENDRALは化学で使われているソフトです。

これは面白かったのですが、余り役に立ちませんでした。そこで新しい方法に切り替えました。これはその年にカーネギーメロンの科学者が音声理解プロジェクトのためにつくった黒板方式を採用してみましたところ、非常にうまく合いました。成功しました。

われわれは両方の方法を使ってみたのです。従ってソフトウェア・ツールは壊れるまで使ってみて、何

が欠けているかを見極める、そして、文献の中の良いアイデアは吸い上げるようにすべきだと思います。

溝口 私、MRSは述語論理的なものだと思います。これはコメントです。お答えは結構だと思います。

有難うございました。

淵 それでは、時間がきましたので終りたいと思いますが、本日は時間が超過しましたが、非常にフランクな、率直なご意見をいただきまして、どうも有難うございました。

Ⅳ. 10月22日(木) 9:30~17:30

1. 招待講演

- (1) 第5世代コンピュータ・アーキテクチャ分析

ニューキャッスル・アポнтаイン大学

フィリップ・C・トレリーバン

- (2) アルゴリズム, アーキテクチャ, テクノロジー

マサチューセッツ工科大学

ジョナサン・アレン

2. パネル・ディスカッションⅡ

第5世代コンピュータ・アーキテクチャ

座長 慶応義塾大学

相磯秀夫

パネリスト イリノイ大学

ブルース・H・マコーミック

ニューキャッスル・アポнтаイン大学

フィリップ・C・トレリーバン

マサチューセッツ工科大学

ジョナサン・アレン

東京大学

後藤英一

㈱日立製作所

浦城恒雄

電子技術総合研究所

内田俊一

3. パネル・ディスカッションⅢ

第5世代コンピュータ・システムの波及効果

座長 東京大学

元岡達

パネリスト 米国標準局(NBS)

フィリップ・J・リガナッチ

INRIA

J-L リヨン

英国インターバンク・リサーチ

チャールス・N・リード

西独数学情報協会(GMD)

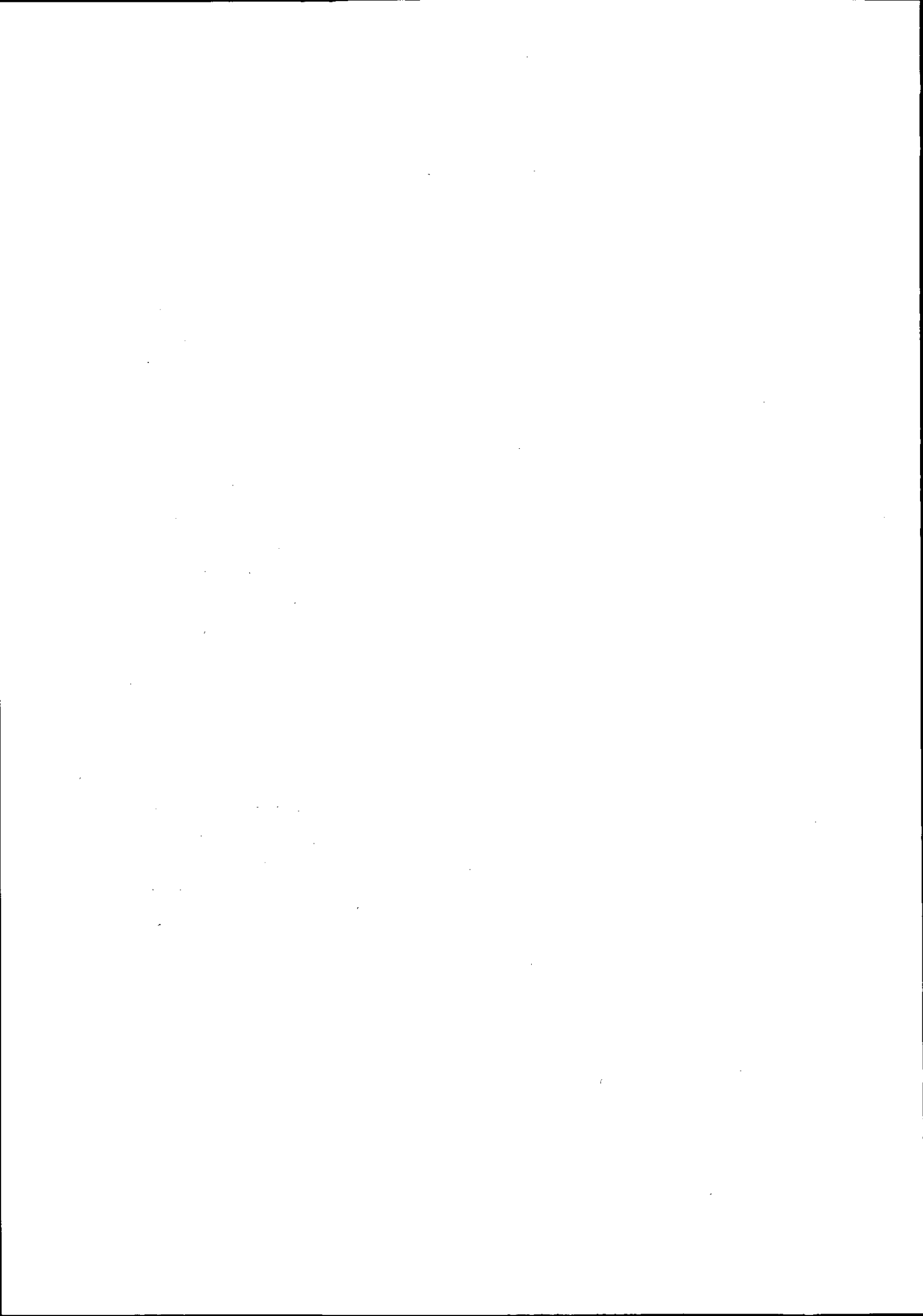
N・ツァイベルスキー

通商産業省

岡松壮三郎

日本電信電話公社

戸田巖





IV.1 招待講演

(1) 第5世代コンピュータ・ アーキテクチャ分析

ニューキャッスル・アポンタイン大学(英国)
フィリップ・C・トレリーバン

講演要旨

最近のアーキテクチャの議論では、ことばの混乱が目立っている。これは非ノイマン型コンピュータというものに対する研究者間の見解の喰違いを反映するものである。ここではコンピュータ・アーキテクチャの基本原則に関するいくつかの考え方を整理し、それらの利点と欠点を明らかにすることを通して、第5世代コンピュータに対する可能性を論ずることにした。

ノイマン型コンピュータは逐次処理を原則とするものであったが、現代のコンピュータ処理のあり方は並列的な分散型のものに変ってきている。これはネットワークが、全国中に地理的に分散化されている場合にも、あるいは多数の極小マイクロプロセッサがVLSIチップ上に集結されている場合にも共通にいえることである。このような並列分散型のコンピュータ・システムに適合しうる可能性をもったコンピュータ・アーキテクチャは何かといえば、それらはコントロールフロー型のもの、データフロー型のもの、そしてリダクション型のものに分類できる。

コントロールフロー型コンピュータでは、命令の実行は明示的なコントロールの流れに従って進められていく。それに対して、データフロー型の場合には、必要なオペランドが出揃ったということでオペレーションが駆動され(データ駆動型)、リダクション型の場合には、その結果に対する要求が出ることによって駆動されるようになっている(要求駆動型)。

コンピュータ・アーキテクチャを議論するときには、次のふたつの側面をよく区別してかかる必要がある。それらは、プログラム機構、すなわち機械語プログラ

ムの表現方式と実行方式に関する側面と、マシン機構、すなわちコンピュータのもつ資源の構成方式と割当方式に関する側面である。コントロールフロー/データフロー/リダクションという区別は、プログラム機構の側面におけるものである。また、最近の研究結果では、マシン機構のあり方も集中型/パケット通信型/表現式操作型に分類することができる。

集中型は現在のコンピュータにみられるもので、プロセッサ資源、交信資源、メモリ資源が、各唯一のものとして存在し、一時点ではひとつの命令の実行に専有されるようになっているものである。パケット通信型は多数の資源がリング状につながった中を、命令が列をなして流れながら実行されていくものである。表現式操作型は多数のプロセッサ資源がツリー状に接続されたもので、表現式の入力構造にしたがってその実行が進められるものである。

プログラム機構に関する決定とマシン機構に関する決定とは、互に独立に行なうことができる。データフロー型のプログラム表現をサポートするのは、どのマシン機構によっても可能であり、これまでに実現されたものもそれぞれの事情に応じて、アドホックな決定がなされてきた。

3種類のプログラム機構の利害得失をみるために、計算機構の基本的な構造、すなわちデータ機構とコントロール機構について、さらに分析を進めてみよう。データ機構は個々の引数が一群の命令によって、どのような方式で利用されるかを定めるものである。リテラル方式は、引数がコンパイル時に明らかになっている場合にとられる方式で、それを必要とする命令のところにそれぞれコピーが埋込まれるものである。つぎ

に、値方式は引数が実行時に生成されてくる場合に用いられるもので、生成された値は複製によって共用され、必要とする命令ごとにコピーが書込まれるようにするものである。最後に、参照方式は必要とする命令に引数への参照番地を書込むことによって共用を行なうものである。

つぎに、コントロール構造は、ひとつの命令によって他の命令がどのように起動され、どのようなコントロール・パターンが生ずるかを定めるものである。逐次型はひとつの命令のあとにはひとつの命令だけが続くものである。並列型は引数が使用可能になったことが制御信号によって知らされ、必要な引数が全部揃った命令から実行に入るようになっているものである。最後に再帰型は、引数の必要性が制御信号によって告

知され、要求を出した命令が引数を必要とするようになったところで、それを生成する命令が実行される。その実行が終了とコントロールは要求元の命令に戻される。

以上のデータ機構とコントロール機構によって3種類のプログラム構造を特徴づけると、図1のようになる。これによっていえることは、コントロールフロー型においてサポートされるデータ共用の方式は、'現地更新'であり、単純な表現式に対してはオーバーヘッドが大きく、大きなデータ構造では有利になるということである。これに対して、データフロー型は'個別コピー'の行き方であり、簡単な表現式には極めて有効であるが、構造を扱う場合には不利になる。また、データの流れとコントロールの流れが一致しているこ

Data Mechanisms

		by value (& literal)	by reference (& literal)
Control Mechanisms	sequential		von Neumann control flow
	parallel	data flow	parallel control flow
	recursive	string reduction	graph reduction

図1: Data Mechanisms and Control Mechanisms.

とから、条件に応じて選択的な実行を行なう場合には問題を生ずることになる。

リダクション型には、ストリング・リダクションとグラフ・リダクションの2種類がある。両者の違いは、同じ表現式に対して複数の参照がある場合、前者は参照ごとに個別のコピーを作成していくのに対し、後者はポインタを利用してひとつのコピーで済ませてしまうものである。前者は単純な表現式の取扱いに有利で

あり、後者は内部構造の複雑な表現式の扱いに利点をもつ。いずれも明示的な出力引数をもたないために制約を免れない。一般にリダクション型では、プログラム、命令、引数のすべてが表現式であり、変数というようなデータ記憶の概念が存在しない。

各方式には互に対照的な利点と欠点が存在しており、応用により向き、不向きを生じている。汎用性のためには合成化が必要である。

Program Organization

Machine Organization	control flow	data flow	reduction
centralised	von Neumann computers		GMD reduction machine [13]
packet communication	LAU [19] GCF [8]	Irvine [2] MIT [6]	Utah AMPS [12]
expression manipulation	Newcastle RCF [22]	Utah DDMI [5]	Cellular machine [14]

図2: Examples of Computer Architectures.

質疑応答

雨宮 NTTの雨宮です。非常に面白い講演でした。1つ質問があります。ノイマン・コンピュータの場合はシーケンシャルに実行します。その時はインプリシットにコントロールフロー、すなわち時間軸上に処理手順を並べていってやることで、言ってみれば、コミュニケーションの立場からすると、コミュニケーションのための情報を、最初から時間軸上に並べておくことによって、そのコミュニケーション量をなくしているという意味で、非常にすぐれた方式だと思います。しかし、処理を分散化させるとか、並列に処理させるということになると、もともと問題を、時間軸上にリニアに並べているので、非常に都合が悪い。それで横に、空間軸上に並べたいということが起こってきて、並列処理のアーキテクチャというものが出てくると思いますが、そうすると必然的に、空間的に並べますからお互いの処理というものを、同期をとるために、いわゆるコミュニケーションをやらなければなりません。そのコミュニケーションの量が、逆に、並列処理をやる時に問題になってきます。

いかにして並列処理を高め、かつコミュニケーション量を少なくするかという2つのトレードの最適ポイントを求めることが、これからのアーキテクチャの1つの重要な問題ではないかと思うわけです。そういう点から考えた時に、データドリブンか、デマンド・ドリブンかという今のご講演が非常に興味深かったわけですが、ある意味では、データフローというものは、もともと記述されているプログラムを、言ってみればデータを直接にインストラクションの間を渡していくという形でコミュニケーションをやらせているわけですが、デマンドの場合には、逆にほしいデータを、その先行するインストラクションにデマンドを出して、その結果をまたもらってこななければいけないという、いわゆるインボケーションアンドリターンという、往復のためのコミュニケーションがふえてきます。データフローは、デマンドというものをあらかじめコンパイルか計算させる前にアレンジしてしまっています。したがってデマンドを出すためのコミュニケーションを減らしているというふうな見方ができると思います。

質問ですが、そういうわけで、コミュニケーションと並列処理性ということの、アーキテクチャ上の最適ポイントを求めるという観点からすると、デマンド・ドリブンよりもデータ・ドリブンの方がより最適な点に近いのではないかと思います。つまりコミュニケー

ションと並列処理という観点から見た時、確かにデマンド・ドリブンは計算理論的に面白い、より強力な性質を持っていると思います。その点に関してご見解をお伺いしたいと思います。

トレリーバン ここで指摘した重要な点は、コミュニケーションについての決定、そしてパラレルリズムに関する決定をマシンの設計の中でするわけですが、プログラムの表現とは別に決定をやってほしいということです。たとえば、プログラムのオーガニゼーションというのは、私の考えではパラレルリズムとかコミュニケーションとは、あまり関係がないと思います。しかしマシン・オーガニゼーションは、この場合に重要になってきます。例で言うと、ローカルティをうまく利用してグローバルなコミュニケーションを減らしたいとします。これはマシンオーガニゼーションだけに関係しています。プログラムのオーガニゼーションについてどれを選ぶかということは、ターゲット言語を表わすのに、どれが一番効率が良いかということだけから選ばれば良いのです。皆様方の場合に、論理型言語だということですが、ですからパラレルなコントロールフローが、論理型言語を表現するのに一番良いという結論になるかもしれません。パラレルリズムをうまく利用したいということであれば、バケット・コミュニケーションのアーキテクチャをとるのが一番良いでしょう。集中型のものよりは、バケット・コミュニケーションが良いのではないかと思うわけです。

これでお答えになりましたでしょうか。

田中英 私の方から質問したいと思いますが、この質問は、データドリブンとデマンド・ドリブンの比較についてのご質問だったように思います。ですからコミュニケーションという観点から言うと、データ・ドリブンの方がデマンド・ドリブンより良いのではないかと、いう雨宮先生のご意見だったのですが、それについてはどうお考えになりますか。

トレリーバン それは関係がないと思います。直接関連がないと思います。データ・ドリブンとデマンド・ドリブンは、プログラムの表現、実行に関して議論すべき問題ですし、コミュニケーションというのは、マシンのリソースのオーガニゼーションを考える時に出てくる問題だと思います。ですからこの2つの決定は、別々に下されるべきだと思います。

私としては、データ・ドリブンあるいはデマンド・ドリブンが、コミュニケーションに関して違いがあるとは思っていません。これが私の意見です。

これでお答えになったでしょうか。ちょっと確信がな

いのですが。

ファイゲンバウム 2つ質問をしたいと思います。1つはかなり技術的な問題で、1つはもう少し概念的な問題です。技術的な問題というのは、お話になったアレイの2つのうち、LISPマシンそしてオブジェクト・オリエンテッド・アーキテクチャをうまく位置づけることができるでしょうか。この2つの中のどちらかに、うまくそれを位置づけることができるでしょうか。

トレリーバン LISPマシンはシーケンシャルなものです。大体において今のご質問は、MITのLISPマシンが理想的な例としてインプリメントされているわけで、そういうふうなことから答えたいのですが、LISPマシンはコントロール・フローに関して大体インプリメントされています。そして間接参照機構を使って、メモリの中のセルをサポートしています。ですから従来型のLISPマシンを見ると、それはここに位置すると思います。LISPというのは、リダクション(Reduction)のプログラム・オーガニゼーションだと思いますので、理想的なマシンとしては、グラフ・リダクション、そしてある特定のタイプのグラフ・リダクションによって実行することができます。そして一番下のこのところに位置するのではないかと思います。ですからLISPをサポートするマシンは図1の右上ですが、それに加えて右下のところのマシンが必要になるのではないのでしょうか。これで良いでしょうか。

ファイゲンバウム それではSmall TalkマシンのようにObject orientedマシンの場合はどうでしょうか。

トレリーバン これは非常に面白いものだと思います。Small Talkというのは、これに関してはいろいろなことが言われておりますが、オブジェクト・オリエンテッドな考え方というのは、かなり共通点があると思います。つまりたとえば、orthogonalなデータタイプ・プログラム言語に共通性があると思います。しかしながらプログラムのオーガニゼーションについて、オペレーションモデルについての話だと、Actorモデルということになるとと思いますが、Actorモデルというのは、大体このあたりに位置するものだと思います。pureなマシン・モデルではないにしても、大体この上、右か左かどちらかのボックスに入るものだと思います。真中の上の方の部分に入ると思います。これに関しては、かなりの研究が行われているわけですが。

ファイゲンバウム もう1つ質問したいのですが、

明白に質問するのが難しいのですが。

プログラムのアーキテクチャとマシンアーキテクチャをうまくまとめようとなさっていました。そして最後の結論のところでは先生は、6つずつセルがある2つの図を示されました。非常に単純化されていたと思います。もし私がマシンやプログラムのアーキテクトであれば、私はシンプルなワールドに住むべきなのでしょう、あるいはコンセプトに関して、もう少しリッチな世界にいるべきなのでしょう。

トレリーバン 非常に良い質問だと思います。フォンノイマン型のコンピュータを設計する場合の普通のアプローチというのは、シンプル・ワールドに生活するという事です。もちろん人々は最初はシンプル・ワールドからスタートするわけです。たとえばストリング・リダクションの世界からスタートして、その次に自分のシンプルワールドが、サポートしたい言語の特徴をうまく満たさないということがわかってきます。そうするとその時点で重要なのは、余分のメカニズムを加えていくということです。これらのメカニズムのほとんどは、淡々ながら人々によって加えられた、そういう性質のものだと思います。私どものこの研究が正しい方向であれば、誰かがリッチな世界に住むということ始めるはずだと思います。各データ・メカニズムとかコントロール・メカニズムを見ていくと、どういう組み合わせが自分自身の特定のコンピュータに望ましいのかという、そういうふうな考え方に移っていくべきだと思います。

フロア外人 もう1つ先ほどの質問に加えたいのですが、たとえば私が第5世代コンピュータのアーキテクトで、私に与えられた仕事は新しいコンセプトを作り出すということであったとすれば、そうするとアーキテクトの世界の中で、私は一週間に一回、一日に一回、一年に一回あるいは一生に一回一つの新しいコンセプトを発明するということになるのでしょうか。そのスペースの複雑さというのが、どの程度のものかちょっと私にはわからないのですが、どういうふうになるのでしょうか。

トレリーバン 避けるわけではないのですが、間接的なお答えになってしまうと思います。私がここでお話をしていたのは、必ずしも普通私たちが考えるコンピュータ・アーキテクチャではなく、計算のモデルと言われるようなものです。つまりその意味は、私たちのコンピュータは、将来のコンピュータは、より良い計算のモデルをノイマン型のコンピュータより良いものを持たなくてははいけないと思うからです。そのことによって今の質問

者自身のワークをサポートするようなものでなければならぬでしょう。ですから何をしなければいけないかという、アーキテクトとしては計算をエンジニアとして、あるいは数学者として理解する方法を見つけ、したがってここで一番重要なのは、計算のモデルを選ぶ、マイクロコードと呼ばれるかもしれませんが、特定のものにふさわしいようなものを選ぶということです。第5世代のコンピュータの場合には、お互いに違うタイプのコンピュータをprofileすることを考えておられるようですが、たとえばノイマン型のコンピュータ、データフロー・コンピュータ、リダクション・コンピュータなどを一緒に使うということを考えておられると思いますが、各コンピュータおのおのについては、ある特定のタスクをサポートするよう考えられています。ですからこのような特定の個々のメカニズムでそのタスクをサポートすることができるわけです。

もうひとつ上のレベルにいくと、計算のモデルが、そのタスクそれぞれが全部うまく満たされるようなレベルにいくわけです。そのことが一番我々の挑戦に値するような一つの大きな発明だと思います。ですから一つ上のレベルへ上ることが一番の大きい課題ではないでしょうか。

カイファン バーデュー大学のカイファンです。私は2つ質問があります。1つはデータフローマシンについてです。データフローというのは、そしてデータドリブンの概念というのは、単一代入に依存しているので、その値を違う命令セルの中で見ていくわけです。先生はメモリへの要求がテンプレイトのためにこれまでのシーケンシャルのコントロールよりも増えたと考えられますか。つまりなぜかという、大きなメモリというのは正しい方向ではないと思うので、この質問をしたいのです。

もう一つは、お話の中で、正しいプログラムを審こうとする場合に、プログラム・オーガニゼーションをミックスした形でやっていくのが一番良いのではないかとおっしゃったように思います。

そしてそれに対応するアーキテクチャをお話しにならなかったもので、どういうアーキテクチャがそれぞれに、アーキテクチャの結合についてお話しになりました。どういうアーキテクチャが対応するのか、プログラム・アーキテクチャとの対応はどうなっているのか、ちょっとお話ししたいのですが。

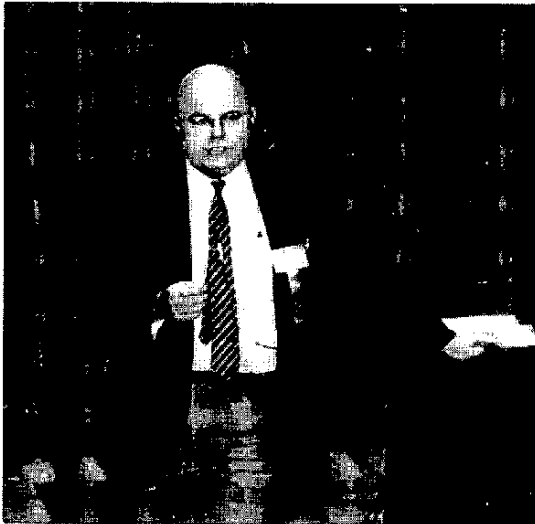
トリーバン まずはじめにメモリの問題ですが、これは1974年以来私たちが考えている問題で、

非常に興味深い質問だと思います。妥当な答えはまだ用意できていない状態ですが、少しでもお役に立ちたいと思い、幾分かお答えしますが、マンチェスター大学で私どもはピュア・データフロー・コンピュータの研究をしていました。そこで私たちはたくさんの数のデータ・トークンがこのマシンの中を流れている状態をちょっと心配し、意識していたわけです。それをシミュレートして、必要な量の情報を得るために、それをやろうとしたわけです。そのほかにもいろいろ切迫した問題がありました。そこで私は、これに関してどのように増大するかというグラフはまだ示すことができませんが、並行性を持たせたいと思えば、ある場合にはペナルティが引き起こされるでしょう。そのペナルティがプログラムの中のスペースあるいは時間という面に出てくると思います。私たちのシミュレーションの中からわかったことは、最初に考えたより悪くないということでした。データフローを考えてみた場合、ライフタイムは非常に短いわけです。マシンの中で非常に速く実現されるのでこれをコントロール・フローと比較した場合には、コントロール・フローの変数のための領域はプログラムのライフ全体にわたってそれが使われなくなるまでずっと存在するので、それをプログラムの一部で使っても良い、そしてそれを全然使わない選択もあり得るといことだと思えます。

第2番目の質問に関しては、非常に良い質問だと思いましたが、少し混同させるような言い方をしてしまったのかもしれませんが、私はなにもプログラミング言語をmixed modelに基づいて行うのが良いと言ったつもりではありませんでした。プログラミングモデルというのは、プログラミング言語の場合に一つの制約になると思いますが、アーキテクチャの中で必要なモデルというのは、ノイマン型のコンピュータモデルで成功した点は、低いレベルのところではプログラムの表現方法と混じり合わないということだったと思います。ですから私たちはノイマン型言語でプログラムすることはしないのではないかとおもわれます。

最近ニューキャッスル大学においてコントロール・フローを選んだということを知っていますが、私どもの論文が書いているのは、これまでデータフロー、コントロールフロー、リダクションについて行った実験の結果で、それらの結果を違ったマシン・オーガニゼーションの分析をしたわけです。長年これらのオーガニゼーションについてシミュレータをいろいろ作ってきました。そしてこのアイデアをベースにいろいろな長所・短所を評価したいと思ったわけですが、そこ

から問題を解決できるようなアーキテクチャが出てくればと思いますが、それに続く期間において、そういうやり方ではなく、一番問題なのは、第5世代のコンピュータのプロポーザルで必要なのは、全部違ったグループについて見てみて、それらの長所・短所を全部見ていくというのが良いのではないかと思うわけです。



(2) アルゴリズム，アーキテクチャおよびテクノロジー

マサチューセッツ工科大学（米国）

ジョナサン・アレン

講演要旨

1. はじめに

第5世代コンピュータ・システムは非常に高度なパフォーマンスを要求されると思われるが、これを設計するに当たって以下の5項目を統合する必要がある。

- ① テクノロジー
- ② アルゴリズム
- ③ アーキテクチャ
- ④ データ構造
- ⑤ プログラム言語

ここでは主にははじめの3つについて述べる。

2. 機能仕様 (Functional Specification)

機能仕様から実際のシステム設計がいかに引き出されるかということが重要であり、システムの自動的な設計の例として以下の2つを挙げる。

① デジタル・フィルタ

バンドパス・フィルタを例にとると、周波数帯域、周波数特性、帯域外減衰などのパラメータ（これらは機能仕様に当る）から実際のデジタル・フィルタを自動的に構成することが可能になっている。但し得られた構成は最適とはかぎらない。

② Simple Circuit Design (MIT)

出力波形の仕様記述など、回路設計者の経験的知識をRule based Systemによって表現することがここらみられてきた。このRuleによってシステムを構成するモジュールおよびモジュール間の接続が自動的に決定され、ある仕様を満たす回路を自動的に構成する。これも最適であるとはかぎらないので、最適化の問題

が残っている。

3. アルゴリズム，アーキテクチャおよびテクノロジーの相互作用

アルゴリズムとアーキテクチャの2つに関しては、まず各々を別々に表現したうえで、あるアルゴリズムを実現するアーキテクチャに変換操作を行い、最適化する。この例としてIBMが行ったゲート・アレイによる論理設計がある。これは機能仕様から自動的に「稚拙な」回路を構成し、その回路を局所的な変換によって改良していくものである。

アーキテクチャとテクノロジーの相互作用に関しては、以下の3つの例が挙げられる。

① ウエファ・スケールの集積における「分割の大きさ」と「冗長性」の関係

この冗長性によってアーキテクチャに潜在的な並列性が生まれる。

② 3次元IC

エピタキシャル・シリコンによる多層化技術によって3次元C-MOSの構成が可能になると、アーキテクチャに大きな影響を与えるだろう。

③ リアルタイム・システム

並列性を導入することによって制御が複雑化することを防ぐために、並列性よりもむしろテクノロジー自体のパフォーマンス（logic speedなど）を利用する傾向にある。

4. アーキテクチャ・ユニットの設計

アーキテクチャ・ユニットの設計に関しては、いくつかの標準モジュールを決めて、構造的に設計する

手法が考えられている。これによってハードウェアの階層的な接続・検証が可能となり、設計者のランダムの詳細設計を軽減できる。

またMITでは、この標準モジュールに対応して、プログラムによってハードウェアの設計を記述するための言語(Design Procedure Language)が研究されている。

CPUの設計についてはMACPITTSアーキテクチャという手法があり、これはCPUの演算部と制御部を明確に分離し、制御部を組合せ回路のみで構成し、演算部は分散ALUとレジスタ・アレイによるデータ・バスによって構成するもので、シンプルな設計を可能にする。

5. 推論システム

第5世代コンピュータ・システムのテーマの1つである知識情報処理に関連して、推論システムの例を考える。

① セマンティック・ネット

プロセッサ・セルをアレイ状にならべたConnection Machineによってセマンティック・ネットを実現する。これはセルの仮想的な接続関係によって知識を表現するものである。

② 音声変換用テキスト

これは入力テキストを分析して音声的なパラメータを抽出するText Analysisとそのパラメータをもとに音声波形を出力する音声合成の2つの部分に分けられる。

自然な音声を得るためにはいくつかのルールによる推論が必要になる。

③ 音声認識/理解

音声認識はパターン・マッチングの問題としてとらえられる。話者の違い、タイミング等の問題はダイナミック・プログラミングによって解決しており、その実現はアレイ・プロセッサでおこなうことが考えられている。

音声理解は語彙の多さ、自然な構文、入力雑音等の問題があり、たいへん難しい。その解決のためには構文、意味、音韻、語彙、声道特性のKnowledge Sourceを用いたインタラクティブなプロセスが必要となる。

6. おわりに

第5世代コンピュータ・システムが直面している多くの複雑さは、様々な結合の程度で相互に作用している

Knowledge Source(アルゴリズム、アーキテクチャ、テクノロジー等)の問題としてとらえられる。

これを解決するためには、的確な知識の表現、および効率的な推論機構が必要である。

質疑応答

ビーベル 先生の基本的な考え方について伺いたいと思います。私自身の理解で申しあげたいと思いますが、ジョン・デリンガーたちの研究、および先生が先ほど言及されたところの、プロトタイプなものを作るというところに先生の基本的なフィロソフィがあると思うのですが、私が伺いたいのは、このフィロソフィは現在の技術に対するものなのでしょうか、それともこのフィロソフィは、たとえば10年後、第5世代コンピュータが実際に実現した時に使われるようなものなのでしょうか。

なぜこのような質問をしたかということをもう少し説明しますと、このフィロソフィは問題の論理的な記述からはじめて、ナイーブな解を作っていくというやり方です。今は専門家が細かな処理をやる形ですが、むしろ機能的なレベルにとどめておいて、先生の言われたような変換は機能レベルに置いておいて、解はそうではないと可能ではなくなるのではないのでしょうか。

つまり解がその記述のまわりで右往左往してしまうようになってはいけないと思ったわけです。ですからもし私がこういうことをやるのであれば、可能性のある解を除外してしまう可能性があると思うのですが。

アレン 確かにそのへんには疑問な点がいくつかあります。まず最初に、私はこのようなアーキテクチャのトレードオフの表現方法というものが今後とも続くものだと考えているかどうかということについては、私は今後とも続くものだと思っています。つまりアーキテクチャのトレードオフというものを体系的に見ることは必要だと思います。それから低レベルの細かいこと、たとえばレイアウトのような細かいものを見るという点について、下のところまで専門家が見なくても良いのではないかというのは確かにそうだろうと思います。今の傾向としては、設計者が低レベルのものはやらない、例えばマスクの幾何学的な構成などというものはやらず、むしろシステムにそれをやらせ、コンパイルさせて、評価を行わせて、設計者がやっているレベルでの答えを出させる。一体電力消費はどれ

だけになるのが、遅れはどれくらいになるかといったようなレベルでのものに引き下げさせる。このように自動的に低レベルのものを作らせるというのが、このVLSIソフトウェアの狙っているところです。今のところはまだ十分に効率的なものではありませんが、人手による設計でやられているものをソフトウェアでやっていこうではないかという考え方です。むしろ私は完全にやっているわけではありませんが、設計者のインタフェースをもっと高いレベルでやっていかなければならないというのが私の今の考え方です。

フロア外人 バイブラインとリソースシェリングの話がされましたが、どのように考えておられるのですか。IBMのマシンなどでは自動的にやっていると言われましたが、そのことは先生の言語ではどういうふうに指定され、どういうふうに自動化されているのですか。

アレン 良いご質問だと思います。われわれがどのように複雑な、あるいは普通のバイブライニングを指定して自動的に問題を解決しているかということですが、今は自動的にやっておられません。今後の研究上の問題としては、現在やっている指定方法はハードウェアの設計言語あるいは指定方法(仕様)の段階ですので、このような要素についてはデザイナーが指定している段階ですので、今後の研究課題にしたいと思います。

フロア外人 シーマ・クレイはバイブラインの天才だと思いますが、彼の方がマシンはCADでやっているよりも良いものを作っていると思います。

アレン

コンパイル・デザインの10~20%については批判が集まっており、まだ商品開発に使用できるようなものではありませんが、設計所要期間あるいは設計コストは無視できない重要なものであると思います。チップだけを考えても、ウェーファシステムでやっていくと、何百万というトランジスタが必要です。ですからそこにはある程度の複雑性というもの存在着しているわけであり、高レベル技術を探して、こういう低レベルの細部を機械に落せるようなものを考えていかなければならないと思います。われわれはまだ初期の幼らん期にあると思うので、そういう問題からさらに学んでいかなければならないと思います。ソフトウェアと比較して例えて言うことは直接的すぎると思います。ただ人手による設計ではうまくいかないものが出てきていると思います。ZilogのZ-8000が最も新しいものだと思いますが、人手による設計のもの

は、これ以降のものは出てこないと思います。あのような複雑なものを手作業でやるという、アセンブリ言語でソフトウェアを作るというのは、ちょっと量が多すぎると思います。

また、実際に産業界でやっておられる方から伺うと、ゲートアレイのシステムをマシンでやった方が手でやったよりも今でさえも効率が良いということがわかっています。この技術はまだ制約されたものですが、大いに発展している段階と言えるのではないのでしょうか。

木戸出 ICの設計の時の分割(partition)についてお聞きしたいと思います。分割についてはデジタル処理においては、デジタルデータのビット精度における分割と、回路的な空間的な方向での分割とあると思いますが、画像においては、たとえば入力データの精度によって4ビット、8ビットなどいろいろあり、たとえば4ビットのデータ用のある回路がほしいし、それをまた積み重ねれば8ビットにしたいという状況もたくさんあります。そういう方向での分割の問題と、機能的にはわけられなくて、要するに空間的な、ある1つの機能を持ったものとしてやりたい場合があると思いますが、そのへんについての考え方と、分割における問題では、VLSIにおいては、セルフテストという問題があり、全部中でテストしなければならないような状態になると思いますが、そういった考えを入れて、VLSI設計におけるICの分割についてお考えをお聞きしたいと思います。

アレン 2つ質問があったと思います。1つは分割の効用、もう1つはチップのセルフテストについてのご質問だったと思います。

まず分割については、ここで一般解とも言えるような答はありません。FFTを実装するというのは、むしろテクノロジーによって変わってくるわけです。並列処理方式や逐次処理による実装が行われている分野では、これをモジュールに分割していくことは大変重要です。逐次的な算術演算の方が並列的な算術演算よりも簡単です。また、新しい結果として生まれてきているようなもの、MITでやっているものも使いたいと思います。各サーキットのイールドを予測するというテクノロジーの開発をMITでやっていますので、これを使いたいと思います。

セルフテストについては、セルフテストはなければならぬと思います。また、最近の研究でヒューレット・パッカートの32ビットのマシンで、最近発表されたものの中には、9000ワードのリード・オンリー・メモリーがチップに乗っており、ここではセルフ

テストが必ず行われるようになっていきます。ですからチップがパワーオンされるたびに、かなりのルーチンが走るようになっていきます。

フロア外人 先生はご発表の中でデータ構造についての話を無視されましたが、これは時間がなかったからでしょうか、あるいは過去にハードウェアの人たちがかなり触れたからですか。

アレン 両方です。5つの中で取り上げませんでした。時間がなかったからなのですが、これについては一般解が述べられるとは思わなかったわけです。特にアーキテクチャやテクノロジーのレベルで語れないと思ったからです。抽象的なデータタイプを与えられた場合に、これがアーキテクチャのレベルでそれほどはっきりした結果とか成果が出ていると思えませんでしたので。

フロア外人 3DC-MOSについてコメントさせていただきたいと思います。先生、アナログタイプのプロセッシングとデジタルプロセッシングのタイプのを組み合わせる可能性についてはどのようにお考えですか。

アレン ええ、実際にそういう形で動いていると思います。アナログとデジタルを組み合わせるといのは出てきていると思います。ただし複雑になるとか、やはりパフォーマンスへのニーズというものは高まってくると思います。C-MOSのプロセスは、現在のところはAからD、DからAのコンバージョンが、同じチップで、同じコンバージョン・テクノロジーでやっている、そのシングルプロセッシングチップでやるというのは、今でもやられていますから、これからはもっと増えてくるだろうと思います。

N-MOSとC-MOSと少しバイポーラ的なものを組み合わせてやることにより、必要なものを選択することができます。たとえばドライバーのためにはバイポーラを選ぶというようなことができます。

フロア外人 先生のDPLの言語で、こういうミックス・デザインについての問題へのアタックはできるのででしょうか。

アレン 直接にはできませんが、それをやる必要があるかどうかはわかりません。まだそういうテクノロジーが出てきていませんが、出てくれば良いと思います。

フロア外人 もう少しVLSIのデザイン仕様記述言語について詳しく説明していただけますか。

アレン 高レベルの機能的仕様記述のことですか。

フロア外人 はい、そうです。

アレン ドキュメントがありますので、御紹介してもよいのですが、レジスタ・トランスファ・タイプのものでLISPに付加されています。特別のフィーチャでデザインワークのハードタイプになっていないものはないと思います。そのレベルならばナチュラルな形で書けますし、LISPは非常に便利だということを使っただけです。ただ低レベルではあります。

これは、LISPの言語で、レジスタ・トランスファ・タイプのものでデザインワークのハードタイプになっていないものはないと思います。そのレベルならばナチュラルな形で書けますし、LISPは非常に便利だということを使っただけです。ただ低レベルではあります。

IV.2 パネル・ディスカッション II

第5世代コンピュータ・アーキテクチャ



座長：慶応義塾大学

相磯 秀夫

パネリスト：イリノイ大学

ブルース・H・マコーミック，東京大学

後藤 英一

ニューキャッスル・アボタイン大学

フィリップ・C・トレリーバン，(株)日立製作所

浦城 恒雄

マサチューセッツ工科大学

ジョナサン・アレン，電子技術総合研究所

内田 俊一

相磯 そろそろ午後のセッションを始めさせていただきますと思いますので、どうぞ席にお戻りいただきたいと思います。

午後のパネルは第5世代のコンピュータ・システム
のアーキテクチャについてです。

それでは第5世代コンピュータ・アーキテクチャ
に関するパネル討論を始めさせていただきますと思
います。

パネリストには全部で6人の方をお招きして
おりますが、皆さんから向って左側からご紹
介申し上げます。

まず一番左側はマコーミック教授、もう改
めてご紹介するまでもなく、この会議の招待
講師、それからその右側はトレリーバン教
授、今朝お話を伺いましたのでご紹介は多
分必要ないと思います。

その右側がMITのアレン教授、アレン
さんも先程お話をいただきましたので、ご
紹介をする必要がないと思います。

日本側は3人の方がおられまして、一番
左側が東京大学の後藤英一先生です。後藤
先生は大変ご高名な方ですから、ご紹介
するまでもないのですが、簡単にお話し
申し上げますと、現在東京大学理学部情
報科学科の教授をされております。

パラメトロン of 発明者としても有名で
すし、最近では記号処理、コンピュータ
あるいはVLSIの電子ビ

ーム・リソグラフィの研究などをされて
おります。

それからその右側は日立製作所の浦城
さんです。浦城さんは1959年東京大学
の理学部物理をご卒業になりまして、日
立製作所にお入りになりました。現在日
立製作所神奈川工場の企画部長をされ
ております。

従って日立の商用計算機の企画を担
当されているというふうに伺って
おります。

それからその右側は電子技術総合研
究所の内田俊一さん、もう前にご紹
介しましたので省かせていただき
ます。

司会は私、慶応大学、相磯秀夫が担
当いたします。

このセッションは時間が限られて
おりますので、まず最初に、既に
沢山のご質問を伺っているの
ですけれども、その辺を踏まえ
まして、まずそれぞれのパネ
リストに10分以内で見解をお
話しいただきたいと思いま
す。

それではマコーミック教授から
お願いします。

マコーミック 第5世代のコン
ピュータ・システムというのを
3つの主要なサブパートに分
けて考えたいと思います。3
つの層からなるようなケー
キとして考えたいと思いま
す。

一番下の所では5Gのパーソ
ナル・コンピュータを見たい
と思います。これはつまりオ
フィス・オートメーション・
マーケットとして非常に大
きな価値がある

と思います。

しかし、このOAの部門、市場を狙うというのは、これは80年代の仕事であります。90年代のことではありません。従いまして、この5Gのパーソナル・コンピュータの目的というのは、いまあるLISPマシンの一般化、又は汎用化ということになると思うんですけれども、この面で商用化といった面でマーケット・シェアを得るためには、かなりの妥協点といえますか、歩み寄りが必要だと思います。

従いまして根本的・基本的な研究の多くは、このコンピュータ・アーキテクチャに関してのものですけれども、これは第2番目の層にあると思います。

これはつまり5Gのスーパー・コンピュータ・プロジェクトです。これまで25年間にスーパー・コンピュータ・プロジェクトがいろいろあったと思います。また、いろいろなアチーブメントがあったと思います。

ファイゲンバウム先生が何年前か前に、こういった面でスーパー・コンピュータのプロジェクトで何か成功例があったかというようなことを私にお聞きになりましたけれども、その時私が思い出したのものには、例えばユニバックのLARCというプロジェクト、IBMのSTRETCH、CDCのSTAR、それからパロースとイリノイ大学のILLIAC IVがあったと思います。またハネウエルとMITのMULTICSというのもありました。

こういったものは、あまり成功を収めていません。完全な目標というのは達成していません。商用化もされていないわけですが、しかしこういったプロジェクトが、これまでアメリカがコンピュータ工学という部門で優勢を占めてきた、そのベース、基礎となったと思います。

プロジェクトとしては、私、その中ではSTRETCHというプロジェクトが良かったんじゃないかと思いますが、というのはこのテクノロジカル・ベースを引き延ばすのに非常に重要だったからです。

このことからIBMが作った非常に素晴らしいシステム、メモリーシステムが生まれてきました。

ですからこの第5世代コンピュータ・システムのプロジェクトも、このテクノロジカルベースとなるべきSTRETCHプロジェクトとなっていたかと思えます。

アメリカでも日本でも同じですけれどもスーパー・コンピュータというフレーム・ワークを確立することが必要だと思います。といいますのはスーパー・コンピュータであるということは政府のものは非常

に長い時間がかかるということをおわかってくれますし、そしてそういったことから生産するのが非常にむづかしいということがわかっております。

官庁というものは自分が誤りをおかしたということをお認めたがらないものです。そういうことから、私はこの研究をやる場合にスーパー・コンピュータというのを目標におくのがいいと思います。

そうしませんと開発に必要なタイム・フレームというのが問題になってきてしまうのではないかと思います。

このようなプロジェクトのゴールというのは、外側では曖昧さ、そしてあいまいな面をどうしても保っておくというのが好まれるようです。

内部的じゃなくて外面だけは曖昧さを残しておいた方がいいと思います。

その次に、この第5世代のコンピュータ・システムの第3番目の層は認識(cognitive)コンピュータという部分だと思います。認識コンピュータ、すなわち自然言語の処理、音声認識とかコンピュータ・ビジョンは、私が話したことですけれども、こういったことを含むようなものでないといけな思います。

ですから、この5Gのプロジェクトが人間がいま示している、人間のみが示している知識とか認識の特徴をリアルタイムで見ているようなものがあると思います。そういうものを目指していただきたいと思えます。これは、これまでどのコンピュータもできなかったものです。

このプロジェクトを進めていく上では、やはりかなり挑戦していかなければならない課題があると思えます。

まずコンピュータの大きさが大きくなり、やはりコミュニケーションというものが必要だと思います。

神経科学、心理学、あるいは認識科学といったような広範な範囲に渡る専門家のコミュニケーションが必要だと思います。

FGCSを1つのケーキと例えるならばケーキの第3層に入っているものとしては、国際協力というものが必要だろうと思えます。

特に商業的なものを考えた場合にロンドンで考えますと、やはり国際協力というものが必要だと思います。

ちょうどサイエンスや物理学の分野におきましては、2つの文化があったと思えます。いわゆる加速器のエンジニアと、それから物理学者という、いわゆる実験

を中心にやる人達という、この2つの文化は、全く異なった問題を解決していたわけでありますが、お互にお互の文化を尊敬して研究を長い間に渡って行なってきたと思います。

私、コグニッション、あるいはパーセプションという認識という意味から申しますと、コンピュータ・サイエンティストは一つの役割を果たし、またコンピュータのエンジニアもまた別の段階を果すであろうと思います。

シミュレーション、モデリング、エミュレーションといったようなところにおきましては、言語学者やその他の心理学者がまた別の形で研究しているものを、やはり1つに凝集していかなければならないと思いますので、そういうような学際的な環境というものが、生まれてくる必要があると思います。

やはりこういう環境ができるということがFGCSというケーキの3つ目の層の中になければならないのではないでしょう。

相磯 マコーミック先生は第5世代コンピュータのプロジェクトの意義あるいは学際的な研究の核としてのプロジェクトの性格、それから第5世代コンピュータの研究開発プロジェクトに対する期待をおっしゃっていただいたと思います。

どうもありがとうございます。

それではトレリーバン教授をお願いします。

トレリーバン 私の考えを申し上げます前に、ここでちょっと、まず最初にこのような会議にお招きいただきました皆様に対して感謝をしたいと思いますし、通訳も最善をつくしてどうということが話し合われているかを伝えてもらっていると思いますので感謝をしておきたいと思います。

さて、私の発表の中で申し上げましたように、ニューキャッスルの私の大学での研究は、ほぼ同じような動機づけを持って研究をしているわけです。

特にこのFGCSのアーキテクチャに関するサブコミティの研究の動機づけと同じような動機づけを持って研究をしております。

私も通産省の資金をいただきまして、こういうコンピュータ・アーキテクチャの研究ができれば、どれ程素晴らしいかと思えます。

私も皆さまと同じ分野の研究をしておりますので、是非そうならばよいと思います。

私自身とてもプロジェクトにアドバイスなどはできません。そこで委員会のこの計画、全部は理解できたとは申しませんが、皆さまのご理解にも役立つと思えます。

英国の政府関係者もここに来ておられると思いますので、こういうことが、ここで話し合われたということ指摘しておきますと、後々私も自分のためになるのではないかと思いますので、そういう観点から一言申し上げておきたいと思えます。

Programming Languages	Single			
	Conventional	Assignment	Applicative	Logic
Program Organization	Control Flow	Data Flow	Reduction	?
Machine Organization	Centralized	Packet Communication	Expression Manipulation	?

図1 Spectrum Of Considerations For Fifth Generation Computer Architecture

これがプログラミング言語のクラス分けです。特にこの会議の皆さまには興味があると思うものを整理いたしました。コンベンショナルな言語はコントロール・フローによってサポートされておりますし、それか

らマシンのオーガナイゼーションは、主に集中化されたものであります。

その次には、関数型言語として2つのクラスがあり

ます、これは2つに分けられます。

1つはSingle Assignment Language(単一代入言語)でありまして、データフローコンピュータに使われており、データフロー・コンピュータは通常こういうようなクラスのマシン・オーガナイゼーションでサポートされています。

それからもう1つ applicativeな言語というものがござります。そのプログラム・オーガナイゼーションとして、最もよく適しているのはReductionであります。

このReductionは、Expression Manipulation Machine で一番よくサポートされていると思います。

ところが?マークのところは埋められないので非常に残念だと思うんですけども、これは、ここが今後論理型言語に向って研究されていくという1つの方向を示していると思います。

現在、集中型のコントロール・フローのコンピュータがありますので、単一代入言語の場合には、データフローの、あるいはReductionのプログラムを作らなければいけないわけです。つまりコントロール・フローの上に積んで、こういう形のものを作らなければいけないわけで、そうしないと効率がよくないわけです。

となりますと、第5世代コンピュータ・アーキテクチャのプロジェクトは、私に言わせれば、これらの領域をすべて勉強して、各マシンを研究する必要があると思います。

そうすることによって、異なったマシンが、常に共

存できるような状態を考えていかなければならないと思います。

そして、それは可能であろうと思いますがバケット・コミュニケーションのコンピュータでデータフローも、コントロールフローも、サポートできるようなものができるかよいのではないかと思います。

つまり、かなり効率よく両方をサポートできれば、コンベンショナルな言語も単一代入言語もサポートできるわけであります。

ということは、かなり効率がよくなるということになります。この考え方に添って考えていきますと、10年間のうちあるいは少し早目にこの?マークの部分の埋めることができるかもしれません。

そうすれば、このマシン・オーガナイゼーションも語る事ができるかもしれません。

そして、そのマシン・ストラクチャというのは、コントロール・フロー、データフローそしてロジックフローのものも含めてサポートできるようなものになるべきでありましょう。

従ってそうなれば、かなり効率のよい形でこれらの言語をすべてサポートできるようなマシンというものが語れるような時代が来るだろうと思います。

コントロールフローとデータフローをサポートするわけですが、ここでは小さな命令を例にとってみました。これはデータフローでプログラムを表現したものであります。

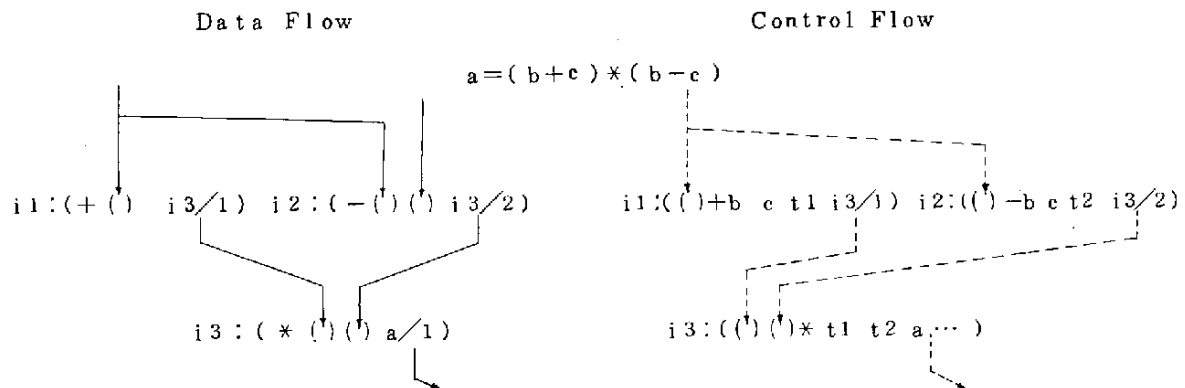


図2 Combining Data Flow and Control Flow

これは今朝使ったものと全く同じでございますが、唯、コントロール・フローの解釈の仕方が少し異なっております。

これはちょっとした小さなコンテナで、トークンが入るようになっています。データ・バリューは入っておりません。単なるシグナルです。

この命令はこのローカル・コントロール・トークンがあった時に実行されるわけでありまして。この命令は、ここがあれば実行されるわけでありまして、この命令は2つのコントロール・トークンがないと実行できないわけでありまして。

この2つのrepresentationを使ってみますと、この2つを組み合わせることは、決してむづかしくはありません。

私はこれをコンバインド・オーガニゼーションと呼んでいます。データフローとコントロール・フローを合わしているわけですが、コントロール・フローに合っているビットにはコントロール・フローを、データフローに合っているビットにはデータフローを使おうというものであります。

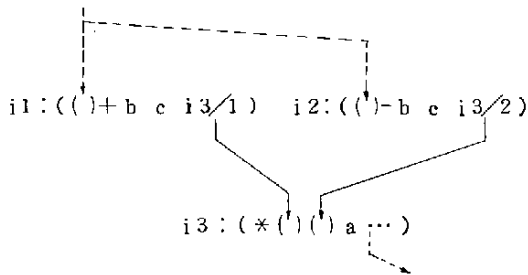


図3 Combined

ここではコントロール・シグナルでこのインストラクションを実行しまして、メモリー・ロケーションから参照しまして値を持ってくるわけでありまして、これは一部分の結果をバス・オンするだけでありますので、ここではデータフローのメカニズムを使っているわけでありまして。

そしてそのあと実行でかけ算をやるわけでありまして。それはデータフローでやるわけでありまして。そして結果をメモリーセルにストアするわけでありまして。あるいはデータ・トークンとしてバス・オンするというものでございます。

従って、私がここで申し上げたかったことは、この2つを組み合わせることを1つのコンピュータでできるのではないかとということです。

従って少なくとも、第5世代コンピュータのアーキ

テクチャにおきましては、同じことをすべてのものについて行なうことによって、マシン・オーガニゼーションとして、論理型言語だけではなくて、コンベンショナルなもの、データフロー・コンピュータのもの、Reductionコンピュータなものまで、まとめたようなものが作れるのではないかと考えています。

以上です。

相磯 多分第5世代コンピュータの一つの目標であります述語論理型言語のためのアーキテクチャへのアプローチというのでしょうか。

もう少し表現を変えますと、もう少しいろいろな観点から、ずっと研究をされた方がいいのではないだろうか、という話であったように承りました。

この辺につきましては、後程また内田さんの方からいろいろとコメントがあるかと思えます。それではアレン教授お願いします。

アレン 私、いくつかの分野に限って申し上げたいと思えます。

まず、この会議の主催者の方々にお招きいただきましたことを大変感謝いたしております。また素晴らしいご口上をお寄せいただきましてありがたく思っております。

私、ここでは、この10年間というのは大変興奮するような10年間であろうと思えます。

私、今朝はテクノロジーの面で私がどういうふうな興奮を覚えているかということの、いく分かのことで申し上げたかったわけですが、技術的な能力というものが急速に成長しておりますので、数多くの可能性が生まれてきている時代であるといえるのではないかとと思えます。

大きなものを考え過ぎているのではないかとという批判も寄せられるかもしれませんが、でも私は、その批判は正しくないと思えます。

やはり何かをやろうという時に物を小さく考えるよりは、大きく考えた方が能力をさらに充分に発揮して、いかにそれに挑戦できるかということを考えることができると思えます。

さて、エンジニアリング・デザインの問題につきましては今朝、私、少しお話しをいたしました。

複雑なシステムをいかにデザインし、それがいかに正しいものであるかということを確認していく必要があるわけでありましてけれども、それを実際に物を作らないでやらなければいけないわけでありまして。

チップなりウェーブのコントロールを考えていかなければならないわけです。

そしてFGCSというのはウェーブよりは勿論大きなものですから、この技術をできるだけ正確に推測してやっつけていかなければならないわけです。また予測していかなければならないわけです。

デザインには様々なレベルがあります。実際の物理的なアーティファクトから離れば離れる程選択がむずかしくなります。正しく選択をすることが、むずかしくなります。ここが正にむずかしい部分なわけです。

しかし、この10年間の間にかなりの発展を遂げて参りました。

そして私は今後このプロジェクトの研究結果や、その他の分野の研究によりまして、かなりのデザインの能力というものが開発されることを期待するわけがあります。

私共、エンジニアリングの問題をここで取り上げているわけですが、これは単に素晴らしいアイデアや素晴らしい概念的な基礎を持っているばかりでなく、それが実際にどのように機能するかということが重要なわけがあります。

デザイン・システムというのは、このプロジェクトの中で進化していくと申し上げてもいいと思います。

デザインを作っていく、それについて経験を重ねることによって、さらに開発していかなければならないと思います。

長い期間に渡るプロジェクトを立てておきながらデザインをいろいろ考えて、実際にはデザインせずに短期間でも経験を積まなければ、悲劇を招くことにもなると思います。

他のプロジェクトでも、実際に途中で作って見なかつたために失敗してしまったという例もあると思います。

様々なアーキテクチャが考えられます。様々な選択肢があります。

私の知っている日本での研究を見ておきますと、様々なalternativeを実験してこられたと思いますし、このプロジェクトにおいても、そういうものは試行錯誤の上で進められていくことを期待するものであります。

さて、知識工学というものが、何回か話題にのぼってきました。この知識工学というものが、まだ充分ではありません。

私はSpeech Processingの立場から申し上げたいと思いますが、まだまだこの分野では未知のところが残っております。

Speech がどのようにジェネレートされ、どのよう

に認識されるかということが、よくわかっておりません。

いま現在のところの、かなりの印象深いようなシステムはできてはおりますけれども、しかしながら、話者あるいは聞き手が自然言語の中で充分にやりとりをしているだけのものはまだできておりません。

FGCSではシステムを、いま現在ある知識を使うようなものを作るだけでなく、知識をインテリジェントに獲得するようなものを考えていかなければならないと思います。

1つ例を挙げますと、Speechのタイムコースというものは、人間がやる以上にうまく把握することができます。

1つには、子音と母音の長さを計るというようなことができます。これは実際に現在もスペクトログラムを音声について作ることによって行なわれております。

これをうまくいけば機械に入れて、マシンフォームにして分析するというようなことができます。ですから自動的に音声波形の分析ということは、できるようになってきております。

そしてユーザとのインタラクションも行なわれるようになってきておりますけれども、こういうことがもっと進められなければ、我々の考えているような期間の間にDesign SpecificationがFGCSで考えられているようなものとして実現されることはむづかしかろうと思います。

従って我々はこういうマシンを使ってデータを獲得する方法を考えていかなければならないと思います。

あたかも、こういうデータがすべて揃っているというような形で開発をしていってはならないと思います。

私が冒頭に申し上げましたように、この10年というのは、大変エキサイティングな10年になろうと思います。

期待よりも、より以上に素晴らしいものになるかもしれませんし、そうでないかもしれませんが、私はこういうエキサイティングな時代に生まれたことを幸せに思っております。

ありがとうございました。

相磯 第5世代コンピュータ・プロジェクトのとりえ方といいますが、それをエンジニアリング・デザインという立場からご意見をいただいたわけです。

デザイン・メソドロジーそのものもどんどん進化をするだろう、それは知識や経験をベースにして、やはり進化をしていくものではないだろうか。

ですから第5世代コンピュータもやはり知識の、あ

るいは経験というものを蓄積して、そしてそれを活用しながら発展するようなものになってほしいというお話だったと思います。

ありがとうございました。それでは後藤先生お願いします。

後藤 まず初めに私がどういうことをしているか、何をやろうと思っているかというところから話します。いまFLATSと呼ばれるマシンを作っております。

専用の数式処理のためのマシンです。数式処理というのはユニークな特徴をもっているのではないかと思います。

というのは、これはシンボリックな計算と、それからニューメリカルな計算、両方を必要としているからです。勿論シンボルを使っているのでシンボリックだということは、当たり前なんですけれども、コエフィシエント(係数)なども使っていますので、数学的な部分もあると思います。

数式処理のもう1つの特徴といえますのはロジカルな深さが非常に深いということです。

例えば、ユークリッド・アルゴリズムを考えてみてください。あるいはよく知られているニュートンの反復法のアルゴリズムを考えてみても、非常に数式処理に、どちらも重要なアルゴリズムなんですけれども、どちらも純粋にシーケンシャルな性質を持っています。

よいコンピュータ・サイエンティストの中には、例えばストラッセン先生などは、この問題を研究なさいました。しかし誰もまだそれらを並列化するアルゴリズム、2つのアルゴリズムにユークリッド・アルゴリズムとニュートン反復法両方を並列形式にtransformすることには、まだ希望を見出していない状態です。

従いまして、田中先生がマシンについての話をして下さったんですけれども、1,000データフロー・プロセッシング・エレメント(P・E)を持つようなものがあるというふうに聞きましたけれども、しかしながら、残念なことに数式処理の場合は、例えば1,000のP.Eを持っていたとしても、999のP.Eは、残りの1つのP.Eについてユークリッド・アルゴリズム、あるいはニュートン反復法などやっている間待っていない必要ありません。ストップしてしまいます。

従って、この数式処理タイプのmanipulationで必要なのは、パラレルリズムからはあまり得られないと思います。

そうではなくて、非常に早いシーケンシャルなマシンが重要だと思います。

これから、今までにお聞きしたいことについてのコ

メントを言います前に、もう少し私の経験について、お話ししたいんですけれども、10年程前、私は数式処理に非常に興味を持つようになり、どのようにしたら、記号処理を行なえるかという研究を始めました。

そこで私がまず始めたのは、L6と呼ばれる言語のimplementationでした。

そして私、どのようにしたらストレージ・マネジメントをするかいわゆる「buddy system」と呼ばれるものについて論文を書いたんですけれども、この結果はCACMで出版されました。これはL6を始めたということ自体、間違いだったと思います、いまから考えますと。

なぜ、L6にしたかったかという、私は非常に面白い映画をベル研究所(BTL)で作られたL6についてのものを見まして、非常に感銘を受けて、そういうことを始めたわけです。

後になってそれが、それ程のものではなかったということがわかったわけなんですけれども、それは、私達の目的には適当でなかったと分りました。そして私はHLISPというシステムを書きました。次にREDUCEという数式処理言語、これはユタ大学のHearn先生が開発された言語でありますけれども、これを活用いたしました。

そして、私共のシステムは日本の大学のコンピュータ・センターで使われており、日本で行なわれております殆どの数式処理は、私共が開発したシステムで行なわれております。

さて私がいくつかのコードを最適化しようとしたときに、これはLISPのシステムですけれども、ちょっとうまくいかなかったわけでありまして、うまくいかなかったという不安があったわけでありまして。

これは第1世代のコンピュータでありましたので基本的なLISPのオペレーションをスピードアップするために特別なレジスタとハードウェアをつけるということが、易しいと思ったわけです。

そこで私、コンピュータをこういう目的のためだけのものを作ろうと考えたわけなんです。その過程において、いくつかの意思決定をしなければならませんでした。

まず最初にデータフローはやらないことにいたしました、というのはデータフローではあまりにも実験的すぎ、ちょっと高くつきすぎるということでやらなかったわけでありまして。

またパラレルリズムもあまり役に立たないと思ったわけでありまして。従ってコンベンショナルなパイプライ

ンのマシンでやることにいたしました。

そういう意味ではコンベンショナルなマシンと言えるかもしれません。

第2にコンカレント・ガーベージ・コレクションも、使わないことにしました。

マシンは従って数式処理中にガーベージ・コレクションでちょっとくらい止まりましても別にロボットとつなげてるわけではありませんので問題にはならないわけです。

さらに3番目の点として申し上げたいことは、ハイレベル言語のエグゼキューション・マシンではありません。

インタープリタや、コンパイラを書いたことがおありになる方ならば、どんな言語を使った場合でもありますがけれども、ダイレクト・エグゼキューション・マシンというのは、ハードウェアのインタープリタ・マシンでありますけれども、言語のインタープリタというのは、くり返しパーズングといったようなことをやるわけであります。

ですからコンパイラにそういう簡単な作業はやらせた方がいいわけであります。

ですからダイレクト・エグゼキューション・マシンというのはリソースの無駄使いであろうと考えたわけであります。

そこでダイレクト・エグゼキューション・マシンを作ることは、やりませんでした。

データタイプ・チェック、あるいはバウンダリー・チェック、アクセス・チェックというようなものがハードウェアでパラレルで行なわれるようになっております。

それ以外にGOTOについて一言申し上げたいと思います。このGOTOは私の名前と発音が似ているので、よく冗談の種にされるんですけども……。

さて、マシンはダイレクト、ハイレベル、エグゼキューション・マシンではありませんので、GOTO FreeではなくてFree GOTOなわけであります。

つまりGOTOへ行く時間というものが吸収されるとどのようにGOTOを、どれだけ入れても時間がかからないような形になっているわけであります。

また我々のマシンはフリーコール、フリー・リターンになっています。大きなマシンのソフトウェアを見てみますと、例えばプログラムなんかで見てみますと、多量なサブルーチンがコールされています。私のもの場合には、フリーコールのフリー・リターンということになっていますから、オンラインのコードの expan-

sionなんていうようなことは心配しなくてもいいわけであります。

サブルーチンへジャンプして戻ってくるというようなことは、なんら問題ないわけです。

そしてまたハードウェア・ハッシングというのをインプリメントしようと思いました。私、いくつかの話の中で、コンテンツ・アドレスメモリーというのを聞いたんですけども、私はこれはハードウェア・ハッシングでできるのではないかと思います。

私が思いますには、もしコンテスト・アドレス・プログラムでは、非常に大きい問題はパワーの問題だと思います。すべてのビットがパラレルに動いたとしますと、すべてのアクティブなものがアクティブになりますので、少なくともパワーの消費量がRAMよりも1,000倍くらい大きくなります。

100万ビットのチップを持っていた場合に比べて非常に大きなパワーの消費になります。ですから、どうやったらいいかと言いますと、コンテンツ・アドレスングをするやり方としては、ハッシングがいいと思います。

それから、もう一つソーティングについて、ハードウェアのソーティングについて話を聞いたことがあるんですけども、私、思いますには、ソーティングは、RAMの中にストアされたデータには必要ではないと思います。

これが有用なのは入出力の目的だけだと思います。と言いますのは、ディスクではなくてメインRAMメモリーにおいては、どういう場合でも、ハッシングができると思います。

その方が、ソーティングよりも安いし速いと思います。ですから、ソートエンジンは私達作りません。ハードウェア・ハッシングをやります。

それからもう一つコメントがあるんですけども、私が聞いた中で、データ・ストラクチャの話の中でバッグ(BAG)といていたものがありました。

それは本質的にはリピーティド・エレメントのセットです。これをハードウェアのデータ・ストラクチャの中に入れていいのではないかという示唆がありましたけれども、私は、それはインプリメントする価値はないと思います。何故なら、その概念が狭すぎると思います。

一般にセット・エレメントからのマッピングにおいて、BAGというのは正整数のマッピングのわけですけども、なぜ、正整数だけマッピングするんでしょうか。

なぜ、正整数のかわりに、ジェネラル・ストラクチャのためのポイント・ツー・ポイントを使わないのでしょうか。

コンテンツ・アドレス・マッピング、あるいはテーブルという言い方もするかもしれませんが、そういうことが私のインプリメントにおいて、BAGのデータストラクチャのインプリメントに関する、私の意見であります。

もっと一般化して、やらなければいけないと思います。

もう少しお話ししたかったんですけども時間がありませんので、ありがとうございます。

相磯 後藤先生は、いまFLATSと称する数式処理のためコンピュータを研究されておられますが、それに関するいろいろなご経験から、いくつかの問題点をご指摘いただきました。例えば、数式処理などでしたら、むしろ非常に早いシーケンシャルなマシンの方が効率的かもしれないし、それからハッシングの機構の方がメイン・メモリーを作る場合にはいいかもしれない。

その他、高級言語マシン、あるいはデータ・ストラクチャに関するいくつかのご指摘をいただきました。どうもありがとうございます。

それでは浦城さんお願いします。

浦城 私、日立製作所の浦城でございます。

先程のご紹介にもありましたように、日立製作所というメーカーに属しまして、ここ20年来、メインフレームを中心として、アーキテクチャのデザイン、あるいはプロダクト・プランニング(製品計画)といった仕事をしてまいりました。

第5世代の研究開発プロジェクトには、調査研究委員会に昨年参加させていただきまして、時々意見は申し上げてきました。本日はそういったバックグラウンドを持った人間の、個人的な意見として、第5世代にどういったことを期待し、どういった感想を持っているかということをお話ししたいと思います。

まず最初は第5世代に何を期待するかということです。いろんなことがすでにレポートに述べられていますが、私は特につぎの3点を強く期待しております。

最初はコンピュータのアプリケーションを質的にも量的にも拡大するということに大きく貢献していただけないかということでもあります。

2番目は、ソフトウェアの生産性を飛躍的に改良できないかということでもあります。

3番目はややありふれておりますが、マンマシン・インタフェースを大巾に改良するということでもありま

す。

特に申し上げたいのは、最初の2点でありまして、コンピュータ・メーカーの人間として、非常に危機感を最近抱いております。勿論第5世代にいたる迄に、さらに10年くらいあるわけでありまして、私共はこういったような問題についての挑戦を現在の技術の延長上において、いろいろ試みるでありまして、第5世代の主な狙いが、従来の技術的な延長をさらに飛躍するような、あるいは技術的な延長線上にはない新しいブレイクスルーを求めるということに大きな意義があるというふうに伺っておりますので、その点に関しましては、大きな期待を持っております。

一方ハードウェアのテクノロジーについては、別途、国からも援助していただいてやっているわけですが、VLSIやハイスピード・デバイスは、さらに20年くらい、つまり今世紀の間は、さらに発展を続けるであろうと、予想しております。

しかるに、こういった技術進歩をただちにコンピュータの分野に充分にいかしていくことが段々むずかしくなりかけており、アプリケーションを拡大し、ひろく適用していくのに積み残しと申しますか、アプライしないうちに次の技術進歩があるといったようなことが、すでに80年代の初めから起りつつあります。

そういった状況において、この2つの期待が出きかけたわけでございます。

次はアーキテクチャについてでございます。私などは古典的な意味、つまり論理的なマシン・オーガニゼーションあるいはソフトウェアとハードウェアとのインターフェースといったふうにとらえておりますが、さらには物理的なマシン・オーガニゼーションの方法もアーキテクチャと呼んでおりますし、また、もっと広い意味でも、アーキテクチャは最近使われているようであります。ここでは、古典的な意味にとりまして現在のアーキテクチャの延長上で、本当にいいのかということに関しましては、やっぱりそのうち充分でなくなるのではないか、そのうちに暗礁に乗り上げるのではないかという懸念が非常に強いわけであります。

それは先程、第5世代への期待として申し上げましたようにソフトウェアの生産性を飛躍的に改良するためには、それからアプリケーションを特に知識ベース処理の分野に拡大していくにはどうしても、アーキテクチャの分野に革新的なブレイクスルーが必要であろうということを、技術的というよりは、歴史的つまり過去のいろんな進歩過程を見ながら、肌で最近強く感じております。

そういった立場から、新しいアーキテクチャというのは、まず第1に真に重要な問題とニーズに対してその解決に有効であるということが、非常に重要なポイントであろうと思います。

次にアーキテクチャは、ただ適当なものであるというふうなことではまずいわけでありまして、いままでの連続的な延長を超えて、新しいアーキテクチャに挑戦するのであれば、やはりここ10年来の情報科学の進歩、特に基礎的な分野での理論に充分準拠し、裏付けられていないと、そのあとさらに10年、20年という、新しい世代においての基本的なアーキテクチャにならないんじゃないかと思えます。

3番目はアーキテクチャの設定にあたってはハードウェア・テクノロジーの傾向というものは充分考慮すべきことではありますが、それに対してあまり依存性の強い形で、即ちハードウェア・テクノロジーとのトレードオフを追求しすぎて決めると、それは必ずしもよくないんじゃないかと思えます。これは従来アーキテクチャの改良に際し、ハードウェア・テクノロジーのことを考慮しすぎて、結果的にその寿命があまり永くなかったという、いくつかの経験を持っているからでありまして、充分考慮すべきだけでも、あまり依存性のあるものであってはいけないという意見を持っております。

次はアーキテクチャ研究分科会の提案する主要な研究課題であるVLSI-CADについて述べたいと思います。

VLSI-CADは非常に重要であるということは充分認識しております。唯VLSI-CADにとって重要なことは、やはり物を作るという立場からは、使えるということが非常に重要なことでもあります。今回の第5世代の開発プロジェクトは先程申し上げたような期待をになり新しい機械を開発することに大きな目標があるわけでありまして、VLSI-CADについてはこれ自身は非常に重要な研究テーマであることは認めるわけですが、何と言っても、道具であり、実際に使える形にしていかなければいけないと思えます。

一方VLSI-CAD自身は革新的な進歩も必要かも知れませんが、我々としては、VLSI-CADは、80年代のものにしなければ、競争から脱落していくという環境にあるわけでありまして、絶えず進化的にインプリメントをしていながら、VLSI-CADを開発していくものであろうと思っております。

次はマシン、オーガニゼーションについてのコメントであります。これには、いろいろなコメントを個人

的にはもっていますが、時間がございませんので、1つだけ指摘したいと思います。

きょうのアレン先生のお話しにもありましたように、VLSI技術の適用として、いろんな形の機能的なモジュールというものが、今後沢山開発される可能性があるわけでありまして。計算機のオーガニゼーションに関しては、従来より汎用的なもの、専用のものがあり、過去はどちらかといいますと、専用のものが消えて、汎用的なもので、専用機能を何とか実現しようといった動きがどちらかと言えば主流であったわけがあります。しかし今後は専用のものが、機能的なモジュールとか、あるいは機能的なエンジンとして、非常に重要な働きをもたらすのではないかと思えます。

しかし、どのようにしてこの汎用的なものと専用のものが協力し合っていくか、あるいは専用のものをどのように定義していくかということに関するプログラム言語上、あるいはマシン・オーガニゼーション上の問題というのが重要な問題になるのではないかと考えております。

最後はプロジェクトの実行に関するコメントであります。パネリストの方からもいくつかご指摘がありましたように、まず第1点としては、充分な調査のあとはKey issuesに対して人的あるいは財政的なリソースを集中すべきであろうと思えます。

2番目の点は、こういったプロジェクトの実行過程においては、評価をきちんとしながら進めることが重要で、特に早い段階から、評価の方法や体制を確立しておくことを提案したいと思います。

3番目の点は、第5世代のプロジェクトというのは、非常に広範なものを本来含んでいるわけでありまして、これを全部同じようなレベルでは、とてもやりきれないわけでありまして、これには国内外のいろいろなプロジェクトでの成果を必要に応じて、できるだけ多く受け入れながら、本当の目的というものを追求していくという姿勢が望ましいのではないかと思えます。

以上簡単にコメントをさせていただきました。

相磯 ありがとうございます。

浦城さんは冒頭に個人的な意見だとおっしゃいましたが、かなりメーカー・サイドの意見だったというふうにも感じられるのですが。

第5世代コンピュータプロジェクトに期待するのは大きく分けて2つあって、1つは質的な改善、それからソフトウェアの生産性の改善、この2点はどうしてもやってほしいと。

しかし、それを解決するためには、いろいろな関連

する技術に、いろいろな問題点があると、例えば、アーキテクチャはやはりブレークスルーする必要があるでしょうし、VLSI-CADの進め方にもやはり現実的なとらえ方があるのではないだろうか。

その他、マシン、オーガニゼーションなどに関する観測をお話いただきました。

ありがとうございました、それでは最後に内田さんに、実は内田さんにはいままで皆さんからいろいろなご質問を受けておりますので、それを踏まえて、ちょっとお話しをいただきたいと思っております。

内田 まず最初にプランを作成しました委員会のメンバーのひとりとして、皆さんが、私共到大変建設的なご意見と、それから場合によってはマンパワーをどうするか、お金は調達できるのか、というご心配をいただきまして、大変ありがたく思っています。

どうもありがとうございました。

私は、きのう発表した者のひとりですが、私共の発表の仕方に対して、いろいろなご注文がございまして、それは殆どのものが、もっともなものでございますが、そして、それに私達は充分、応えるだけの時間を与えられておりませんけれども、そのうちの一部でも、お答えできればと思います。

まずいくつか、お断りしておかなければいけないことがあると思います。

その1つは、この赤い文字で書いた「what is most appropriate for our initial state?」でございますが、これは、PROLOGやDFMが出発点としてのもの、そのものではなく、PROLOG like, DFM like という意味ですが、いろいろな人が、いろいろな日本語で発言いたしまして、それが英語にとらえられるときに、英語ではmustという単語になってしまいます。

必ずしもそれは英語で言われる程のmustではないと、場合によってはplan toであるとか、would like to do であるとか、そういうことであると、受け取っていただきたいと思えます。

さて、本題に戻りますが、我々が2年間の委員会の議論で決めましたのは、我々のイニシャル・コンディションであります。

そして、そのイニシャル・コンディションのうち多くのものも、Key issueを決めたのでありまして、それだけに限定するものではありません。

従いまして、私の話の中でも、まず新しいアーキテクチャのmajor issueといたしまして、高級言語と、

それからパラレル・アーキテクチャというのをお話しするわけですが、これについても、いま申し上げましたような一つの出発点であるということをお断りしたいと思います。

まず、いろいろな皆さんから、なぜデータフローを選んだのか、それから、なぜPROLOGのような述語論理型言語を選んだのかという質問をいただきました。

それを選ぶ過程について、どういう意見が棄てられたのかというようなことを問われましたが、どうも日本的な決定のシステムではそういうのというのは、必ずしも明白ではないのであります。

明白でないからこそ、協調していけるのではないかと、私は個人的に思うわけです。

本日、私は公式な見解を述べることは、要求されておりませんから、極めて個人的に答えられますので、自由に発言してよいと思えます。

まず、高級言語というのを考えてみますと、一番最初に高級言語に要求されるものといえますのは、何と言いましても、まず、アルゴリズムを自然に記述できるかどうかと、さらに従来のアルゴリズムでありがちでありましたフォンノイマンのスタイルの中にとらわれて書くのではない、そういうものが必要です。

多分、今朝の講演でもアレン教授はそのように言っておられました。

アルゴリズムというのは、インプリメンテーションと異なりまして、それ自身がパラレルなアーキテクチャ、データフローのようなアーキテクチャ、ノイマンアーキテクチャ、それでちょっとカテゴリーが違いますがVLSIのアーキテクチャというようなところに適用しようと思ったときに、それ自身の記述がそのまま使えなければいけない、というようなことであります。

次に、これとは違うかもしれませんが、並列処理とか並行処理と呼ばれるようなものの記述ができるというようなこと、こういうことが、やはり大事だと思います。

そしてさらに、私はアーキテクトでありますから言語の特徴としての素晴らしさもさることながら、パラレル・アーキテクチャに対するcomputationalなモデルをきちんと与えてくれるような言語が欲しいわけがあります。

従いまして、FGKLという核言語を決定するに当たっても、我々はランゲージ・フィーチャをどういうふうに集めてこようかということを議論したわけです。

言語の中には、述語論理型言語とか、関数型言語、

それからオブジェクト・オリエンティッド言語とかいろいろあるわけです。

例としてはPROLOGですとか、LISPですとか、データフロー用のSingle Assignment LanguageそれはMITのIdだとか、日本ではVALIDというようなのがあります。

それからSmall TalkやAdaがあるわけですが、このうちいろいろな言語はいろいろな素晴らしいフィーチャを持っているわけで、可能か不可能かは別として、なるべく多くの良いランゲージ・フィーチャをとり入れていきたいと思っているわけです。

その支柱というものを、PROLOGに定めたというふうにご理解いただきたいと思います。

従いまして、私の昨日の講演の中でも、オブジェクト・オリエンテッドなものを取り入れたい、そのためのアーキテクチャも考えたい、そういうようなものを含めているわけです。

次にこの言語から、パラレルなアーキテクチャへの橋渡しをしてくれるcomputationalなモデルを考えると、バラレリズムというものを、どう考えていくかということが問われます。

これは極めて多くある論理でして、先程も後藤先生が数式処理というのは論理のレベルの深い問題も多くあり、並列処理ができない、すなわちバラレリズムが含まれていないような問題も多いんだという話。

それから人工知能の研究者の一部の方から人工知能の研究というのはパラレルというか、非決定的なものを、決定的なものにしていくのが研究だとか、いろいろあります。

しかしながら依然として、その一番下の知識情報処理等の上のレベルから下のレベルまで、ずっと見ていったときの、比較的下の方のレベルを考えますと、従いましてローレベル・オブ・インフレンスというわけですが、そこには実際にプログラムを書いてみても、かなり多くのバラレリズムを見出すことができると思います。

しかしながら、そのバラレリズムというのは、昔から問題になっております。こういうtree(木構造)の探索空間を並列にサーチするという問題でありまして、これは今までのハードウェアの並列処理をやっているものにとっては、極めて問題があるように見えます。

実は私も昔は、F.F.T.プロセッサですとか、図形処理用のプロセッサを開発しておりまして、そういう意味からしますと、今朝のアレン先生のおっしゃった

ような分野の、ちょっと古いハードウェアを研究していたわけでありまして。

そういう人から見ると、treeをサーチして得られる答えは、ここで1個でいいわけです。

たとえ並列にサーチしたとしても、殆どのものの結果は無駄になるように見えるわけです。

例えば、F.F.T.の処理では全部の結果は、殆どアウトプットのアレイとして用いられるわけですから、これは極めて納得がいくわけですが、このような並列処理では殆どが棄てられる。それだったら、人間がもしこの部分をやれば答が得られるという部分へ直接的にいくようなアルゴリズムがあるなら、それはそれを機械にのせて下さい。そうすればアーキテクチャをやるものは、あまり迷わないですむというわけですが、そろそろそういう時代でもなくなってきたのではないかと思うわけです。我々はこれから先に、やはりこういうようなcomputationの環境というのを考えざるを得ないのではないかと思っているわけです。

PROLOGというような論理型言語は、そういう意味でも、コストパフォーマンスとか、そういうものの基準を、ちょっと変えなければいけないような新しい状況を我々に要請するわけです。

そういう意味でも興味深いということで、ロジック・プログラミングというものが、あがっていると理解しております。

しかしながら、このような探索空間の広さというのは、通常極めて急速に膨張いたしまして、コンピュータではとり扱えない、すなわち一昔前でしたならば、それは冗談であるというふうに言われても仕方がなかったわけです。

そこで、その辺につきましては、やはりこの探索空間をより狭くするというような意味でのメタ・レベルのコントロールというものが、プログラミング言語の中に組み込まれていかなければならないと思います。

それがPROLOGにどう組み込まれるかということは、PROLOGと言わないでFGKLと言った方がベターだと思いますが、そこに組み込まれていくかというのは、これからの第5世代の重要な研究テーマであります。

それからさらにトレリーバン先生も同感いただけると思いますが、プログラミング言語を考える方は、是非より精密なcomputationalなモデルも合わせてお考えいただきたいと思うわけです。

その中には、やはりアーキテクチャの方からいくとコンピューティング・エレメントは、常に有限であり

まして、それを考える上でのマッピングだとか、リソースアロケーションの strategy というもの、合わせてやはり考えていただけるとありがたいと思います。

それから、これはちょっと純粹にハードウェアの問題になるかと思いますが、やはり並列処理の形態は我々が従来慣れ親しんで参りました行列計算よりは、やはり例えば quick sort のような処理に近いと、もっと irregular かもしれない。

私がここで irregular と申し上げたのは、行列計算に対して quick sort の方は、やはり irregular であろうと思うからです。

さらにこれから我々が目指すのは、もっと irregular かもしれない。

ある瞬間には 1,000 個の並列性のあるジョブがあるかもしれないけれども、その次の瞬間には 1 個しかないかもしれない。

後藤先生が言われましたように、平均するとやはり 90 個くらいは遊んでいるプロセッサが出るのではないかと思います。

しかし、私はそれでもいいのではないかとそういうふうに思っております。

このような状況でデータフロー・マシンというのを考えたいと思っております。

それで今申し上げましたような効率に関する仮定がないのであれば、やはりきわめて専用的な CRAY-1 のようなマシンを作らなければいけないのではないかと思います。

しかしながら、やはり知識情報処理とか、記号処理のような世界におけるスーパー・コンピュータというのは、クラシックなアーキテクチャの概念から言えば、無駄な計算ばかりやっているようなコンピュータというふうに言わざるを得ない。

そういうようなコンピュータがやはり、これからの知識情報処理におけるスーパー・コンピュータではないかと思うわけです。

従いまして私達の定義では、データフロー・マシンと一言で簡単に言いますが、実はメッセージ・ドリブンとかデマンド・ドリブン、ストリーム・ドリブン、これはパイプライン処理のようなものであります。こういうものも、やはり考えざるを得ないだろうと思います。

これは先程、トレリーバン先生がいろいろなものをお示しになりました、マトリックス、 3×3 のマトリックスの全部を勉強しなければいけないと、正にそういうことを示しているんだと思います。

それ自身要求されるものとしては、先程申し上げましたような irregular なパラレルリズム、それからやはりプロセッシングパワーを改善できなければいけない。

それからこれはちょっと変な話と思われるかもしれませんが、応用問題は irregular であるにもかかわらず、マシンを regular にしたいというわけであり、irregular なマシンを作るとはやはり原理的にはちょっとむずかしいのではないかと思いますので、やはりその辺はプログラミング言語とか、ソフトウェア、それからさらには、何らかの形での妥協が必要だと思えます。

アーキテクチャ屋は常に妥協しているのでありますから、別に今さらこのくらいのことで驚くことはありません。

最後にそういうような意味からデータフロー・マシンを推論マシンに適用するというようなところで、いまのリクワイアメントをチェックしてみますと、その computational なモデルから考えて、このような irregular なパラレルリズムというのを扱えるかという点に対しては、「イエス」だと思います。

これが「イエス」であるというマシンはあまりありません。データフロー・マシンもまだ出来上がったものではないので、そういう意味からでは、確定的な意味での「イエス」ではないのですけれども、現在もとても期待のできる候補だと思います。

又、more power が得られるかと、more power とはパワフルという意味ですね、するとこれも多分「イエス」だと思いますが、まだ？ マークであります。

次に、いくらかかるかこれについては、まだまだわかりませんが、これがわかるくらいなら、もうすぐ作っていいという感じですが、これはやはりこれからの VLSI とか、その他の先程のアルゴリズムの話で言えばその探索空間をどのくらい効果的に縮め得るか、というようなことと密接に関係するわけでありまして、それはハードウェアのコストということもさることながら、そのようなアルゴリズムの研究と並列処理のアルゴリズムという研究が、極めて重要だと思われま。

そしてレギュラー・マシン・ストラクチャで実現できるかという点は、これはたぶん「イエス」であります。この辺はかなり irregular になるかと思えます。

やはり純粋な関数型言語を、そのまま完全に分散したような、グローバルメモリーが一切ないような、完全な分散というのは、やはり極めて理想的なもの

でありまして、やはりその辺は、少なくとも電源スイッチは1個にしなればいけないのではないかと思いますから、そういう意味での、セントライズされたコントロールというのにはあり得ると思います。

少々長くなりましたが、あまりキチンとした答えではないかもしれませんが、ちょっと私の考えを述べさせていただきます。

どうもありがとうございました。

相磯 ありがとうございます。

内田さんは、第5世代プロジェクトの最初のスタートポイントとして、取り敢えず核言語を設定したわけですが、その核言語を設定した背景、その考え方、それからその中に含まれております並列処理、パラレルプロセッシングとデータフロー・マシンとの関係、この辺についての考え方をお話しいただきました。

ありがとうございます。

これで一通り6人のパネリストの補足見解をいただいたんですが、ここでいよいよディスカッションに入りたいと思います。

まず今までのお話しに関連しまして、ご質問あるいはご意見などがございましたら、お受けしたいと思いますが、いかがでしょうか。

淵さん。

淵 電総研の淵です。

後藤先生のお話しに、ちょっとご質問したいんですが。

FLATSでいろいろ決定されたお話しは非常に有益であったわけですが、FLATS自身は数年前に計画されて、いま出来つつある素晴らしいコンピュータだと思っています。

そういう点では、そういう決定されたということは非常に正しいし、例えば我々のプロジェクトでも、最初の数年に作るマシンというのは、かなり似た決定をすることになると思います。例えばシリアル制御であるとか、ハッシュを使うとか。

ですがお聞きしたいのは、もし1988年頃に決定されるとすると、FLATSと同じ決定を再びされるかどうかということをお聞きしたいんですけど。

後藤 88年になったら、どうかということに対しては、数式処理マシンを作るのは、やはりさっきの決定でいいと思います。

それから私の決定の中で、要するにバラレリズムを追求しない、コンカレント、ガーベージ・コレクションを追求しないという点は、応用分野が違えば当然バラレリズムができる応用分野にしたらいし、コンカ

レント・ガーベージ・コレクションが必要なロボティクスならしなければいけない。

ですからあとのポイントはメインメモリに入っているものについてはハッシングをやった方がソーティングより常に得だから、と思います。

quick sortというのは出力にものを出すときだけいるので、メモリーの中に入っているものは、要するにハッシュ構造でした方が安くて速いのです。

それから先程も言った中で要するにバラレリズムとコンカレント・ガーベージ・コレクションのことは、これは応用分野が特別だから、あとの申し上げたことは、つまり高級言語のダイレクト・エグゼキューションなんていうのは、つまらないから恐らく他の人間は誰もやらないでしょう、ということです。

相磯 どうもありがとうございました。

いかがでしょうか、他にご質問。

それでは後ろのローソンさん

ローソン 私のコメントは、内田先生の最後の方の話しに関連があるんですけども、それはirregularなバラレリズムを持つということでしたけれども、uniformなコンピュータ・アーキテクチャの場合を考えまして、それがirregularなバラレリズムということですけども、私はあまりそれについて楽観的な考えではありません。

というのは、つまり私達この職業に携っておりますものは、これまで歴史上いくつかの新しい共通なストラクチャの考えでのアイデアを發明してきました。それに、すべての問題にそれを適用しようとしてきました、例えば、それでILLIAC IVをすべてのタイプのプロセッシングに適用しようとしてきたわけですが、それは成功しなかったので、結論として出てくるものは、これが反証を受けるということになるんじゃないかと思っています。

そして、もう1つトレリーバン先生とアレン先生になんですけれども、両方のご講演の中で、どういうタイプのアイデアを使って、このアルゴリズムをやっていくのかということで、非常に現実的にお話しになりましたけれども、もう少しナチュラルなマッピングをコンピュータショナル・ストラクチャとインプリメンテーションとの間に必要だということだと思います。

この点に関してコメントが出てきたんですけども、そこで何らかの共通のレギュラーのストラクチャ、つまりヘテロジーニアスなストラクチャが何らかの問題を起こすのではないかと私は思います。

そこで現在の状況の経験から、私達は多くのこれま

でのアーキテクチャのコンセプトに縛られている状態だと思えます。これまでのハードとかソフトの考え方に、それがナチュラル・マッピングのアンナチュラルなマッピングになってしまっているのではないかと思います。

アレン そのとおりだと思います。

2つのテンションがあると思えます、今朝私が申しましたように、問題の本質を考えると、そのコンピュータショナルなストラクチャやインプリメンテーションでマップをやらうとしますと、悪いところは歴史から見ますと、どの問題領域におきましてもモデル化する技術というのは、まだ充分ではなく、何千年前に戻りましても、プロセスをモデリングして、そしてその時のテクノロジーを調べようとしたわけでありまして、それが風車の技術であろうともそれよりもっと近代的技術であっても、同じようなプロセスをやってきたと思えますし、条件は変わっていないと思えます。

トウイデイ マコーミック先生がおっしゃった点なんですけれども、スーパー・コンピュータがこれまで、過去ではターゲットを満たしてこなかったということでしたけれども、この5世代のマシンが10年に引き延ばして考えると、その目標に、プロジェクトのマネージメントということで、コンティンジェンシ・プラン（予備的計画）がなければならないと思うんですけれども、第5世代についてのコンティンジェンシ・プランというのはあるのでしょうか、それについてお伺いしたいと思います。

相磯 ちょっとその前にマコーミック教授に質問させていただきたいんですが、スーパー・コンピュータというのは、どういう意味でおっしゃったんでしょうか。

普通スーパー・コンピュータというと大変スピードの早い科学用のコンピュータのイメージを描くんですけども。

マコーミック 私は、第2段階、第3段階、このプロジェクトの第2、第3段階でのことを申しましたわけで、パーソナル・コンピュータではないというプロジェクトの後期でのコンピュータという意味で申し上げたわけでございます。

いわゆるクラシックなスーパー・コンピュータ・プロジェクトが実ってきて何かできるのではないかとというようなことで申し上げたわけです。

私がさっき申しましたことの要点というのは、社会的なメカニズムを考えると基礎研究が行なわれなければならないわけなんですけれども、その基礎研究が互いに無

関係な形になってしまっているくらいがあるのではないかと思います。

ですから技術の進歩を図っていくためにはコンピュータの分野では、もっと関連づけたような形で行なわれなければ、ならないのではないかと、それから、そのマシンが何をやるかというようなことを、よく見ながら進めていく必要があるということを、私、言いたかったわけでございます。

相磯 先程の方のご質問に対しては、内田さん何がありますか。

もう一度ご質問を簡単にしてくれませんか。

トウイデイ 私の言いたかったことは、長期計画でございますね、これは10年間ですから、10年以上に延びるかもしれませんけれども、過去の例を見て見ますと、メーカなんかやっているような場合には、いわゆるRCAなんかではそうですが、そういうレベルでも大きなコンピュータから小さなコンピュータに移ったというようなこともあったと思うんですけれども、私の言いたかったことは、つまり、このプロジェクトのターゲットは、どのようなコンティンジェンシ・プランを入れていらっしゃるのでしょうか、つまりインフォメーション・テクノロジーで、もしプランとのギャップが出てきたら、そのギャップをどういふふうに埋めるような計画を考えておられるのでしょうか、マシンの。

相磯 もう1つちょっと質問させていただきたいんですが、コンティンジェンシ・プランというのは、どういう意味ですか。

トウイデイ 2次的な計画というふうに申したらよろしいでしょうか。

内田 勿論、私共は未知の技術が1985年頃出てくるであろうことは予想しております。

そして、場合によっては、我々が今考えていることが、1989年には正しくなかったというようなことが、わかることがあるかもしれません。

すでに説明いたしましたように、10年間を3つの段階、初期、中期、後期に分けております。各段階の計画は、各段階に入りましたときに再検討を加えることにしております。

と申しますのは、私共はかなりの進歩を予想しているからです、かなりの技術進歩が達成されることを予想しておりますので、それを今ここで予測することはむづかしいわけです。

私共は、できるだけ段階の初めのところで計画の見直しを正確に行ないたいと思っております。

そういうことでよろしいでしょうか。

相磯 簡単に要約しますと我々はプロジェクトを3つの時期に分けて、それぞれの時期のスタートポイントで技術の見直しをいたします、ということだと思います。

それでよろしいでしょうか。

トウイデイ 部分的にはお答えいただいたと思います。

もう1つお尋ねしてもよろしいでしょうか、パーソナル・コンピュータについて伺いたいと思います。

先程のマコーミック先生はO.A.について、10年のうちには出てくるだろうと言われましたけれども、どのようなパーソナル・コンピュータをお考えになっておられるのでしょうか。

そのパーソナル・コンピュータというのを、どういうパーソナルなものと考えておられるのでしょうか、みんなが買うようなパーソナル・コンピュータというものを考えておられるのでしょうか、あるいはいわゆる現在の電卓マーケットに代るようなものを考えておられるのでしょうか。

内田 一部分的にはそうでございますけれども、殆んどはそうではございません。

パーソナル・コンピュータの開発の初期の段階ではソフトウェア・エンジニアにいいプログラム開発環境を提供しようという考え方です。ですから最近のスーパーパーソナル・コンピュータ、例えばM.I.T.のLISPマシンとか、DORADOといったようなものを考えているわけですが、唯、違いがあります。

我々のパーソナル・コンピュータは、我々の核言語をサポートするものであります。核言語をFGKLと呼んでいますけれども、これはPROLOGみたいなものでございます。

しかしパーソナル・コンピュータは、ファームウェア・ベースのものであり、従ってもし研究者が他の言語をインプリメントしたい場合、例えばLISPとか、Adaといったようなものをインプリメントしたい場合には、それができるようになるというものを考えております。

サボリ 第5世代コンピュータを知識ベースマシンの並列処理のものであり、VLSIアーキテクチャ的なものであると考えれば、(トリーバンさんとお話しをしまして、パラレルリズムについて申し上げたいと思っておりましたが)私がコントロールレベルで考えていたパラレルリズムというのと、後藤先生がおっしゃっていたのとはよく合っていると思います。

後藤先生は限られたパラレルリズムというものが、達成可能であるということを描きされました。それはシェアード・オブジェクトがあれば、パラレルリズムは可能であると思います、シェアしているものがなければパラレルでものをやるということではできないわけでありまして。

例えば、現在世界にあるタイプライターが今別々なことをやっていると同じようにであります、例えば、私はいま、ひとりで話していて、他の方は聞いていらつしゃるというのは、パラレルリズムができないから、こういう形になっているわけでありまして。

従ってアプリケーションによって、パラレルリズムができるオブチマムレベルというものがあると思います。

タスクによっては、パラレルリズムに向いているものもあると思います。例えばFFTのターリー・チューキーのインプリメンテーションなんていうのはそうだと思いますし、他には全くパラレルリズムには向いていないというものもあると思います。それがパラレルリズムについて申し上げたかったことです。

それからアーキテクチャとの関連で申しますと、例えば極簡単なN個のカラムを足すというようなタスクを考えてみたいと思います、1つのプロセッサならば何秒間かですとできるとします、それが1つの方法、あるいはN個を2で割るというような場合にlogの底が2のサイクルのカラムがあったとします。

しかし、日立で作られたもの、あるいはニクスドルフの作ったものでもいいんですけども、1つのプロセッサがNステップであると、そしてこのラムダNの加算をやるのにエラーがなかったとします。

そしてN倍2のlogNのエラーレートが出てくるわけでありまして。

ですから私がアレン先生の言われましたVLSIアーキテクチャについて申し上げたいのは、マルチプルパラレル・プロセスをやつて、そのセルフチェックングをやるということになりますと、このラムダの値が、ラムダの2乗になってしまうわけで、2乗になってしまうわけです。

ということは、アレン先生に具体的に質問をさせていただきたいわけでありまして。浦城先生とアレン先生に伺いたいんですけども、質問は、どの程度までセルフ・チェックングを、VLSIのインプリメーションの中で考えておられるのでしょうか。

アレン まだやっていないので、やらなくてはいけないということはおかります、おっしゃっていることを伺っております、私、サーキットとして通常

のテストタイムの中でやらせるというようなことを考えていたというのは、1つのフォールドでもあったかと思えます。

というのは時間の関数でありますし、そのメカニズムをどうしたらよいかということについて、私自身確たる確信があるわけではありません。

まず、どういうふうにやるかということは理解しなければ、事情を理解しなければできないわけでありまして、ある程度のエラー、特に現在のメモリーのテクノロジーでは、エラーが出てくるだろうと思えます。

それはデバイスに固有の理由があるかでありまして、ビットを高密度メモリーでストアしますと小さな容量でやりますと、そこにやはり問題が出てくると思えますので、フォールト・トレランスを入れるということについて反対ではありませんけれども、この問題をその理由が何であるかということを理解するところから始めたいと思えます。

そして、そこにいまはメモリー・テクノロジーから研究すべきではないかという印象を持っております。

浦城 エラー・チェックもいろんな段階があるわけでありまして、現在すでにシステムとしてはかなりのエラー・チェックが含まれております。VLSIを使う際にも、いままでいろいろ積み上げてきました、故障やどこか異常があればチェックするという技術をますます強化するでしょう。今日現在すでに全体の論理の、例えば30%とか、40%に及ぶ範囲が大体エラーチェックやエラーディテクティング、診断などのために使われており、この傾向は今後共続くと思えます。

各チップでシークエンシャルなエラーチェックを使用される前にやるという考え方もあります。マイクロコンピュータのようなものでは、そういう論理を内蔵して使われる前に最初チェックするという方法です。このような機構がVLSI自身に内蔵されるということも考えられますが、VLSIがコンポーネントに適用された場合には、システムとしてもいろんなチェック機構を含み、全体としてのチェック機能は、システム・レベルでも行なわれるということになるんじゃないかと思えます。

つまり各コンポーネントレベルでのチェック、あるいは、サブシステムとしてのチェック、あるいはシステム全体としてのチェックといったように、いろんな段階で行なわれるようになるんだろうと思っております。

相磯 よろしいですか。

それでは大須賀先生どうぞ。

大須賀 東京大学の大須賀でございます。

第5世代のコンピュータが知識情報システムであるとしますと、これはロングターム・メモリーを中心にものごとを処理していくというふうな、処理体系の変化が起っているんだろうと思っておりますね。

そうしますと、知識ベースを探索するということ、大量のデータ知識が入っている知識ベースを探索するということに、かなり時間がかかってくるのではないかと、そうしますとパラレルイズムの問題に関係しているんですけども、多数のプロセッシング・エレメントを並べても、それから共通のKnowledge Structureをシェアしなければいけないという状況のもとでは、パラレルイズムは生きてこない。

従って、むしろこれからの問題というのはメモリーを違った目的で、いかに共通にシェアするかという問題にあることのように思いますが、この点いかがでしょうか。

内田 いま大須賀先生がおっしゃったとおりだと思います。

残念ながら、私共のアーキテクチャのレベルでは、いま大須賀先生が言われたような、知識情報処理における、いろいろな処理の形態といいますが、どのくらいの量がシェアされるのかとかそういうのはよくわからないわけです。

それは、やはり先程パラレルイズムがどのくらいあるのかとか、それからローソン先生がコンピュータのオーガナイゼーションのイレギュラリティというような話を指摘されましたが、そういういろいろなものとみな関係してくるのではないかと思います。

一番簡単なケースはやはり、私が昔やっておりましたF.P.Tプロセッサのようなケースが一番簡単でありまして、あの場合には、問題の性質とその記述、それからマシンのストラクチャというのは殆どすべてよく対応しておったわけです。

そうしたときに、やはりアーキテクチャの方としては、常にそういうバーチャル化のレベルが少ない方が能率はいいし、見通しがいいと思いますが、知識情報処理でも、これからソフトウェアの研究がどんどん進むにつれて、現在まだ何も見えていないようなところが、徐々に見えてくるのではないかと確信しているわけです。

中には、セマンティック・ネットワークを直接VLSIで実現しようというような試みもありますし、この

ような例がいくつも行なわれれば、徐々にそこに共通に見られるような性質というのが、見出せるのではないかと思います。

そういう意味でやはり第5世代の中心は、アーキテクチャをやる、前に誰か言いました分業をやりすぎではいけないと、すなわちソフトウェアをやっている人も、アーキテクチャをやっている人もみんないろいろ入りまじって、その情報交換を密に進めていかなければ、ならないんじゃないかと思います。

現在考えているマシンにおける分散というのは、やはりハードウェアの立場として、パラレルリズムがなるべく多く行なえるような基準に考えがちです。

まだ、アプリケーションとどうマッピングさせるかのようなものは欠けているというのが実のところですから、やはり第5世代の研究の中では、そういうようなところを積極的に進めていかなければいけないと、いうことになっております。

直接のお答えではありませんが、よろしいでしょうか。

相磯 ありがとうございます。

大須賀先生は東京大学で実際に知識ベースシステムの研究をされているわけですが、その研究の過程で同じ情報を同時に呼び出したい、沢山のプロセッサが同時に呼び出す特殊なマルチポート・メモリーのようなものも必要になってきていると指摘されています。このへんは、これはやはり実際に経験をされないとわからないところでございますから、是非そういうご経験を我々の方にフィードバックしていただきますと、我々は新しい研究課題に取り組むことができると思います。

どうもご質問ありがとうございます。

他にいかがでしょうか。

レナード 私、浦城先生に伺いたいと思います。

プロジェクトの実行につきまして、先生が先程おっしゃいましたことは、リソースをKey issuesに集中するというのを先程おっしゃいました。

調査後にKey issuesについて、焦点を合わせるべきだと言われましたけれども、先生にとってのKey issuesは何なんですか、5Gプロジェクト自体が優先的なものなんですか。

浦城 第5世代に関する2年間の調査報告によりますと、第5世代としてやらなければいけない開発テーマというのが非常に沢山抽出されております。2つのレベルで分類されていますが、大きな分類のテーマ中にいくつかのオルターナティブがあるので、その中

でやはりどれかをやるという絞り方をすべきでありましょうし、それから研究対象となる高いレベルでの分類の対象自身も絞るといふ、そういう2段階でフォーカスをすべきであるという意見であります。

相磯 具体的にどこにありますか。

浦城 先程お話ししましたように、やっぱり目的にもいろいろウェイトがあるわけでありまして、今回第5世代の研究がアーキテクチャに非常にウェイトが置かれているということですから、核言語を選び、アーキテクチャのデザインをし、それによってソフトウェアの生産性の向上の可能性や知識ベースへの適用性を見つけ出し、実証することについては非常に重要であると考えていますので、それらに関連するものが非常にプライオリティが高いのではないかと思います。

それに対して、それをサポートするためのシステムなどはやるにこしたことはないわけですが、それはまたいろんな形で別のプロジェクトでも行なわれますし、従来の延長でも行なわれると思います。例えばローカル・ネットワークだとか、先程から出ておりますスーパーパーソナル・コンピュータといったようなものも、やるにこしたことはないわけですがプライオリティは下げてもいいのではないかと。CAD自身も、これもどんどん追求して参りますと、これだけでも大きなプロジェクトになりますし、最後には知識ベースにもつながる大きな研究テーマであることは認めています。最大の目標がそういった道具作りかというのと、やはり道具作りはセカンドレベルではないかという意見を持っております。

相磯 いかがでしょうか、他にご質問ございませんでしょうか、それでは一番後ろの方。

木戸出 東芝の木戸出です。

今までのアーキテクチャへのご質問は、何か本体そのものばかりだったと思いますけど、第5世代のコンピュータが素人にも使えるとかいうお話があるように、人間とのインタフェースに関する新しいアーキテクチャの考え方とか、そういったものはないのでしょうか。

データフロー・マシンにせよ、何か並列で作るわけですけど、それを人間が使うときには、どう見えて、どうとらえればいいのかという、その辺での新しい考え方へのアプローチはどういうふうになっているのでしょうか。

相磯 今までのご説明は、第5世代のアーキテクチャ、つまり中心のお話が多かったんですけど、使い易さを求めるという目的のためにどんなことをされるのでしょうか、という質問です、内田さんどうぞ。

内田 日本語でお答えします。

スケジュール全般の研究開発計画の中では、最初に良質の高品質のマンマシン・インタフェースを実現するための一番コアになるところを先に開発したいということを考えております。

それがパーソナル推論マシンのようなものとして最初は実現されるのだと思いますが、その上に今度は機械翻訳システムのようなもの、それから質問応答システムのようなもの、このようなものがソフトウェアのシステム、それからそれに附随するようなスモール・スケールのハードウェアシステムとして、いろいろ研究が重ねられて、前期はそのような基礎研究が行なわれて、中期くらいからそろそろハードウェア化の要求が出てくるのではないかとこのように考えておりました、それからハードウェアを作り始める。

一番最後のファイナル・ステージでは私のお見せしたスライドの中にインテリジェント・マンマシン・インターフェイス・マシンというインテリジェント・インターフェイス・マシンというのが挙げてありましたように、その時点では、そういう高品質の周辺機器というのを開発するというふうに考えております。

プランを作るのはどうもまだ、いま中間段階で、今年度の終りまでかかるわけでございまして、その時点では、その辺が討議されるというふうに考えております。

相磯 それでよろしいでしょうか。

そちらの方、どうぞ。

シンドラー ポールシンドラーです。

米国のコンピュータ・ニューズペーパーのレポーターです。

この2日間に渡って、いろいろ聞かせていただいたわけでございます。

このセッションに参加をし具体的な計画を聞き楽観的な予測のお話しがございました。

第5世代コンピュータの開発に向っての楽観的なお話しがございました。

レポートというのはそもそも楽観的なものでございますので、ずい分エキサイティングな話だなあと思ったわけでございます。それが午前中までの話です。

ところがここ2日間午後になりますと、インタビューをさせていただいたわけでございます。富士通さんの上の方のインタビューをいたしまして、きょうの午後は日立の上の方のインタビューをさせていただいたわけなんですけれども、こういうインタビューをした

方達はいずれも第5世代コンピュータ・プロジェクトを尊敬している素晴らしいメンバー構成である、私共の会社の人間も参加しておる、しかし10年間で成功するとは思えないと彼らは言っていたわけです。

いずれも立派な日本人の紳士でいらっしゃいましたけれども、私の直接の質問には答えていただけませんでした。

私、この会議の皆さんは楽観的すぎるのかどうかということをおつたんですけれども、返事してもらえませんでした。

ということは、恐らく私の推定では、AIの原則をここから吸収して、コマースのプロダクトを、1992年に作れる可能性については、疑問を持っておられるのではないかとこのように思ったわけなんですけれども、いかがでしょうか。

皆さんは自分達の計画が楽観的すぎると思っておられるでしょうか。

相磯 それに関しましては元岡先生何かございますか、私も一言と思っておりますが、どうぞ。

元岡 どうも、むずかしい質問になると私が答えなくてはいけないので、いつも困るんですけれども。

このプロジェクトでできたものが、すぐコマース・ベースにのるのであれば、それはその会社が自分でやればいいので、ナショナル・プロジェクトとして取り上げる必要はないと考えています。

従いまして、このプロジェクトでできたものをベースにして、それぞれの会社がさらに数年間苦勞してそれを商品にするようなレベルの問題を我々は考えているというのが1つのポイントであります。

それから知識情報処理というような分野はこれは非常に奥の深いものでありまして、勿論我々がこの10年間で知識情報処理のすべてを解決するなどは、我々としても勿論考えていません。唯、そういう分野が、それからあとに開けるための土台を作るのが我々の目的であります。

まだいろいろ言いたいことはありますが、このくらいにしておきます。

相磯 それでは私も何か言わないといけないのではないかと思いますので、申し上げます。

元岡先生がおっしゃった通りで、私はこういう情報処理の分野というのは、まだまだ20年30年は成長をし続けるだろうと思っています。

それから、これからの20年30年は今までの20年30年とは違ひ、もっともっとスピードの速い発展をするだろうと思っています。

従いまして、このプロジェクトはそのスタートポイントだと私は了解しています、ですからすぐに役に立つというような意味で期待をされると、勿論いくつかの成果は多分すぐにも役に立つだろうと私は思いますけれども、そういうような意味では、私は基礎研究だと思っています。

それからもう1つこれは別の観点からコメントを申し上げますと大体やはりメーカはコンサバティブです、これは当然でありまして、またコンサバティブでなければ多分いけないだと思います。過去に通産省がリードしたいくつかの先進的なプロジェクトがありますが、その過程でも大変コンサバティブでした。

しかし現在ではメーカも多分感謝をしていると思います。いや、かなりです。

ですから、そういうような意味でも、このプロジェクトが、どういう形で終るかは、わかりませんが、将来に引継ぐ土台を作っていけばいいのではないかと、私はそう思っています。

他にいかがでしょうか。

フロア外人 日本の政府の方にAIにつきまして、1つ伺わせていただいてもよろしいでしょうか。

ケンブリッジ大学のライトヒル先生が、AIについてのレポートをまとめられたことがあったと思います。

このレポートについては、皆さんもよくご存知だと思いますが、日本の政府の関係者の方がライトヒルのレポートについてはご存知でしょうか。

マコーミック 私、個人的に関係があるので申し上げます。

私、このライトヒルさんという方をよく知っておりますけれども、この人はAIについて、よく知っている人ではありませんので、このライトヒル・レポートが出たために英国はAIに関しては、もう仕事ができなくなってしまっているわけでありまして。

エジンバラ大学のミッキー・バーネ先生は、とうとうイリノイ大学へ逃れていってしまったわけでありまして。

ですから、その結果、ロボティクスに関する文献の中で、かなりの批判が出ているわけでありまして。

というのは、このレポートが出たためにロボット技術では進んでいた英国をこの分野から追い出してしまったことになるわけです。

マーガレット・サッチャーさんは、あるいはこの10年間の間にどれ程変わっていたかということ、よく知っておられるべきであったのではないかと思います。

フロア外人 あのレポートは、それでも重要だと思っています。

私は残念ながら、マコーミック先生とは意見を異にします。私は、それでもなおかつ日本の政府の関係者の方に、これを是非読んでいただきたいと思います。

相磯 どうもありがとうございました。他のコメントを。

カイファン 私、大変オープンに勇気をもって、日本の学会の関係者の方、あるいは通産省の関係者の方が、こういうような会議を開かれたということについては、高く評価をするものでございます。

こういうナショナル・プロジェクト研究開発計画というものを国際的なフォーラムの中で紹介されたということは、評価したいと思います。

そこで参加者の方から、もう少し伺いたいと思うんです。アメリカのメーカの方ヨーロッパ、あるいは日本のメーカの方に特にご意見を伺いたいと思うんですけれども。

今までのところ、メーカからの意見があまり出なかったように思いますので、それをもっと聞きたいと思っております。

それから内田先生に1つ伺いたいと思います、トレリーバンさんからお答えいただいてもいいと思うんですけれども。

私の質問は、データフロー・マシンに関するものです、イレギュラ・パラレルリズムがシンボル・マニピュレーションのアルゴリズムに関係があり、これが知識ベースの処理には関係があると思います。

しかし、サイエンティフィックな大規模なコンピューテーション、例えば、マトリックスのようなアレイタイプの演算の場合には、インテリジェントなインタフェース・マシン、あるいは推論マシンのインプリメンテーションが必要だと思います。

音声分析あるいは音声合成やパターン認識の分野においては、不可決であろうと思います。

過去におきましては、こういうようなマシンをシンクロナスなパイプラインのマシン、あるいはシンクロナスなアレイのマシンでやって参りました。

こういうような大規模な処理をシンクロナスなマシンでやってきたところがあると思うんですけれども、私の質問は、シンクロナスなコンピューテーションというものが、エイシンクロナスなデータドリブンのタイプのマシンで効率的に処理できるかというもののなのです。

この点については真剣に検討されたんでしょうか、

そして今までのところの結論はどうかを、お聞かせいただきたいと思います。

内田 先生はシンクロナスな計算がエイシンクロナスなマシンでできるかどうかということをお聞かれるのですか。

シンクロナス・マシンとエイシンクロナス・マシンとの区別は高速のコンピューテーション・モデルを考えたときには重要であると思います。

もし私がデータフロー・マシンのようなエイシンクロナスなマシンを作る場合にはシンクロナス・メカニズムを、マシンの中はかなり入れるだろうと思います。

ですから、ちょっとご質問いただいて、戸惑っているんです、というのは先生はエイシンクロナスなマシンをトップレベルから作ることを言われたわけがありますね。プログラミング言語から、And, Orとかいう、そういうゲイトエレメントのようなボトムレベルまでエイシンクロナスなものを作るというようなことは、私はしないだろうと思います。

シンクロナスなメカニズムをエイシンクロナスなマシンの中はかなり入れるだろうと思います。

カイファン 私、計算そのものに言及しているわけです。

かなりの大がかりなマトリクスタイプの演算の場合には、スペシャル・パーパスのレイ・マシンを作ってメモリー・ストラクチャがマッチしていれば、そしてリソースがあれば、かなり高速にできると思うんです。

しかし、ここでは、エイシンクロナスなモデルを作ってシンクロナスと言わなくとも、シンクロナイズしたコンピューテーションをやろうとしておられるように思うんですけれども、それが効率がいいかどうかというのは、オープン・クエッションだと思うんですけれども。

内田 そうですね、それはその通りだと思います。CRAY-1あるいはF.F.Tプロセッサといったようなスーパーコンピュータは完全にシンクロナスなマシンであります。どのレベルをとってもシンクロナスなマシンだと思います。

しかし、先生のご質問は、エイシンクロナスなマシンが、そういうようなプロセッシング、いわゆるパイプ・ラインとかF.F.Tのような処理を、そういうコンピュータと同じような効率でできるかどうかということをお聞かされているんだと思いますが、答えはノーだと思います。

イレギュラーなコンピューテーションをやりますと、

なんらかのオーバー・ヘッドが出てくるのは当たり前ではないでしょうか。マッピングがありますから、つまり仮想化をやらなくてははいけませんから、どうしてもオーバー・ヘッドは出てくるわけであります。

コンピューティショナル・モデルから実際のハードウェアシステムへの仮想化をやらなければいけませんから。

そういうオーバーヘッドは、かなりの研究者が、データフローマシンにつきまして研究をしているのではないのでしょうか。

最近の結果を見ておきますと、現在のCRAY-1なんかを見ますとネガティブだと思いますけれども、将来はこの問題は、先程説明いたしましたような理由で、イレギュラーになってくると思いますので、このようなジョブはCRAY-1なんかでは、とても処理できなくなると思います。

相磯 ご質問があるかと思いますが、もう2時間以上たってしまいました。

いろいろのご質問、ご意見、様々な角度からご指摘いただきました。

我々もこれを充分考慮しまして、これから研究を進めていきたいと思っています。

一つだけ言えることは、この第5世代の研究の価値が、大いにあるということは、一致した意見であったように思うんです。

どうぞ皆さんのアドバイス、それからコラボレーションを期待しておりますので、よろしくお願ひしたいと思います。

どうも、きょうはご出席のパネリストの方々、それからご質疑、ご討論いただきました方々ありがとうございました。

IV.3 パネル・ディスカッション Ⅲ

第5世代コンピュータ・システムの波及効果



座 長：東京大学 元 岡 達
パネリスト：米国標準局(NBS) フィリップ・J.リガナッチ 通商産業省 岡松 壮三郎
INRIA J-L リオン 日本電信電話公社 戸田 巖
英国インターバンク・リサーチ チャールス・N.リード
西独数学情報協会(GMD) ノーバート・ツァイベルスキー

元岡 それでは、最後のセッションを始めさせていただきます。と思います。

最初に、パネリストの方を紹介させていただきたいと思います。一番右側から、C.リードさん、リードさんは、現在IBROのディレクタをしていらっしゃいます。IBROは、英国の主要な銀行が共通して持っている研究組織です。

リードさんは、社会科学、経済学といった分野のバックグラウンドをお持ちで、情報技術をマネジメント・サイエンスや実際の応用に使うことを研究してこられた方です。

また、政府のいろいろな委員会に関係しておられまして、英国の首相であるサッチャー婦人のインフォメーション・テクノロジーに関するアドバイザーもしておいでです。

次に、フランスのリオンさんをご紹介します。リオン先生は、パリの科学アカデミーのメンバーであられるとともに、カレッジドフランスの教授、INRIAの所長、インターナショナル・マセマティカル・ユニオンの事務局長をしていらっしゃいます。

次は、米国のリガナッチ博士です。リガナッチ博士は、米国標準局(NBS)のコンピュータ・サイエンスとテクノロジーのインスティテュート(ICST)

のディビジョン・チーフをしていらっしゃいます。

このICSTというのは、政府関係のいろいろなコンピュータの研究開発といったものについて責任を持つ部門であります。

いろいろな経歴をお踏みでありまして、ロックウェル・インターナショナルあるいはGE、IBMといったところでも働いてこられ、大学でグラデュエイト・コースの講義もなさっておられます。

次は、ドイツのツァイベルスキーさんで、ツァイベルスキーさんは、現在、数学と、データ処理の権威者でありますGMDの所長をしていらっしゃいます。

経済学、あるいはビジネス・アドミニストレーションの専門で、そういった分野へのコンピュータのアプリケーションをやっておられます。それから、コロン大学のブランニシグ理論の教授もしておいでです。

その次が、日本の戸田博士です。戸田さんは、日本電信電話会社の横須賀電気通信研究所のデータ通信研究部長を現在しておられます。

大学を卒業になってから、ずっとコンピュータ関係の仕事をなさっておりまして、現在、電電公社が中心として開発しているDCNAと言われる、ネットワーク・アーキテクチャの責任者であります。

最後は、岡松さんです。通商産業省の機械情報産業

局電子政策課長であります。このプロジェクトの推進をやってくださる、私からみると、スポンサに当たる方です。

きょうの題は「Impacts of Fifth Generation Computer Systems」ということですが、その問題と、最後のセクションでもありますので、このプロジェクト全般について、幅広い議論をしていただきたいと思います。それでは最初に、リードさんお願い致します。

リード ご紹介有難うございました。まず初めに、私、これから申し上げるのは個人的な意見であって、英国の私の同僚並びにサッチャー首相の意見の代弁者として述べるのではないということをお願いしておきたいと思ひます。



私は、このテクノロジーのユーザとしてお話をするので、皆様方と違って私、エンジニアではありません。

どんなテクノロジーの開発でも、よい面もありますし、その開発の発展の中には悪い面も必ず含まれています。

火薬は、大変役に立ちますが、人を殺してしまうという影響があります。自動車も、都心の中ではいい面もありますけれども、大事な資源を使っており、また、公害をもたらすということがあります。

ですから、こういうことを考えて、私どもがこのような政策を決定していくうえで、このような良い面、悪い面が出てくるのだということを考えていかなければいけないと思ひます。

情報のテクノロジーというものも、例外ではありません。今迄のところ、このテクノロジーというのは、その与えた影響は物理的な環境に対してのものでした。けれども、現在、精神的な環境にも与えるような影響が生まれてきています。そして、大きな、革新的な変化が社会に対して行われようとしていますし、また、個々人の生活の質に対して行われようとしています。

テクノロジーが、現在の開発の現状であっても、やはり、既にいろいろな問題を含んでおります。そこで、私が一番重要だと思ふようないくつかの問題をまとめてみたいと思ひます。

情報を官僚組織とか政府のような形で、あるいは、各企業が使うということは、単に、各個人のプライバシーを犯すということだけではなく、自由をも犯します。

新しい法律が、その問題を扱うように多くの国で制

定されようとしています。そして、私、ここでちょっと申し上げたいのは、こうした法律の性質というのは、どのようにそれがなされなければならないかというテクノロジーに関するものではありません。

そうではなくて、その法律は、何がなされるかという、誰がそれをしていいのか、どういう目的であるのかという、それをコントロールしようとしているわけです。

そして、個人個人の情報に対して、だれが何をしているかということを知ることに関して、市民の権利を認めます。

第2に、情報技術というのは、非常に効率的なコントロールをする能力を持っています。特に、例えば、警察とかその他の研究者、銀行、そういったものを監視できるような力を持っていると思ひます。非常に効率がよくなってきますと、これは危険なことにもなると思ひます。

私、個人的に申しますと、官僚の政府の非効率性というのがありますけれども、これは逆に、私が思ふ中で一番効率よく自由を守るものであると思ひます。

情報技術は、今迄の、伝統的な状況のあり方を象えて、ノウハウをだれにでも提供するというふうになってきました。そして、いろいろな形の、仲介機能を破壊する、つまり、売手と買手との間に直接取引が成り立つような形で、これまで仲介者として存在してきたものの役割をなくしてしまっています。

また、多くの機能を果たすべき労働力というのが、この情報科学によって大きく変わってきて、量においても、また構造においても必要性が変わってきました。

多くの国で、失業という問題が大きく取り上げられています。そして、誰もが、それに対して満足のできる解答を与えることができません。

農業における労働力というのは、テクニカルな発明の影響で、全部工業の方に行ってしまう、農業で労働力が不足しております。工業の方では、非常に効率がよくなってきております。

サービス業においても情報技術が広がろうとしております。ですから、私たちはいま現在、サービス業においても労働力にとって変わられようとしています。そうすると、労働力はもう行くところがないわけです。

そうすると、その結論はどうなるでしょうか。このような危機をうまく乗り越えることで問題は解決できるのでしょうか。多分私たちは麻痺状態になってしまうのではないかと思ひます。

この分野で、いろいろな専門家がやっていますけれども、それほど能力のない人はそれに関係がないということになってきています。

例えば、週に3日しか働かないでいいということになってしまうのではないのでしょうか、あるいは、3日か4日しか働かないような、フィジカルな面ではどうなのでしょう、そういうことは…。日本ではどうなのかと私、疑問に思うんですけども…。

情報技術というのは、そのほかでも、権力の構造にも影響を与えています。

銀行の自動化というのは、業界の力、あるいは労働組合の力ということが出てきて、アメリカでは銀行のストライキが出てきています。これまでは、そういうツールがなかったので、ストライキがなかったわけです。

そういう、過去のことから、いろいろな危険が起こってきていると思います。こういうことがテクノロジーによって指導され、エンジニアの方では、非常に熱狂的に研究していますけれども、賢明ではない面もあるのではないかと思います。

エンジニアに対して、どういう使用法をしていったらよいのか、何を望んでいないのかということをおっしゃっていただく必要があると思います。

ファイゲンバウム先生がおっしゃったように、可能性をみるのではなくて、必要性をみなければいけないのだということをおっしゃいました。それに非常に賛成しています。

私の意見は、エコノミストとか市場の力を全く無視するものではありませんが、例えば、コストだけを低減すればいいとか、生産性を向上すればいいという、それだけが究極の目的ではないと思います。

それ自体だけを求めますと、大きな害がもたらされ得ると思います。人間の生活の質、人生の質、といったようなことも考えに入れていかなければならないと思います。

こういうことは、正札は付いていないわけですが、経済学者とか、そういう人々はこういうものを考えに入れていません。そのことが、これまで何十年かの間に大きな害を社会に与えてきたように思います。

生産性が高いということ、コストを低減するということは、望ましくない影響をいろいろ与えてきました。そういう害がよりよいテクノロジーによって与えられてきたと思います。

例えば、セルフサービスの商店とか銀行へ車で行く

というようなことになってきますと、だれかに話しかけることすらいらなくなるということになってきます。

情報の分配、テレビとかラジオといった情報の分配によって、それがサポートされているから、具体的な人間に話しかけるということが阻害されています。

その結果として、人々が孤立してしまう、お互いに分離されてしまう、そして、このことこそ、多くの人々に関する限り、最も望ましくない結果だと思えます。

そうすると、もっと大きいコストを払っても、その方が望ましいのではないか、ほかに選択肢があるのかというそれはなかなかないと思います。

私は、このような望ましくない、賢明ではない経済的な見方というのは、第5世代のコンピュータ・システムでは、社会的な反対には合わないと思います。それと反対の見方がとられていかなければならないと思います。

第5世代のコンピュータ・システムが、役に立つようになるためには、人間に対して、非数値な情報をうまく処理していかなければならないのではないのでしょうか。

これがどのようになさなければいけないかということが問題です。そして定性的というか、質的な判断を意思決定においてやっていて、それに対して、現在数値的なデータばかりに価値が注がれている、それを変えていかなければならないのではないのでしょうか。

経済的な原則だけを重視していくということではなくて、そういうようなことをやっていくことによってのみ、第5世代のコンピュータは正当なものとなり得ると思います。

ひとつの知的な発展というのは、非常に大きな社会に影響を与えることができますけれども、それは、意思決定の質を変えていく、特に、政府によって行われる意思決定の質を変えていくということだけだと思います。

もうひとつ、第3番目として述べたいのは、これまでほかの講演者の方々が述べてこられたことですが、情報のテクノロジーの、人間の組織への適用から得られる最もよいものは、そこで課された原理によって情報を組織化して、目的を立て、手続きをはっきりさせるということ、多くの場合、90%の利益というのが、実際にコンピュータを買うことによってではなく、あるいは一行のコードを書くことによってではなくて得られると思います。

このことが、同じように、非数値のデータにも、また、それに関する、知的な概念にも当てはまることだ

と思います。その方が、より大きな利益をあげられるのではないのでしょうか。

世界中の人々に対して、第5世代のマシンが現実に行えるものであるということ、だれもが非数値的なデータをうまくコントロールできるということになると、素晴らしいのではないのでしょうか。さもないと、通産省が予算を出されるというようなこともなかったと思います。

第5世代のコンピュータは、実現されますと、すばらしい可能性を秘めていると同時に、悪への可能性も秘めていると思います。したがって、いま、この時点から、エンジニアリングの研究と併行して、いわば、ソーシャル・エンジニアリングの研究も行われるべきではないかと思っています。

社会的な目的というものを考えていかなければならないと思います。だれでも否定できないような一般的な言い方ではなく、具体的な、社会的な目的というものを掲げていかなければならないと思います。

社会的なニーズをみて考えていかなければならないと思います。我々は一体どういうことをやりたいのかということに照らして考えなければならぬと思います。

新しい法律は、テクノロジーが最も便益的な方法で使われるように書かれていかなければなりませんし、また、電気通信をうながすような形でつくっていかねばならないと思います。

既に指摘がありましたように、情報がエキスパート・システムで使われるためには、それはもうツールがあり、できるものがもうここに揃っていると思います。

しかし、使いこなすためには、管理者、医師、あるいは法律家といった、さらに、政府の役人を含む、すべての人々の教育を考え直し、新しいツールを使いこなすための教育というものを、いまからやらなければならないと思います。

もうひとつの、調査と申しますか研究のコンセプトを申し上げたいと思います。

ここで、インエキスパート（非専門家）システムという考えを導入したいと思います。世界の人口の約半分は、平均以下の能力しかありません。数学者や経済学者でも、このところがわかっていません。

多くの国々においては、日本とは情勢が違って、平均的な国民の能力というものはそれほど高くはありません。簡単な仕事でなければ、能力に限りのある人々はやりこなすことはできないわけです。

複雑なシステムの中で機能していくためには、共同

の努力が必要であり、そのための社会的な環境が必要なのわけです。ハードウェアが安くパワフルなものになってきているわけですから、ジョブを人間に合ったようなものにしていかなければならないと思います。

この方が、第3世界のニーズをより満たすものになるのではないかと思います。中東で一度これをやったことがあります。そこで得られる労働力の質は低く、また、ソフトウェア、ハードウェアが非常に高かったわけです。

現在のテクノロジー、特に第5世代のテクノロジーというのは、インエキスパートシステムのための将来を築くものだと思います。それは非常にバラ色の将来だと思います。

ほかのテクノロジーにも目を向けるかもしれませんが、例えば、医学的な健康の面、それから、下水とか水の供給、高いレベルの外科手術、そういったものと同じように、これが必要だと思います。

このようなことが、今週のこの会議の中でもっと責任を持った、適切なテクノロジーの使い方というのをもう少しお聞きしたかったように思います。

最後に、エンジニアリングのみを中心にしたシステムだけを追い求めると、どれだけ望ましいシステムができるかということについては私は予言することができません。

ノンエンジニアリングのタイプの研究が併行して進められ、それが取り入れられていかなければならないと思います。そうしますと、私は、このプロジェクトを本当に心からサポートすることができると思います。有難うございました。

元岡 どうも有難うございました。次にリヨン先生をお願いします。

リヨン ここでは私は個人的な立場でお話を申し上げまして、別に、正式な見解を設けるわけではありません。まず最初に、この素晴らしい会議を組織してくださった方々に対して、心から感謝をしたいと思います。

ここで、私は非常に奇妙な立場におかれておりまして、私の仕事は、ひとつは数値解析です。ここでは非数値解析という話がされておりますので一。

私は、インタナショナル・マセマティカル・ユニオンの事務局長をしております。あまり数学を使わない方がいいというような批判も聞かれておりますけれども、私は、ここに出されている考え、そして第5世代の目標に賛成するものであります。

このディスカッションのトピックは、コンピュータのインパクトであります。ここで私は、6つの点に

ついて申し上げたいと思います。

まず第1は、特に科学という面でのインパクトであります。このことはあまり強調されていなかったようです。つまり、ほかの科学分野へのインパクトという意味ですが…。

このプロジェクトが成功するとした場合、勿論、成功すれば私は非常にうれしいわけですが、成功したあかつきには、非常に重要な、根本的なインパクトをほかの科学に与えるであります。



例えば、生物学と科学とか、こういった面において、このプロジェクトは非常に大きなインパクトを根本的に与えるものであります。

2番目の点ですが、専門家をどのようにしてみつけるかということです。先程、どう

やって専門家をみつけたらいいかという話が今迄出て参りました。

ここでひとつ、メンバーについて申し上げたいのですが、このように、非常に野心的なプロジェクトをやる場合は、これは非常に魅力のあるものですから、そのプロジェクト自身が魅力的であるために、若い人たちは、こういったプロジェクトはやりたいというふうに考えると思います。それによって、専門家をみつける問題はなくなると思います。

つまり、若い人というのは、大型プロジェクトに魅力を感じるものですから、これは、ある程度、危険もあるかもしれませんが、若い人たちにとって、このプロジェクトは非常に興味深いものであると思います。勿論、歳とった人にも非常に魅力のあるものですが…。

3番目の点ですが、訓練と教育の問題であります。日本で、このことについて研究されたがどうかについて伺いたいと思います。

それは、このような重要なプロジェクトは、もし成功すれば、一部成功したとしても、非常に重要なインパクトを教育に与えると思います。

つまり、現在10歳である子供は、このプロジェクトが終るころには20歳になっています。ですから、いまから、こういうことを考えなければいけないと思います。つまり、こういった、子供の教育ということに関して、何か日本で研究がなされておりますでしょうか。

そして、4番目の点であります。私は、ひとつ批判的な意見を持っております。ユーザの考えが十分に

反映されていないように思います。私は、このプロジェクトのすべてをはっきりわかっているわけではなく、勿論、いろいろなコンサルティングもユーザとの間の話し合いで行われているかもしれませんが、それが少し足りないのではないかと気が致しました。

5番目の点であります。これは国際的な立場に立った見解ですが、先程国際問題も出ていましたが、最初の問題では、科学的なインパクトというのは、国内ではなくて国際的なインパクトになると思います。

ですから、このプロジェクトというのは、国際的な協力の中で行われますので、このプロジェクトのインパクトが国際的なものになるであろうと思います。国際的なプロジェクトであるために、若い人たちに魅力的であるということも言えると思います。

それからまた、同様に、基本的、根本的な面においても、国際協力でできると思います。

6番目、これは最後の点でありますけれども、これは問題でもあります。更にどのぐらい協力できるかというのは、日本人の方によっているわけです。

つまり、どのようにしてある国と日本が協力していくかということが、はっきりしていないと思います。日本のプログラムで結果の出たものを、果たして私どもが入手できるのかという問題があると思います。

例えば、初期の段階、実用化に入った場合、それぞれに、どの程度の協力が得られるのかという問題があります。このことは、ある段階で考えなければいけないと思います。以上です。有難うございました。

元岡 リガナッチ博士にお願いします。

リガナッチ 有難うございます。私も、先程来お話がありましたように、この会議の主催者の方に対して、このように、貴重な機会を与えていただきましたことをお礼申し上げたいと思います。

このような大きなプロジェクトの計画段階において、コメントを申し上げる機会に恵まれたことを大変有難く思っております。同じような協力が、今後続けられますことを大いに期待するものでございます。

また、この会議の期間中に、福井先生がノーベル化学賞を受けられたというニュースを耳に致しました。私、外国からの参加者を代表致しまして、福井博士と福井博士を生み出された日本に対して、心からの祝福の気持をお伝え申し上げたいと思います。

私の意見は、3つに分けて申し上げたいと思っております。第5世代コンピュータは、いわばグランド・デザイン的なものであろうと思います。テーマ、また、

動機づけも、社会の情報化に従って動いていくものだと思いますので、アマチュアとしての哲学的な意見を申し上げたいと思っています。



その次に、社会的インパクトについて申し上げ、かつ、米国と日本との比較を試みてみたいと思います。

3番目に、今後の第5世代コンピュータの動きについて、いくつかの提案をさせていただきたいと思っています。

哲学的な意見ですが、まず最初に、多くの方のご存知の定理について申し上げたいと思います。これは、1930年代に出てきたゲーデルの証明ですが、これでは、どのような公理の体系においても、初等代数よりは複雑なものでありますと、証明できるかできないかのいずれかです。

つまり、真か偽かのどちらかでもあるということです。

例えば、“I am a liar.”(私は嘘つきである)というのは、真か偽か、あるいは両方かもしれません。

推論や論理的思考過程では、この命題を解決することはできないかもしれません。プロノウスキーは、素晴らしい記事を1960年代の半ばに書いて、このコンセプトをLogic of the Mind という記事の中にまとめています。

第5世代コンピュータのグランドデザインの意味を考える場合には、このプロノウスキー氏がLogic of the Mind の中で語っている内容を加味する必要があります。

次に、アナロジーとシンメトリーです。このアナロジーとシンメトリーというのは、人間の推論の中で大変役に立つ手続です。

この、非常に複雑な、知識ベースの問題を考える場合に、これを活用することができようと思います。

現在、数多くのコンピュータ・ネットワークが作られており、これは、人間の頭脳の中に入っている要素の数を考えてみますと、膨大なものだと言えないかと思っています。

マコーミック先生が言われたように、生物学的に考えますと、左側の脳と右側の脳との間に、推論メカニズムというものがあり、これは、右側の脳と左側の脳とよく似ているものがあるというお話がありました。ここに驚いた時の反射作用がわかるという気がするわけです。

アーキテクチャ上のシンメトリーを考えることによって、自然が見つけたシンメトリーを活用することができるのではないかと思います。

SF小説、ハイナインの『ムーン イズ ハーシュミストレス』という小説の中に、コンピュータが月の表面下を占拠して、そこで初めて自分が何かかわかるといったようなことが書かれていたことを思い出しました。

修辭学的な質問として、マコーミック先生やほかの先生方に申し上げたいことは、コンピュータ用のセンサーをつくって、それが、バイオロジカルなセンサーに似たようなものをつくることができれば、まあ、それは第7世代ぐらいになるのかもしれませんが、そこに初めて本当のコンピュータ・ビジョンが出てくるのではないかと思います。

いわゆる、生物学的な世界で見るところの視界と同じようなものが、コンピュータの中でも実現できるようなものが出てくるのではないかと、いうふうに私は思ったわけです。

更に続けて哲学的な観察を申し上げますと、現在、3,500種類ぐらいの言語があります。

我々、コンピュータ・サイエンスの研究者は、これぐらいの人工言語をつくっていかなければいけないということになるのではないでしょう。

と申し上げますと、400年前にモンタギューは大変面白いことを言ったわけです。

それから、哲学的なコメントの最後として、曖昧さの性質ということを上申し上げたいと思います。そして、この曖昧さの性質というものが、第5世代コンピュータ・システムを考える場合に重要な要素だと思います。

情報というのは、集合になった要素の特性であります。散文、詩、音楽というような順序にだんだん曖昧性が増えていくわけです。

バーンスタインは、ウイザー・ミュージックという話を4、5年前にしましたときに、このコンセプトを音楽というスペクトルから話しました。

いま我々は、音楽をルールによって合成できるようなコンピュータをつくることができます。第5世代コンピュータのコンセプトは、いわば、創造的な音楽まで創作できるようなものをつくっていこうとしていると言えるのではないかと思います。

いわゆる、『Seven Types of Ambiguity』というクラシックな本の中にも見られるようなものをつくろうとしていると言えるのではないかと思います。

さてここで、社会的なインパクトについて申し上げ

げたいと思います。日米の比較を試みてみたいと思います。

社会的な価値への影響というものは、過少評価してはならないと思います。第5世代以降のコンピュータ・システムというものは、世界的に影響を与えていくものであらうと思います。

今回のこの会議がひとつの例であらうと思います。本会議においては、全体のグループを対象にした会議が開かれているわけですが、米国においても、同じような関心が生まれてきています。

科学、あるいは技術の専門家が、20年先を展望する、情報化社会に関するレポートを発表しています。しかし、このレポートは、米国では日本と違った使われ方をします。

詳しい内容は申し上げませんが、NBSの前の長官であり、現在IBMのチーフサイエンティストをやっている人は今年80歳を迎えますが、米国のボストン市、ここでは、日本の弁護士の数と同じ人数の弁護士がいるところですが、ここでふたつの社会を比較されました。

したがって、日米社会は、弁護士の数においても違うように、第5世代コンピュータ・システムの型においても違うでしょう。

では次に、人口の分布状況の違いについても申し上げたいと思います。

米国ではフリーウェーに従って、そして、日本や東京ではあるひとつのパターンというものはないと思いますけれども、過去において、人間は、交通の主要路に沿ったところに集まって居住していました。

パーソナル・コンピュータというものが生まれ、分散型のネットワークというものがもし実現されたならば、家内工業というものは、世界的に新たに生まれ直さなければならないと思います。そして、いわゆる、新たな家内工業が生まれてくるのではないかと思います。

つまり、第5世代コンピュータが生まれ、分散ネットワークができることによって、家内工業が再び台頭してくる可能性があるのではないかと思います。

もう少し具体的に申しますと、米国では、プライバシーの問題がこここのところ問題になってきています。

法律も通っておりますし、また、規則も出されております。フランスでも問題になっていると聞いておりますし、ヨーロッパでも問題になっており、日本でも次第に問題になってきていると聞いています。

情報へのアクセスとは、これはパワーにつながるものです。いわゆる、コンピュータが使えらるようになって

情報へのアクセスが進むこととなります。第5世代コンピュータが実現されることによって、予想もしていなかったような形で社会の変化が生まれてくるであらうと思います。

国と国との間の情報交換というものが、これから先、第5世代コンピュータが直面する問題ではないかと思えます。かなりの成功を第5世代コンピュータが治めた場合に直面する問題ではないかと思えます。

もうひとつ、人口の特色というものをあげておきたいと思えます。米国における'70年代の人気のあった本は、スタッタクルの『Working』という本です。これは、様々な職業についている人にインタビューをしてみとめられた本であり、それぞれの人が、どんなふうにものを考えていたかということをもとめたものであります。

この本は、アメリカの人にとって、新しい情報を提供したものであります。鉄鋼業に働いている人がどういうことをしたか、タクシートの運転手がどういうことを考えているかを知るための、新しい本でした。

同じような本が日本にあるのかどうかは存じませんが、ここでひとつ質問をさせていただきたいと思えます。

第5世代コンピュータのプランナーの方々は、社会からのボトムアップの知識というものが変わってきているということも考えておられるのでしょうか。

さて次に、第5世代コンピュータへの具体的な提言をさせていただきたいと思えます。

International Standard Organization (ISO) がここ2、3年間にわたって、オープン・システムズ・インターコネクション (OSI) を考えております。

これは、機能的な階層でありまして、7層から成っています。物理層から始めて、現在のプロトコルから申しますと、ネットワーク・プロトコルあるいはアプリケーション・プロトコルまでいくものを考えております。

オープン・システム概念は、どのようなシステムでも、ほかのシステムに通信できるようなものを考えています。

したがって、通信プロトコルをつくって、その上にファイル転送プロトコルとか仮想端末プロトコルとかグラフィックとか、そういったものを乗せようというものです。つまり、特別なアプリケーションとして考えているものを乗せようというものです。

ですから、それらに共通なものをつくって、アプリ

セッションのところまで積んでいこうというもので、そして、トランスポートやセッションのレイヤーについては、共通のものをつくっていこうという考え方で

す。このコンセプトは電気通信の立場からつくられたものであります。このモデルを、データの蓄積や検索、プロセス、トランザクション、資源管理、情報の表現、システム管理の点でも考えていってはどうかという提言が出てきています。

私は、こういうものが重要な点だと思いますので、第5世代コンピュータのスタッフの方々は、この国際的な努力とも連絡をとられ、第5世代コンピュータのプロジェクトの中の一環として考えられることを提言致します。

ふたつ目のポイントは、マンマシン・インタフェースです。特に、画像それから音声入出力です。画像処理あるいは音声処理の入出力に関係している人たちは、やはりデータベースが必要だということを認識しております。

世界中の研究者が、アルゴリズムや手続きを詳しくお互いに研究していこうと思えば、データベースが必要です。かなりのものがあります。

例えば日本では、PIPSのプロジェクトでデータベースがつくられていて、幾分かそういう性格を帯びております。日本の言語、あるいはそういうようなものがつくられていますけれども、まだ十分ではないと思います。

したがって、第5世代コンピュータのスタッフは、フォーマット、データベースなど、マンマシン入出力アルゴリズムに使われるようなものの、国際的に同意できるようなものをつくるような方向で努力していかれるべきであると思います。

次の点は、様々な観点から話し合われた点であり、私の同僚であるキルシュ先生からもお話のあったことですが、このプロジェクトはいまは少し考え方が広すぎるという意見も出ましたし、狭すぎるという意見もありました。しかしこれは、リソースとの関連で考えなければならないと思います。

神経学、生理学、認知科学、あるいは言語学といったような研究も含めてということなのでしょうか、リソースはどこに焦点を合わせるのでしょうか。

第5世代コンピュータのプランナーたちは、エンジニアリング的な判断と、科学的な興味との間のトレードオフを慎重に考えなければならないと思います。

このような形で考えようということは大変素晴らしい

ものだと思います。我々は、成功例よりも失敗例から多くのことを学ぶことができました。

第5世代コンピュータはうまく設計されていけば、かなりの成果を収めることができると思いますけれども、失敗を重ねていくことによっても、また多くのことを学ぶことができると思います。

第5世代コンピュータが、今回の会議の質疑応答の間で、第5世代コンピュータのプランナーの方がどういうふうに考えられているかということがかなり明らかになってきたと思いますけれども、それを、何らかの形で文章の形ではっきりしておきますと、非常によいのではないかと思います。

最後の提言として申し上げたいのは、これは坂村先生のご意見と全く同じでありまして、ハードウェアとソフトウェアを早い段階で分けずに、ユニフォームなスタンダードなインタフェースをつくっていくことを考えるべきだろうと思います。このステートメントはまさにそのとおりだと思いますので、ここで再度繰り返しました。どうも有難うございました。

元岡 どうも有難うございました。次に、N. ツァイベルスキーさんお願いします。

ツァイベルスキー ご紹介有難うございました。私は、日本の友人方と一緒に座っていますが、私の喋る内容がそれによって影響を受けるというわけはありません。

私、ここでまだちょっと疑問に思っているのは、どういう役割を私たちが果たすべきかということで、目的に関してより詳しい情報を得て…。



もうひとつ思いますのは、このチャレンジを追求して、第5世代コンピュータのアイデアに何らかの答を出していくということですが、私は、そういったことについてこれから個人的な印象をお話する、それから、私に与えられた役割ということから、出来るだけお答えしていこうと思います。

まず導入部分として、第5世代のコンピュータ・システムという名前は、マーケティング・アイデアとして非常にいいものだと思います。

というのは、そうすると、第3世代のコンピュータという名前からそれほど離れていないし、また、このことによって、第5世代のものは'90年代まではそういうものは来ないということが明らかであります。

そこで私が、その間でわかったことは、このプロジェクトは実際のところ、プログラムとプロジェクトの混合したものであるということです。私は、テクノロジー的な変化のフィロソフィをこの情報処理の分野でどういうものであるのかということを見てきました。

ご発表には非常に感銘を受け、これまでのアウトラインや仮設やデザインに非常に感銘を受けました。

それは、プロジェクトをこれまでいろいろやられてきた方法があるわけですが、その方法で成功するよなということ、楽観的な見方をどうしてもとめるのです。

また一方、私は、日本の情報産業というこの業界が、これから10年間も、このような、チャレンジングな製品システムが出てくるのを安穩に待つてはいないと思います。

このような野心的な目標、プロジェクトということを知り、このようなプロジェクトは、つまり、これまで工業的に発展してきた発展を示すものだと思うのです。これが、興味深いツールとして、マーケティング・アプローチやそういうアイデアを与えてくれるものだと思います。

業界における研究開発における、そういったものを示してくれるものだと思います。ですから、第5世代コンピュータは、ただ大きな試みであるということだけではなくて、非常に大きなビジネスのツールだと思います。

国際的な協調についての話が先程出ましたが、第2段階目の段階、第3段階目の段階がどういう関係にあるのかということがありましたが、そういうところでも同心円のようにして、この国際協力をうまく取り扱っていくのかということが大事だと思います。

国際的な研究、という面での協力がまさに実行されなければいけないと思います。研究所、大学、民間の会社等すべてと、国際的な協力を進めていくべきだと思います。

ここでもう少し基本的な私の考えを述べたいのです。どうい影響を与えるかという、このプロジェクトの影響に関したものです。そして、どのような期待、どのような価値、基準が与えられているのかということです。

このプログラム自体は、テクノロジー的なインパクトのアセスメントは易しいものではありませんから、いくつかの側面だけでとらえて話したいと思います。

皆様方のプログラムというのは、4つの、非常に興味深いエリアに分かれると思います。

ひとつはVLSI(CAD)、第2番目の分野はコンピュータ・アーキテクチャ、これが、プログラミング・ワールドのある特定の条件に関連したものです。

第3番目が、知識ベースの考え方、それから、マルチ・メディアのインテリジェント・インタフェースという4つです。

これらの4つのサブセットについて見てみたいと思います。これらの4つが、学問界、専門的な世界、産業界、また、社会の個人々人に対してどうい影響を与えるかということ、簡単に述べたいと思います。

学問界についてですけれども、直接的な影響を与えると思います。というのは、学問研究者は、コンピュータ・サイエンス、その他に関係のある人は、非常に大きな興味を持っているからです。

そのほかにも、間接的な影響というのもあると思います。つまり、認識とか言語学、心理学といった分野にも間接的な影響を与えると思います。

そこで問題なのは、このような、学問の発展、科学の発展が孤立したものになってしまうか、あるいは、それがほかの学問を刺激し、促進できるのかということです。

物理学というのは、多くのほかの部分でこれまで刺激してきました。ですから、コンピュータ科学というのも、現在、このプロジェクトをやるということ、同じこと、つまりほかの学問に影響を与えるということをやっていると思います。

そこで、数学的な考えだけを取るのではなくて、それをいろいろなもの、数値的なものだけではなくて、いろいろな学問に広げていくべきだと思います。

もうひとつ、専門的なエリアを見てみますと、認識のスタイルというのはどういものでしょうか。いろいろな職業についている人々の、専門的な認識のパターンというのはどういものでしょうか。

そういうものは、何かのやり方で援助を得たいと思っています。また、この会議の初めからわかってきたことですが、質を上げる、あまり訓練されていない専門家のレベルを上げることが、社会が発展していくためには必ず必要です。

そのようなタイプの、発展した、進んだ資格のあるスペシャリストがいないと、我々はこのような問題を解くことはできないと思います。

例えば、南北の対話というような問題も、ただ単に、人間がただ平等になろうということだけではありません。あるいは、もう一度貧しいレベルに戻ろうということではなくて、我々の能力を最大限に使って、それ

を達成していこうということです。

ですから、専門的なレベルにおいて、特別の技術をうまく適用させていくということで、ある特定の状況、いろいろな文化における、いろいろな状況に適用させていくべき能力を発揮しなければならないと思います。

そのためには、現在ここで考えられている開発中のツールが、十分柔軟性を持つものであって、いろいろな条件に適用できるものでなくてはいけないと思います。

そして、この道具は人間が使うものですから、人間自身、私たち自身を訓練して、このような、実際に与えられる状況に適用するものでなくてはいけないと思います。

さて今度は産業界ですけれども、第5世代コンピュータ・システムのような発展というのは、確かに、製品が売られるやり方を変えていくことになると思います。

売るというシステム自体ですが、これは産業界のいろいろな部門において問題になってきており、また、それよりも、コンピュータのツールでハードウェア、ソフトウェアの訓練という各部分で、ほかの業界にも増して問題になってきていると思います。

ですから、違ったタイプの市場があるということによって、非常に広い、サプライヤ側からいろいろ重大な問題が上がってくると思います。

ですから、いろいろな分野でのエンジニアリング、そして、ハードウェアとソフトウェアを供給するというものが合わさって考えていかなければならないのではないかと思います。

確かに、そういうふうなことであれば、この、経済というものが、情報のマーケットの中でうまく分化していくということにつながるかもしれません。

私、もうひとつ、社会的な役割についてもう少し敷衍したいのですが、より多くの問題がそこではあがってくると思います。

社会的な側面ということについては、いろいろな問題があると思います。現在の、政策という面から考えても関連性があると思います。

一方で革新があつて、一方では、世界的な問題に対する問題の解決への貢献という2つがあります。ですから、政治家にとって、この2つをうまくかね合せて、一番いい方法を見出す、自分自身の国ではより多くの技術革新を求めて、しかも、より多くの、いま求められている国際的な貢献をするというのを求めていかなければなりません。

それをしなければ、戦略的な分析、自分の国自体の適切な戦力の分析をやらなければいけないと思います。

それによって、人間的な資源の問題、資金として得られるリソースがどれくらいあるのかという問題、そういう問題も全部考えていかなければならないのですが、ここで、情報というリソースが、私たちの国の中で利用できるものです。

私自身の国、西ドイツというのは、皆さん方の国とそう遠い状況にあるわけではありません。非常に近い状況にあります。天然資源は少ないし、また、情報の資源というのは日本と同じようになりあると思います。

ですから、このような、インパクトの分析をするという、私自身、もっと広い意味でやったのではないかと思いますけれど、このことが、ひとつだけの見方ではなくて、いろいろな方法で見なければならぬということをよく示していると思います。

つまり、何が一番いいアーキテクチャで、何が良い言語であるのかというような、ふたつのものだけの検討ではなくて、ここで大きな問題というのは、このプロジェクトが、私自身がみるところで言いますと、国としてのニーズが本当にここにあるのかどうか、そしてこれが、いろいろな、違ったツールだけではなくて、マンパワーとか資源とか、いろいろ考えていかなければならないのですが、ほかにやり方があるのだろうかということを考えていかなければならないと思います。

集中化された計画の方法でよいのでしょうか。マーケティングのやり方でもいいと思いますが、賢明なやり方でマーケティング・パワーを使っていく、そして、プランニングの力をそこで伸ばしていくことによって問題解決につながるのではないのでしょうか。

ドイツの立場というのは、私たちはこの新しい挑戦を受けとめなければならない立場にあると思います。私たちの立場というのは、それほど悪くないと思います。あまりにひどい立場であれば、この分野にやる気をなくしてしまうわけですが、それほど私たちの国は悪い状態にはないと思います。

そして、最終的には成功できるような見通しをとってやっているわけですが、したがって、一国で成功することは決してできないわけです。したがって、国際協力というのは、不可欠であろうと思いますし、それが不可欠であろうということは、決して悪いことではないと思います。

また、今後の期待については、もう既にほかのペネ

ル・メンバーが発表しておられるので、ここは飛ばして、最後にひと言申し上げたいと思います。

非常にオープンに発表されたことは、大変素晴らしいことだと思います。こういう内容をこれだけオープンに言われたというのは、素晴らしいことであると思いますし、我々にとっても、質するところが大きかったと思います。以上です。有難うございました。

元岡 どうも有難うございました。次に戸田さんをお願いします。

戸田 本日の会議に出席させていただきまして有難うございます。

第5世代のプロジェクトは大変重要でかつ、野心的な計画であり、是非成功させて、コンピュータの歴史に、新しい一頁をつけ加えていかなければならないという感じを一層深くしております。

折角の機会ですので、この計画について2つの観点からコメントを差し上げたいと思います。

ひとつは、第5世代コンピュータの考え方というか、ポリシーについてであり、もうひとつは、第5世代コンピュータのマネジメントについてコメントを申し上げたいと思います。

まず第1の観点からですが、コンピュータの発展を機能面から振り返ってみますと、このスライドのようになるかと思えます。

最初のコンピュータは、ご存知のように計算だけをやったわけですが、その後、EDVACとかEDSACが出てきて、プログラム制御、メモリーといった概念が出てきました。

やや遅れて、オンラインの機能とか、TSSの機能ということで、コンピュータは通信の機能を持ってきたわけです。

最近のコンピュータは、文字認識であるとか音声の認識、センサーの技術によって、若干不十分ではありますが、知覚の機能を持ってきています。

問題は、コンピュータが持つべき次の機能は何かということだと思いますが、理解で

あるとか推論とか学習、判断、想像、といったいろんなキーワードが氾濫していますが、次の機能が推論であるということを中心にかなり明確に指摘されたのが第5世代の計画の特徴ではないかと思えます。

大変妥当な判断でありまして、敬意を表しております。



また、コンピュータの構造という面から見ますと、アーキテクチャと部品実装技術というハードウェア技術の二つの側面があります。

部品の進歩を振り返ってみますと、真空管、トランジスタ、IC、LSI、VLSIというように発展してきました。我々は、既に、VLSIの時代に足を踏み入れようとしていることはご存知の通りです。

アーキテクチャは、機能と、ハードウェア技術のトレードオフの結果決まるものであり、両方の発展に非常に密接に関係しています。

ここに示したように、コンピュータの機能の発展と部品技術の発展を対比してみますと、ひとつの大きな違いがあると思います。

機能の発展というのは、追加的に行われてきていて、部品の発展は世代交替の形で行われてきています。

コンピュータは次々と新機能が導入されてきていますが、古い機能がなくなってしまうということはないわけで、新しい機能というのは、古い機能と併存して進歩して現在に至ってきています。

これに対して、部品技術の方は、トランジスタが入ってきますと真空管は殆んど使われなくなってしまうという、置き替えという形で進歩してきています。

今回の第5世代の計画は主として推論機構の研究を中心として展開するというように私は理解しています。

推論の研究ということは、重要なことは極めて明らかであります。第5世代のコンピュータというのは、計算・記憶・通信・知覚の機能も当然持っているはずであり、こういうような機能全体を、VLSIでいかにして実現するかという観点からの研究も極めて重要ではないかと思っています。

次に、汎用か専用かという問題です。

現在のコンピュータの応用分野というものは、情報処理、通信、制御、といった3つの分野に大別されると思います。特に、制御への応用は、マイクロコンピュータの発明によって、全産業分野にわたって非常に急速に普及しています。

第5世代のコンピュータのインパクトというのはいろいろ考えられますが、その中で特に、推論の機構が入ったということによるインパクトということを考えてみたいと思います。

情報処理の分野での応用というのは当然ですが、この会議の初めから議論されているように、エキスパート・システムというような、いろいろな専門家の人が、自分の専門技術の拡大としてコンピュータを使用していくというのに大変役立つのではないかと考えま

す。

このためには、各分野の専門家が協力して、数多くのエキスパート・システムを建設してみるということが重要ではないかと考えております。

通信制御の面では、推論機能は、先程リードさんがインエキスパート・システムとおっしゃいましたが、ああいう感じではないかと思えます。マンマシン・インタフェースの改善という点で有効ではないかと思っております。

としますと、非常に広い範囲の産業界と関連が出てくるので、広範囲の応用に耐えるような汎用の推論機構を実現するというのが、第5世代プロジェクトの最大の目標となると勝手に思っています。

私の経験から申しますと、まだ経験が十分に重ねられていない分野で、最初から汎用性を狙った機械を設計するというのは大変困難ではないかと思っております。

反面、それぞれ勝手に別のものをつくるのがよいとは思いませんが、汎用なものをつくるということ、当初から念頭に置いて、当面は個々の、応用毎のシステムの実現に力を注ぐ方が実際的ではないかというふうに考えています。

このようなプロセスを得て、汎用的なフレームワークというものがどんなものであるかということが徐々に浮かび出てくるというのが本筋であると考えております。

次は、プロジェクトのマネジメントに関して若干申し上げたいと思えます。

プロジェクトのマネジメントという面からみますと、第5世代のプロジェクトは、プロジェクトが、金額的にも、要員の的にも、研究項目というような点でも、非常に大規模であるということ、第2番目に、基礎研究的な要素が非常に強いプロジェクトであること、3番目に、ここに皆さん、非常に国際的な人たちが集まっておられるように、国際的なプロジェクトであるということが、プロジェクト・マネジメントの観点から見た大きな特徴であると考えております。

そういう点を考えますと、ひとつは、テクニカル・トランスフェについて、事前準備を予め考えておく必要があると考えております。

先程も申し上げたように、コンピュータは、人間のためのツールであり、推論の機能が追加されると、その応用の範囲というものは急速に拡大する可能性があらうと思っております。

基礎研究の成果を実用化に持っていくためには、労力的には、基礎研究での労力の10倍以上の労力が実

用化に必要であると思っております。

したがって、この第5世代プロジェクトでの研究成果というものを早期に社会に還元していくためには、研究成果を、出来るだけ広い範囲の関係者が容易に入手し得る体制を整備しておく必要があると考えています。

第2に申し上げたい点は、柔軟な研究体制の整備が必要であるということです。10年にわたる長期計画ですので、計画の性格自体というものが、初めと終りでは変わってくると思えます。

例えば、計画の前期の部分ではフィジビリティの研究というようなことに重点が置かれますから、多くの代替案の並行開発ということが行われ、発散的なマネジメントが必要だと思えます。

後半に至りますと、実行の案を絞って、重点的な開発を行う必要があり、収束的なマネジメントをやっていく必要があると思えます。

こういうような、プロジェクトの成果、性格の変化に追従できるような管理体制が必要であります。

推論システムの開発とか、特に、その評価のためには、いろんな分野の、多数のエキスパートの協力が必要であると思えます。

先程の、インエキスパート・システムの開発にも、いろんな産業分野の専門家の協力が不可欠であります。

第5世代プロジェクトの推進には、政府、大学、産業界、のコンピュータ・エキスパートが緊密に協力して進めていくということは勿論必要ですが、コンピュータ以外の専門家も、喜んで協力してくれるという体制が是非必要ではないかと思っております。

国際協力がスムーズにいけるということも、プロジェクト・マネジメントの観点からはひとつの大きな問題ではないかと思っております。

このように、第5世代のプロジェクトというのは、技術的な目標という点から見ても、野心的で、挑戦的であるというふうに私は考えておりますが、研究開発のプロジェクト・マネジメントという観点から見ても、非常に新しい、数多くの挑戦的な問題を私どもに投げかけているのではないかと思います。

第5世代のプロジェクトを通じて、私どもは、技術的な知見を得るばかりでなく、研究開発のマネジメントの手法についてもいろいろ新しい貴重な体験を得るのではないかと期待して居ります。どうも有難うございました。

元岡 どうも有難うございました。では、最後に岡松さん、お願いします。

岡松 通産省の岡松でございます。もう少し早く政策担当者がどこかに顔を出すとよかったのかもしれませんが、今迄のテーマは、極めて高度な、技術的な事柄であって、私どもの出る幕がなかったものですから、最後のパネルに参加させていただきました。

今回、第5世代コンピュータの国際シンポジウムを企画したわけですが、このように多くの方々が海外から参加していただきましたことを、このプロジェクトを推進しております通産省として厚く御礼申し上げる次第です。

この会議で、今日を含めて4日間、議論に参加していただいた結果、大体おわかりいただいたと思いますが、第5世代コンピュータの開発といっても、いろいろな可能性の中から、将来のコンピュータ・システムのあり方、重要な影響を与えるコンセプトをつくり出していこうということが狙いです。

その意味で、このシンポジウムで、第5世代のコンピュータというものが、ズバリ答が出てくるとお考えになって参加された方には若干失望を与えたかもしれません。

しかしながら、この国際会議で、日本がこれから進めようとしているプロジェクトについて、こういうふうに、計画の段階から諸外国との技術的な意見交換を行っていく、そして、今後、非常に大きなインパクトを与えると思われる第5世代のコンピュータ・システムのあり方について、ともに考えていこうという意図で開催されたものです。

私どもと致しましても、また日本と致しましても、誠に初めての体験であり、また、非常に挑戦的なものであったわけで、このへんを先ずご理解いただきたいと思います。

今日のパネルのテーマは、インパクトということで、丁度いい機会ですので、通産省として第5世代コンピュータをどのように考えているかという政策的な背景をいささかご説明申し上げたいと思います。

これは、先程来、他のパネリストの方からお話のありました、社会的なインパクトについてのご質問にもお答えすることになります。



1980年代を迎えるにあたって、通産省としては、'80年代というのは一体どういう時代がくるのだろうかというところから政策的なアプローチを始めています。

その結果、結論から申しますと、日本の置かれた国際的

な環境というところから、'80年代になさなければならぬ通産政策の基本的な考え方は3つであるというのが私どもの結論です。

ひとつは、世界のGNPの10%を占めるようになった我が国は、ある意味の経済大国として国際的な貢献をしていかなければならない時代に入っているということが第1の点です。

第2の点は、資源的な制約があるわけですが、我が国は極めて天然資源に恵まれていません。狭い国土の中に1億1,300万人という人口が住んでいます。

この国を支えていくためには、今後、技術を以って国を立てていかなければならない。我々はこれを「技術立国」と言っていますが、これを第2の柱に据えているということです。

と同時に、我が国は、ややもすれば、技術あるいは産業、経済というものだけを前面に打ち出し、生活面への配慮が足らなかったのではないかという反省も込めて、これからは、活力と同時に、ゆとりのある社会をつくり上げていかなければいけない、これが私どもが掲げている3つの目標であります。

ところで、この情報技術を生かしていく、私どもはこれを「情報化」というように捉えています。我が国の情報化というものを考えると、'70年代は産業を中心に、特に製造業を中心に情報化が進んできました。

これが、申すまでもなく、我が国の産業の国際競争力の基盤になってきたわけですが、今後、これらの情報化というものは、社会、あるいは生活というように、広く広がっていくものであるということです。

即ち、情報化の技術というのは、単に、産業における生産性の向上という側面だけでなしに、もうひとつ、我が国にとって非常に大きな問題である省資源、省エネルギーに貢献するものだということ、更に、医療問題、教育問題、あるいは交通問題、都市問題といったような、これから、我が国社会があるいは先進国が共通に立ち向かっていかなければならない様々な社会的な問題の解決にも寄与していく技術であるということ。

更に、これらの技術を生かして、人間の活動領域というものを、更に更に広げていくことができるということです。

これらの情報化を生かしながら、活力とゆとりのある社会を構築していくということが大事であるというふうに考えています。

このために、いまのような目標を掲げて、これを実現するためにどのような政策を立てていってほしいのかというのが私どもが次に取るアプローチであります。

私どもはこれを、豊かな情報化社会の構築であるということで目標として掲げたわけですが、この情報化社会の構築のためには、まず、それに必要な社会的な基盤、制度的な基盤を整備しなければならないということです。

これが、先程来お話のでもした社会的なインパクトの問題にもつながるわけですが、ひとつは、コンピュータ・セキュリティと言われるカテゴリーの問題を処理していかなければならないということです。

即ち、コンピュータが更に産業、生活といったところまで広がってきますと、システム安全対策という問題、あるいはシステム・ダウンに対する対応策、更には、残念ながら我が国においても最近広がってきている、コンピュータを利用した犯罪についても対応策を考えていかなければならないということです。

このような、コンピュータ・セキュリティの問題がひとつの大きな課題であり、次に、プライバシーの問題があります。

先程、パネリストの方からお話がありました、プライバシーの問題は、我が国においてはまだそれほど深刻になっていませんが、我が国としても、OECDの委員会で決められた勧告の線に沿って、制度を整備すべく目下政府部内において検討を進めているところで

す。また、先程リードさんからお話がありましたように、失業問題といったものも、このコンピュータ化に伴うひとつの大きな社会的なインパクトであると思っています。

幸い、我が国においては、欧米諸国とは違って、それほど深刻な問題は目下のところ生じていませんが、今後、我が国社会も急速に高齢化社会を迎えていくということから、これについても十分配慮していかなければいけないということです。

と同時に、我が国においても、コンピュータ化というのは、管理社会につながるのではない、あるいはコンピュータ・アレルギーといったような現象が生じていることも事実であり、このためには、先程ご指摘のありましたような教育面における問題、あるいは様々な訓練、あるいは、いま申し上げたような、様々な問題に対する対応策を考えていかなければいけないということがひとつです。

また、これと同時に、私どもが考えていますのは、コンピュータの技術、いわば自然科学の進歩を社会科学の遅れが阻害するようなことがあってはいけないという観点から、様々な社会制度の遅れを是正してい

なければいけないと思っています。

そのひとつは、通信回線の問題ですが、現在のところ、通信回線については、国内的には極めて利用が制限されているということがあります。

しかしながら、技術の進歩は、オンライン情報処理、分散処理を可能とするような形になってきていますので、このような技術の進歩を社会に生かせるようにしていくということが、望ましい情報化社会の実現のためには必要ではないかということで、これもまた、政府部内において、目下様々な検討が進められているところ

です。これらの、いま長々と述べましたことが、情報化社会をつくり上げていくための基盤の整備の問題ですが、これらと並んで私どもが掲げているのが、情報化技術の振興策、即ち、情報化技術の研究開発の促進という課題です。

私どもは、ふたつのものを、うまくバランスをとりながら進めていかなければならないということで、全体の政策体系をつくり上げているのだということをお願いし上げて、前段のお話をしたわけです。

技術開発分野についても、政府がどのような分野に参与していくのかということが次に課題になるわけですが、私どもが考えていますのは、非常に大きな波及効果を持ちながら、実用化までに長期間を要するもの、危険負担、資金負担が大きなプロジェクト、あるいは、社会的なニーズが高く緊急の対応を迫られている分野、ひと口に言うならば、民間企業に任せておいては、逆に言えば、プライス・メカニズムに任せておいては進まないと思われるような研究分野について、政府はこれを進めていく必要があるのだというのが私どもが考えている基本的な考え方です。

と同時に、これらの研究開発は、いまご説明したところからおわかりいただけると思いますが、基礎的な分野、基礎的なR&Dの分野にこれを絞って政府は支援をしているということです。

第5世代のコンピュータもまさに、その中の重要なテーマであり、従来、我が国の情報化技術の研究開発は、外国において既に発表されたものを追跡する形で、後追い型の開発であったわけですが、今回の第5世代というような新しいテーマを設定し、いわば、創造的、先端的技術開発をしていこうという意味で、非常に新しい試みであるということです。

この狙いとするところについては、今日まで既に議論されてきたところですが、従来のコンピュータでは解決できないような大きな問題というものを、新しい

技術によって解決していこうということであり、その予想される波及効果、あるいはリスクの大きさというところからみて、先程申し上げた意味で、政府がこれを積極的に進めるべき分野であると私どもは考えています。

具体的にどのように進めていくかという話に入りますと、今迄のお話の中にしばしば出てきていますが、全体を3つに分けてスタートさせていくということなのです。

その前段階として、今年度、1981年度においては、第5世代のコンピュータ研究開発について政府の予算は僅かに1.500万円です。この費用は調査予算であり、3つの調査課題があります。

ひとつは、国として研究開発を推進すべき重要な技術開発課題は何か。研究課題の中で、国として扱うべきものは何かということ。

2番目が、この研究開発の計画のスケジュールを決めようということ。3番目は、この第5世代コンピュータの開発体制をどういう形で進めていく必要があるかということ。この3つについて今年度調査をしております。

したがって、これらについて答が出るのは、最終的に、今年度末、来年3月になるということなのです。

それ以降のプランとしていま持っているのは、先程申し上げました、3つの段階に分けて進めていこうと思っております。

第1段階を最初の3年にあてようとし、ここで、基本技術についての開発を行うことを予定しています。

その初年度として、来年度、国家予算として約5億円を要求しています。この5億円を含めて、第1段階の研究開発、即ち、最初の3年のことですが、これに約100億円の国家予算を投入する必要があるというふうに考えています。

3段階と申し上げましたが、第2段階はサブシステムの開発にあてる、これに4年を要し、最後の3年間でトータルシステムを組み上げ、合計10年間でプロトタイプをつくり上げるというのが計画の概略です。

いま、5億円、100億円という数字を申し上げましたが、実はこれは、通産省の計画で、目下大蔵省と折衝を重ねています。

お集りの皆様も、各々お国の事情でおわかりになると思いますが、財政当局というのは、財政再建、特に我が国においては財政再建ということがここ3年の課題になっていて、出来るだけ政府予算をカットしたいという意向を持っていますし、私どもは、情報技術と

いうものは、社会の根幹に触れる問題、社会基盤投資であるという観点から、是非ともこれを実現したいということで、非常に激しい議論を連日展開しているところですので、いまは、政府の原案というものではなく、通産省の案としていまのような数字を持っているとご了解いただきたいと思っております。

プロジェクトの推進体制ですが、私どもとしては、何か新しい団体を外につくり、そこに人材を集めて計画を進めていきたいと思っております。

最後に、国際協力の観点についてお話ししますと、今回のこのプロジェクトは、基礎技術の開発であるということ、非常に大規模のプロジェクトであるということ。

更に、極めて大きなインパクトを全体に与えるものであるということ。また、我が国としても、技術の先進国として国際的な貢献を果たしていかなければならないという政策姿勢等々を考え合わせて、何らかの形の国際協力を進めていくべきプロジェクトであるというふうに考えております。

その第1歩としてこのような形で国際会議を開いたわけですが、今後、具体的な進め方については、今年度の調査の中で明らかにされていくわけですが、国際的な場においてもこのプロジェクトについて、今後とも、何らかの形で絶えず発表をしていくという機会を作っていきたいと思っております。

また、このようなことを通じて、現在先進国下において議論を致しますと、どちらかというところ、成長についての悲観的な見方が多いということですが、このような、先進的な大プロジェクトを通じて、世界経済を再活性化していくことに貢献していければ誠に有難いと考えている次第です。以上、最初のコメントとしてお話しさせていただきました。有難うございました。

元岡 ども有難うございました。これでひととおりお話いただいたわけですが、パネリストの方でどなたかコメントをなさる方、おいででしょうか。それではフロアーからコメントをどうぞ。

日本人フロア 北海道の釧山で100人も死んだという事件が先日ありましたが、コンピュータを利用し採鉱というか、人間が、炭鉱の底とか、条件の悪いところにもぐらないでもいい、例えば鉱夫は、テレビジョンの前に座ってロボットを動かして、中に入っていくのはロボットだというふうな応用はどうでしょうか。

日本は、世界の陸地の0.3%しかないのに、日本の国内だけ考えますと、釧山のコンピュータ化というの

はあまり需要が多くないように見えるかもしれませんが、いまの話で、国際的な役割をとうことでしたので、99.7%の鉱山は世界のどこかほかにあり、そこにもくり込んで非常に危険なことをやっているということです。このプロジェクトがうまくいくとそのような危険も解決できるのでは。

何かコメントがありましたら…。恐らくは、世界中についても同じことが言えると思うのです。

唐津 おっしゃるとおりで、初日に私、ダーティワークを知能ロボットにやらせようと。そのときに例として出しましたが、たまたま、まだ爆発していませんでしたので、非常に深い海の底の仕事、それから、非常に危険なフィールドでやらせる、そういう例をあげたわけで、それは、私どもの分科会でも、いわゆるインテリジェント・ロボットを是非やるべきだということで、また、このプロジェクトの何番目かにかかっていると思います。

元岡 リヨンさんから、訓練と教育の問題が重要なインパクトとして取り上げられて、唐津さん、そちらについてかなりご見識をお持ちのように伺っていますので…。

唐津 確かに私、1日目に落ちこぼれ問題といういわゆるドロップ・アウトといわれるもの、それがコンピュータの導入によって非常に減るであろうということをご説明したかと思います。

先程からあります推論の問題が出てきましたが、この技術が発達しますと、世に言われるC A I、現在のものは、言ってみればまだ非常に低いレベルのC A Iですが、推論というものがコンピュータによって巧みに処理されるようになりますと、ドロップ・アウトの問題というのは非常に減るであろう。

更にもっと重要なのは、特に日本では老人が増えてくるということが大問題です。ということは、職業構造が非常に変わってくるということです。

この2つを考えると、いわゆる生涯教育という問題が非常に出てくるわけで、そのときにいかに効率的にやるかということのためにこの第5世代が非常に役に立つ。また、このプロジェクトの中にもそれを取り上げていただくように提案しています。

恐らくそれは、かなりの効果をもたらすであろうと思います。これは、先進国だけではなく、特に途上国に対して、このテクノロジーのトランスファーというのは非常に意味があるのではないかと考えています。

カーシュ 私どもが、コンピュータ・サイエンスやエンジニアリングでどのような利益があった

かということ言えばいいのですが、いずれも30年ぐらいの経験しかないわけです。

しかし、幸いにして、200年前の米国の大統領が勧告を出しています。

リードさんの政府と私の政府が、米国の建国についての議論をしたあと就任しましたジョン・アダムス大統領が、1780年に書いたことを引用したいと思います。『私は、政治学と戦争について勉強しなければならない。そうすると、息子は自由に科学とエンジニアリングを勉強できるだろうから。そして、その更に子孫が、音楽や芸術や彫刻やダンス、そして哲学を勉強する自由ができるだろうから』と言うのが、アダムス大統領が1780年に言った言葉であります。

世界の国々の中には、第3世代を迎えようとしている国があるわけですが、コンピュータ・サイエンスやエンジニアリングの世界において、第5世代に入ったら、第3世代を助けるようなものができないでしょうか。

元岡 インダストリーからいろいろの方が見えているので、インダストリーの方々の意見を聞いてもらいたいという声が強いのですが…。

特に、IBM初め、米国の多くのインダストリーの方々が見えているので、最後に何かコメントをいただけないでしょうか。

ショーア 今回、お招きを受けまして非常にうれしく思っております。ほかの方がおっしゃったことですけれども、皆様方、非常に卒直な態度を持っていらっしゃるということ、プランを公開して下さったこと、非常にうれしく思っております。

これまで、うまく状況を説明するようなものがいろいろ出たと思います。前向きな性格が強いプロジェクトだと思います。皆様方、非常に興味深いプランニングをこれまでやっておられます。

淵先生のお話の中などで、このプロジェクトは出発点である、基本計画の出発点であるということをおっしゃいました。

メーカーはどちらかという保守的な片寄りがある、また富士通などでも同じだと思いますが、そういう傾向があるということをおっしゃいましたが、私自身は研究部門から出てきたので、それほど保守的ではないと思いますけれども…。

皆様方が示してきた、この前向きな性格というのは非常にうれしいものだと思います。基本研究という面から見ても、非常にリスクの大きいものだと思います。勿論、全部が成功するとは限りませんが、

非常に心強く思っています。

そして、1年後にここにまた戻ってきて、皆様方がもっと具体的な成果を発表できるような準備ができていっしょに段階になるであろうと、非常に期待しています。そういう機会に恵まれて、皆様方の、こういう結果が得られたというお話を是非聞きたいものだと思っています。

ディラギー デジタル・エキップメント(D E C)のビストラギーです。ファイゲンバウム先生が言われましたように、DECではエキスパート・システムを使っております。

私、レーガン大統領の代弁をするつもりはありませんが、このプロジェクトは大変素晴らしい明快なものであると思いますし、また、同時に、多くの人が参加できるような枠組をつくっておられると思います。

また、一貫したやり方で協力できるような方法をとっておられると思います。

もし、小さい声でファイゲンバウム先生に申し上げさせていただけますならば、エキスパート・システムはかなり数多くのものが世界に出てくるであろうと思います。そして、その中で、何が重要であり、重要でないかということがだんだんははっきりしてくるであろうと私は思います。

バット インドのバットです。このセクションでは、ロジカルに第3世界の持っている懸念というものを表明していたように思います。

私の国の現状から考えましても、エレクトロニクスの進歩というものは、私は記者に尋ねられましたときに、例えば、1.600メートルの長距離走者は、4分の1ぐらい走ったときに、また力を出して走っていくだろうと思います。だれかに追いつかれることがあるだろうと思います。

テクノロジーというのは、変動があり、そして大きく飛躍していくものであります。したがって、国際的な理解というものが必要だと思います。

先程のジョン・アダムス大統領の言葉の引用がありましたけれど、いまや、経済における新しいアートを学ばなければならない時代にきていると思います。

ひとつ警告を発するならば、リードさんが言われたように、農業から製造業へ、そしてサービス業へと変わってきましたが、サービス業の多くの機能が自動化される時代にきているということ認識する必要があります。以上です。有難うございました。

外人フロア はば今回の会議の終りに近くなっていると思います。終るということは、ひとつの、

社会的インパクトのあるものです。

ずいぶん前に亡くなった大統領の引用がありましたので、スタンフォードでまだ生きているコンピュータ・サイエンティストの言葉を引用したいと思えます。

知識工学の詳細を勉強していますと、よくマッカーシーさんの言われたことを思い出します。

インスピレーションについて言われたことです。

『100年分の思考をすべての決定に盛り込むことができたならば、世界はどうなっているだろうか、あるいはどうなるだろうか』ということをよく言われましたが、第5世代のコンピュータを考えるときに、それがやはり、ひとつの手掛かりになるのではないかと思います。

元岡 どうも有難うございました。まだいろいろの方がらご意見を伺いたいのですが、もう時間も終りに近づきましたので、これで閉会にさせていただきますと思います。

最後に、この会議を組織致しましたものとして、お忙しいなか、世界中からこんなに沢山の方がお集りくださいまして、我々のプロジェクトについて終始熱心にいろいろのご意見を、それも、非常に建設的なご意見をいただいたことを深く感謝しております。

このプロジェクトの応用という面からいろいろのご意見があったと思います。勿論、第5世代のコンピュータを考えるときに、それが何に使われるか、何に使うかということが非常に重要な問題であるということは我々もよく理解しております。

しかしながら、一方でリソースの問題もあり、それほど、ひとつのプロジェクトを幅広いものにするということができないという判断から、このプロジェクトでは、応用の分野を切り捨てた格好になっています。その問題について、いろいろご批判があったように私は理解しています。

そのへんの、調整という問題が、これから大事になってくるのではないかと思います。

少し大袈裟なことを言いますと、知識の世界と言いますが、このプロジェクトで目指すものが、知識の世界におけるスペース・シャトルのようなものであると考えます。

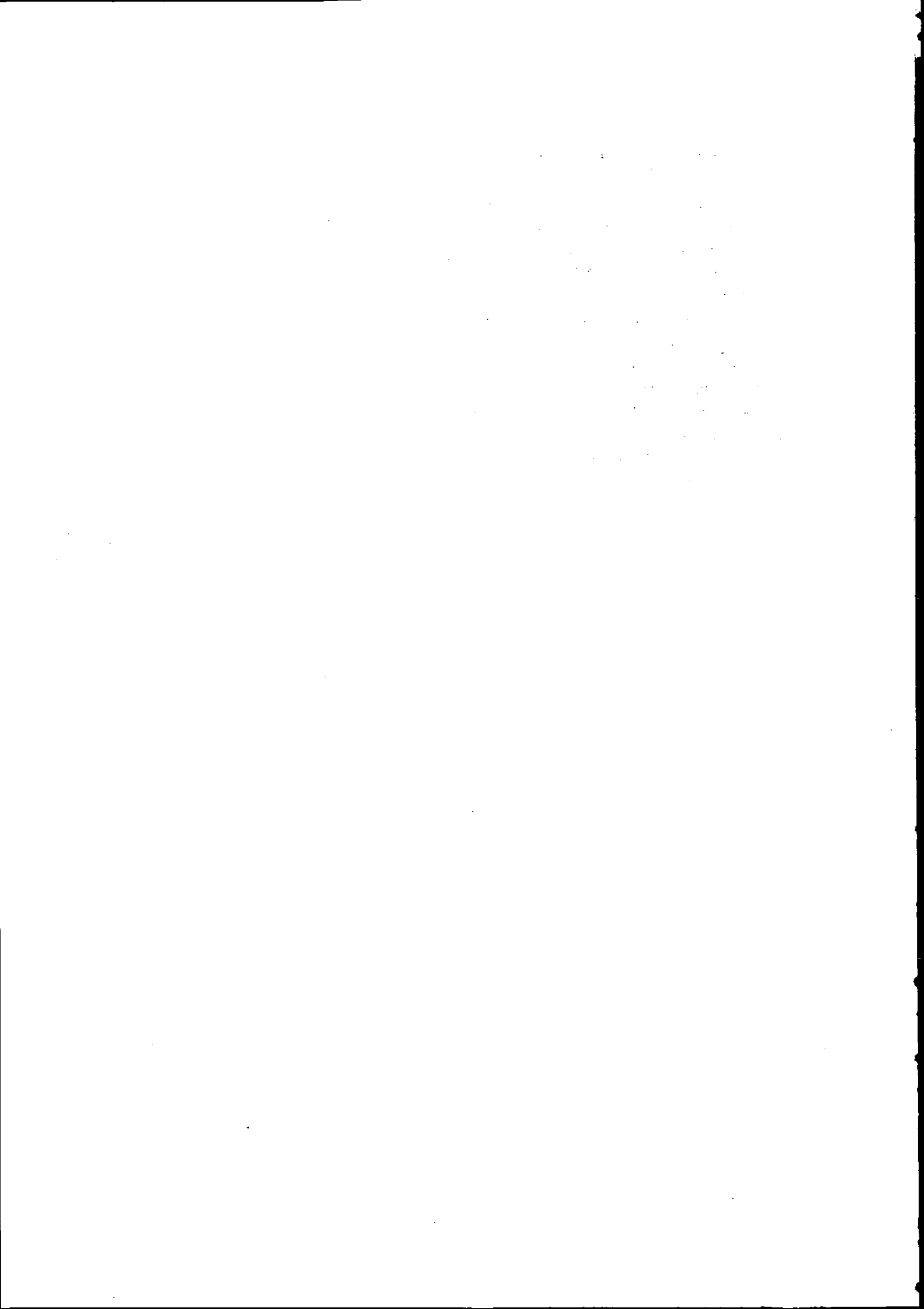
スペース・シャトルというのは、あの上でいろんなものを実験したり、あれから新しいものが生まれてくるということ思い出して、しかも、そういう、新しい未知のものへの挑戦に対する経済的な要素も考えて計画されたプランであると私は考えます。

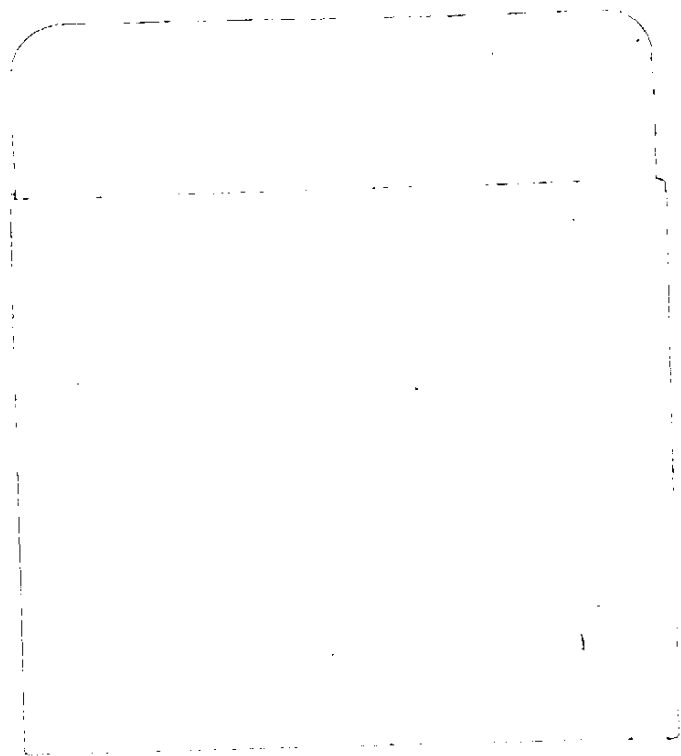
スペース・シャトルは、実際の、我々が目に見える

宇宙でありますから非常に解りいいのですが、このプロジェクトは、人工の宇宙、アーティフィシャルな宇宙、あるいはバーチャルな宇宙といった、知識の世界におけるスペース・シャトルのようなものですから、わかりにくい点がいろいろあると思います。

スペース・シャトルで何が国際協力で行われたかということを考えてみますと、スペース・シャトルの上でどんな実験をするかということに関しては、世界中からいろいろ案を募ってやっています、日本からも、何年か先のスペース・シャトルには、日本がプロポーズしたものが乗る話になっていると思います。そういった国際協力の方法もあるのではないかということ、この会議の間に考えておりました。

長い間、本当に有難うございました。これで閉会にしたいと思います。







LATE PAPER

**INTERNATIONAL CONFERENCE
ON FIFTH GENERATION
COMPUTER SYSTEMS
1981**

JAPAN INFORMATION PROCESSING DEVELOPMENT CENTER

*Copyright 1981 by Japan Information Processing Development Center.
No part of this publication may be reproduced without written permission of the publisher.*

ALGORITHMS, ARCHITECTURE, AND TECHNOLOGY IN VLSI

by

JONATHAN ALLEN

Massachusetts Institute of Technology
Research Laboratory of Electronics and
Department of Electrical Engineering and Computer Science
Cambridge, MA 02139

ABSTRACT

With the advent of VLSI, the design of high-performance digital systems changes from the realization of an algorithm on a given architecture fabricated in a fixed technology to an integrated process wherein the mutual interaction of algorithm, architecture, and technology establishes a balanced design satisfying the system goals. In this paper, these factors are examined in terms of the computational needs of Fifth Generation Computer Systems, with emphasis on applications in signal processing, speech generation, and speech understanding.

The evolving silicon planar technology is providing capability for the fabrication of individual circuits containing millions of devices. Complex systems cannot exploit this technology, however, without the limitation of architectural elements to restricted classes of canonical forms which can be modularly expanded to reflect the natural hierarchy of the task domain. A specific domain where such a basis set of functional units is particularly useful is that of spectral analysis and filtering, where the technology now permits full utilization of all possible parallelism. For example, N point FFTs can exploit up to $(N/2)\log_2 N$ complex "butterfly" units, and such a high performance architecture in turn demands a technological base requiring wafer-scale integration and accompanying redundancy techniques dictated by yield considerations. In this way, the modular size of units, the architecture of interconnection, the technological base for testing and restructuring, are all seen to interact in complex ways. Whereas in previous systems, natural modular elements were built from smaller separate elements, now the characteristic limitations of the technology establish the degree of redundancy and size of modular elements, thus strongly constraining the overall architecture. Today redundancy techniques are applied at the chip level, but large new systems will require system level utilization of redundant elements. These large systems will also require a testing architecture and technology capable of comprehensive testing of modules at fabrication time for use by interconnection processes, but also self-testing during field utilization to continually establish the viability of these complex computational engines. Technological innovations may also lead to three-dimensional structures that will permit new interconnection strategies and architectural forms.

The massive parallelism afforded by the evol-

ving technology reveals, however, the need for fundamental understanding of algorithmic representation. As usually expressed, algorithms confound the competence or semantics of a task with its performance. In a world restricted to single-sequence von Neumann machines this was not a severe limitation, but now the system designer requires the systematic means to explore space/time/power tradeoffs as various performance options all the while retaining the basic competence of the prescribed task. New results providing the theoretical basis for these performance explorations are finding use in design systems that provide the system designer with a comprehensive choice of options, hence merging a class of algorithms for a given task with a corresponding set of architectural possibilities. The availability of extensive parallelism has also encouraged new compilation techniques for the design of data path-units incorporating distributed arithmetic-logic capability over a set of registers. VLSI design strategies of this sort thus encourage the designer to exploit parallel operations on specialized registers in a way not possible with previous general-purpose architectures. Future design systems will discover such task-specific parallelism and automatically compile architectures that reflect these possibilities.

The interacting constraints due to technology, architecture, and algorithm provide a particularly rich set of design options in systems for digital signal processing, speech generation from text or concept, and speech understanding. Systems capable of supporting interactive design for these applications will soon permit the realization of high performance complex engines. It is of especial interest that the techniques described here for use in Fifth Generation Computer Systems will provide the basis for both the requisite design systems and the application systems produced by their utilization.

I. INTRODUCTION

Fifth Generation Computer Systems will be used to solve very complex problems with a very high level of performance. In order to design these systems successfully, they must be seen as the result of integrating together several different (but interacting) constraint domains, each of which has its own characteristic structures. The constraint domains are

1. Technology
2. Algorithm
3. Architecture
4. Data Structure
5. Programming Language

It is the claim of this paper that the best

designs are obtained when these structures "match" or interrelate in a complementary way, each supporting the other. Each of these design domains has a substantial internal theory, but also overlaps with the other domains, and the system designer must be aware of the design freedom and constraints at each level, as well as the consequences of design decisions at any given level on all of the other interacting levels.

In this paper, we will focus on the first three constraint systems, although we of course do not wish to diminish the importance of the additional constraints. This will enable us to once again see integrated circuit systems as the realization of multiple interacting constraints specified by different levels of representation.

Once again, we will observe the necessity for well-formedness within each constraint system, but also that these representations "line up" with each other (e.g. logic, circuit, and mask layout) so that changes to one level automatically propagate to other levels. We will investigate the interplay between algorithmic representation and architectural representation, and examine the influence of technology on the latter. We will then concentrate on the nature of integrated circuit design, and the way in which these systems are specified at all levels and verified for correctness. Examples of this process are drawn from signal processing, where there exist many useful canonical structures that are rich in parallelism. These algorithm structures and representations are well-understood, and allow us to readily appreciate and invent appropriate architectures and technology.

We then move to architectural considerations where there are less well specified canonical forms and primitive computing elements. We consider a scheme for implementation of large-scale semantic nets, and then move to a discussion of speech synthesis and understanding. Once again, we will encounter multiple interrelated levels of representation. Most current speech recognition systems consider only the waveform aspect of a speech utterance, but we show that comprehensive recognition of a broad class of utterances must integrate together the constraint strengths from a number of different levels of language representation, including syntax, semantics, phonology, and the structure of the lexicon. In this way, we see again a problem domain where various levels of representation must be brought together in complex ways.

In the discussion above, each of the constraint domains is of course a knowledge source, and we wish to point out that algorithm, architecture, and technology are all complex knowledge sources, each with a substantial internal theory, but also with non-trivial overlap. It is the specification of systems to cope with such problem areas that will substantially occupy the designers of fifth generation systems.

II. FUNCTIONAL SPECIFICATION

An initial specification of an algorithm and

corresponding architecture is currently done in most cases by the human designer in an informal way. An important area of research, however, is the means to derive an initial specification of the system from a functional specification. Here, we must understand much of the designer's expert knowledge, and how it is used to construct the system. We consider two examples:

1. Digital Filter design. Programs exist to convert a functional specification of, say, a band-pass filter (frequency band, in-band ripple, out-of-band attenuation, etc.) to a canonical digital filter form, complete with numerical specification of the digital filter parameters. This may be thought of as a complete functional to architecture (and implied algorithm) transformation, and hence captures much design knowledge.

2. Analog circuit design. Starting with a functional specification of an output signal waveform, there has been some initial success in representing the circuit designer's knowledge in a rule based system. These rules decide on the parts (or modules) of the resulting system, and the circuit topology (or interconnection of modules), and produce one possible circuit that satisfies the functional requirement. These design rules do not specify a "best" circuit, and so there is need to further investigate the design along the dimensions of various performance criteria.

Research on such design strategies starting at the functional level is clearly of fundamental importance. The initial research indicates that it is possible to represent the output of the designer's initial activity as an architectural structure composed hierarchically of basic primitive or canonical elements. But there remains the task of investigating and varying the performance of the design.

III. ALGORITHMIC AND ARCHITECTURAL REPRESENTATIONS

In many problem domains, we are often given an initial algorithm and architecture together. Unfortunately, this confounds the competence (what an algorithm does) with its performance (how the algorithm is executed) since the algorithm is expressed in terms of an architecture. Usually algorithms are expressed in terms of a von-Neumann single-sequence computer architecture, but this tends to obscure potential parallelism, and represents the algorithm at only one performance level.

1. We would like to have a fundamental means to represent separately the competence and performance of an algorithm. While there has been some initial research, there is not yet such a technique appropriate for broad classes of algorithms. There is need for much more research here.

2. It is possible, however, to introduce competence-constrained performance transformations that permit systematic manipulation of performance while guaranteeing that the competence is preserved. Two examples:

a. A system specified by a hardware design language scheme can be transformed to any other theoretically possible performance. Techniques for these transformations utilize basic ideas of conflict resolution.

b. In specific cases, such as gate arrays, the system designer can change performance features (such as fan-in/fan-out, pin count, logic delay) with a set of basic transformations that guarantee that the intended logic function is not altered.

3. Once a given architecture is obtained via transformations, then its consequences can be derived in a given technology. For example, estimates of spatial area, time delay, and power consumption can be made in a technology-dependent way. The realization of a particular architecture in a selected technology is done hierarchically in a way that displays the available useful parallelism, and this means that module interconnections must be made explicit (topologically and in area utilization), and this problem induces close coupling between architecture and technology.

Examples of architecture/technology interaction:

a. Partition size vs. redundancy for large systems utilizing wafer-scale integration. The need for redundancy in such systems introduces additional potential parallelism in the architecture, and means that logical locality may not translate into physical locality, with resulting performance limitations in space and time.

b. Integrated circuits are thought of as 2-dimensional arrays of devices now. However, with introduction of stacked epitaxial silicon layers, 3-dimensional structures are possible, and these may be profitably exploited for C-MOS. Furthermore, the metal interconnect and contact technologies have very strong effects on the architecture and its performance in both space and time.

c. In general, technological changes can be traded for architectural changes. In the design of real-time systems, a high-speed technology does not require as much exploitation of parallelism as a lower-speed technology. There will always be a tendency to exploit technological performance in an attempt to limit the control complexity required by parallelism.

IV. GENERATION OF ARCHITECTURAL UNITS

The effect of technology on architecture is reflected in a continuum of interconnections strategies:

1. fixed at design time
2. fixed at fabrication time
3. dynamic connection strategies that utilize on-line testing and link formation.

Due to faults arising from defects, performance may not be strictly predictable at design time but only boundable. These bounds can be obtained from new techniques that statistically predict yield from a detailed mask layout specification. Knowledge of yield factors at design

time in a quantitative way permits the designer to further explore architectural options of module size and parallel redundancy. Such techniques, together with reconfiguration procedures at fabrication time, have increased yields of 640 K RAM chips by a factor of 3.5, a fact that has enormous economic consequences.

We now assume that a desired architecture (complete with redundant elements if needed) is available, and we must implement it using some IC technology such as NMOS or CMOS.

1. Several levels of representation must be generated, correctly maintained, and properly aligned.

These include architecture, logic, circuit, device, and layout specifications. Due to the complexity of modern chips, there is a search for universal canonical modules and hierarchical connection and verification strategies that can reduce the amount of "random" detail considered by the designer, and hence these are called structured design strategies.

(1) A number of basic canonical forms have been specified. These include standard cells, gate arrays, program logic arrays, finite state machine structures, register arrays, and arithmetic-logic unit components.

(2) It is important that these forms be represented in a way that permits parameterized instantiation (i.e. logic function, interconnect "pitch," transistor size). This suggests an analogy with software procedures, and the idea of a type. Hence a module is represented as a program. To design a module is to write a program to generate it. Such techniques permit the correct generation of many logic and register functions by a direct transformation from the functional specification to detailed mask layouts. There is no question that increasingly the IC designer has diminished need to concern himself with the high complexity of layout with such canonical procedures. These techniques also show the equivalence of program and data, and hence the popularity of languages such as LISP for this purpose.

(3) Not only can individual modules be represented by program, but entire systems can be instantiated using similar techniques.

a. control can be realized with finite state machines that represent multiple processes.

b. maximally parallel data paths can be formed by distributing arithmetic-logic units over a register array in a parallel way. Programs exist to automatically place these elements and create the requisite local busses for interconnection. The designer can thus specify special purpose registers together with corresponding dedicated processing capability.

(4) Modular, Hierarchical Design Overview.

(5) Themes Throughout.

2. Example

Signal processing tasks can utilize the techniques for design that we have discussed.

a. Basic canonical cells. There are a small number of basic computations that are extensively used in signal processing. Examination of a number of tasks (recursive biquadratic filter,

radix 2 FFT computational element, finite impulse response filters, lattice filters, and correlation) shows the elements that can be contained in a basic reconfigurable cell.

b. Fully parallel FFT. In order to exploit all possible parallelism in an FFT, a large structure is needed suggesting restructurable wafer scale integration. Such a realization requires partitioning and redundancy strategies that transform functional modularity into technological modularity and augment the basic parallelism with additional elements dictated by redundancy.

c. This example shows the way in which constraint domains represented as knowledge sources are integrated together in a complementary way to achieve the final design.

V. SYSTEMS FOR INFERENCING THROUGH INTEGRATION OF KNOWLEDGE SOURCES

In this section, we consider system design where the abstract levels of representation are not as well understood or established as in the previous sections when we dealt with more concrete entities such as logic, circuits, and signals. Our examples are drawn from the field of language processing.

1. Semantic nets realized by a connection machine. Here we consider the basic problem of retrieving knowledge. A few simple operations are used repeatedly:

- a. deducing facts from semantic inheritance networks
- b. matching patterns, or finding the best match
- c. sorting according to some parameter
- d. searching graphs for sub-graphs of a specified structure.

A switching network is provided that permits the realization of many virtual networks on one basic architecture. Communication is achieved by message passing over the uniform array of cells, each of which contains a simple processor containing a few registers, a state vector, and a rule table. Work is currently under way at MIT to generate a large machine of this type for artificial intelligence research.

2. Text-to-Speech Conversion for unrestricted English text is now well understood. It can be represented as a linguistic analysis of the text followed by a generation of the speech waveform from an underlying phonetic representation.

a. In the text analysis, word structure, letter-to-sound rules, stress rules, boundary adjustments, and low-level phonetic contextual adjustments are made.

b. The speech waveform is generated by first deriving a timing structure, then the pitch waveform, and finally the detailed phonetic parameters needed to drive a model that produces the output waveform.

Although there remain problems of representation for lexical units, the relation of syntax to phonology, and pitch contour, these techniques are presently capable of producing highly intelligible and reasonably natural sounding

speech. A VLSI hardware system to realize these functions, however, has not been constructed, although it should be possible using wafer-scale restructurable technology. It is important to see this system as a means to discover and integrate a wide variety of language constraints in order to achieve the construction of a satisfactory speech waveform.

3. Speech Recognition and Understanding

Here we contrast two views of the problem.

a. Most current commercial systems implicitly assume that the information in the speech signal is sufficient to recognize the utterance, and hence the speech recognition problem becomes a pattern matching problem, usually employing some parametric representation of the speech (such as linear predictive coding). Because of speaker variation and timing changes, an optimal dynamic programming match is performed. Fortunately, a regular two-dimensional architecture can be devised for these intensive computations, and great parallelism can be exploited to perform the dynamic time warping in real time.

b. For large vocabularies, realistic syntax, multiple speakers, and noisy input, the speech understanding problem becomes very difficult. Much research is needed to develop a cognitive model for human speech perception. While much remains to be learned, the best current knowledge represents the generation of a percept as an integration process over many knowledge sources. Each of these knowledge sources has a substantial internal theory, but also interaction with the other sources. In this way, constraint strengths due to syntax, semantics, phonology, and the lexicon are exploited at appropriate points in the time-evolution of the perception to build the linguistic or textual representation of the utterance. Even at the phonetic level, many different parameters interact in complex ways with context-dependent tradeoffs to determine the phonetic segment features. Fifth Generation architecture can be expected to provide the framework needed to represent these structures and provide for their mutual interaction. We should realize, however, that these are exceedingly difficult problems which will require extensive contributions from many talents.

VI. SUMMARY

We have shown how architecture, algorithm, and technology interact to form VLSI systems. The way in which these structures interact is also found among the various levels of VLSI design representation, and the several constraint domains utilized in natural language tasks. These observations show that much of the complexity faced by the need to represent and construct solutions to the problems faced by Fifth Generation computing systems is profitably thought of as systems of mutually interacting knowledge sources with various degrees of coupling. Further research will specify in more detail the nature of such knowledge representation, and the inference mechanisms needed to yield consistent and correct systems.

VII. ACKNOWLEDGEMENTS

I am pleased to acknowledge the many contributions of my colleagues at MIT and elsewhere who have contributed to the work reported here. They include D. Antoniadis, J. Batali, P.

Blankenship, Y. Brown, K. Crouch, J. Darringer, S. Garverick, D. Hillis, W. Joiner, G. Miranker, P. Penfield Jr., E. Pierce, J. Raffel, G. Roylance, D. Seccombe, H. Shrobe, D. Silver-smith, J. Siskind, J. Southard, G. Steele, and G. Sussman.

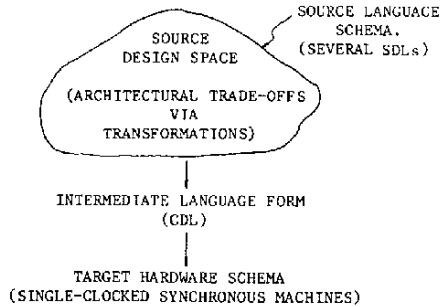
COMPUTATIONAL SYSTEM DESIGN

MUTUALLY COMPLEMENTARY CONSTRAINT DOMAINS

1. TECHNOLOGY
2. ALGORITHM
3. DATA STRUCTURE
4. PROGRAMMING LANGUAGE
5. ARCHITECTURAL UNITS

BEST RESULTS WHEN THESE CONSTRAINT STRUCTURES MATCH

TRANSLATION OF HIGH LEVEL HARDWARE DESCRIPTION LANGUAGES



DIGITAL FILTER DESIGN

1. TRANSFER FUNCTION DESIGN MODULE:

$$H(z) = \frac{\sum_{i=0}^M a_i z^{-i}}{(1 + \sum_{j=1}^H b_j z^{-j})}$$

FIND M, N, a_i, b_i

2. FILTER STRUCTURE AND COEFFICIENT OPTIMIZATION
3. ANALYSIS

FREQUENCY/TIME RESPONSE
ROUND-OFF NOISE

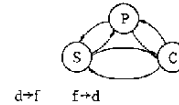
HARDWARE DESCRIPTION SOURCE TRANSFORMATIONS

- 3 KINDS OF BLOCKS:

- P-BLOCKS: no stated order of execution of statements
- S-BLOCKS: standard sequential order
- C-BLOCKS: all RHS's simultaneously executed, then all assignments in parallel.

- 2 ITERATIVE CONSTRUCTS:

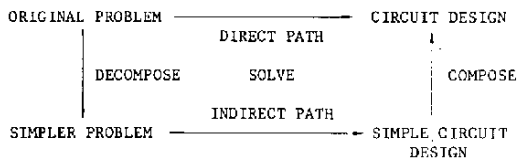
- DO: Standard Fortran temporal
- FORALL: i ∈ [] do spatial



GLEN MIRANKER

SIMPLE CIRCUIT DESIGN

STRATEGY:



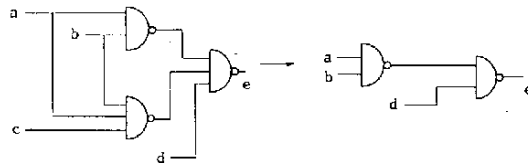
COMPUTATIONAL ORGANIZATION:



GATE ARRAY LOGIC SYNTHESIS FROM A FUNCTIONAL SPECIFICATION

1. GENERATE A NAIVE IMPLEMENTATION AUTOMATICALLY FROM A FUNCTIONAL SPECIFICATION.
2. IMPROVE THE NAIVE DESIGN INCREMENTALLY BY APPLYING LOCAL TRANSFORMATIONS.

NAND GATE TRANSFORMATIONS:



A PARTICULAR ARCHITECTURE
IN A
SELECTED TECHNOLOGY

REVEALS

AVAILABLE USEFUL PARALLELISM
EXPLICIT INTERCONNECTIONS

ARCHITECTURE/TECHNOLOGY INTERACTION:

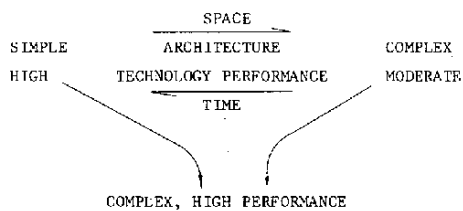
1. PARTITION SIZE vs. REDUNDANCY
 WAFER-SCALE INTEGRATION
 INTRODUCTION OF ADDITIONAL PARALLELISM
 LOGICAL LOCALITY/PHYSICAL LOCALITY
 YIELD PREDICTION
2. 3-DIMENSIONAL STRUCTURES
 EPITAXIAL SILICON FOR MORE LAYERS
 3-D CMOS

BASIC CANONICAL FORMS
FOR
VLSI DESIGN

1. NAND, NOR, FLIP FLOPS
2. STANDARD CELLS
3. MEMORY MACROCELLS
4. GATE ARRAYS
5. PROGRAM LOGIC ARRAYS
6. FINITE STATE MACHINES
7. REGISTER ARRAYS
8. ARITHMETIC-LOGIC UNITS

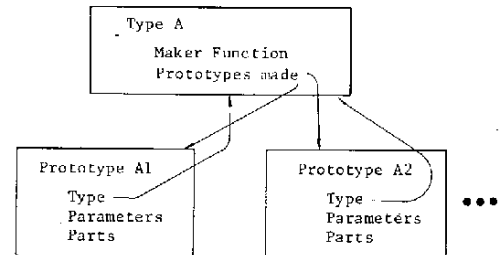
DESIGN TRADE-OFFS

LOW	PARALLELISM	HIGH
HIGH	LOGIC SPEED	MODERATE
SIMPLE	CONTROL	COMPLICATED



DESIGN PROCEDURE LANGUAGE

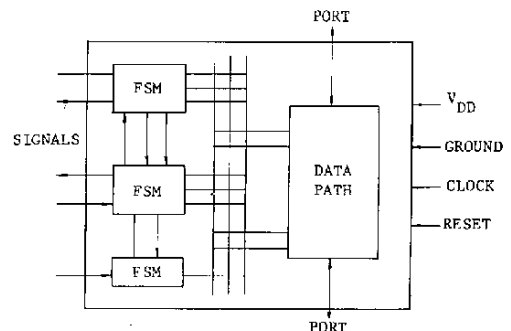
DESIGN A CIRCUIT IN DPL BY WRITING A PROGRAM THAT,
WHEN RUN, CREATES A REPRESENTATION OF THE CIRCUIT IN A
DATA BASE.



LEVELS OF REPRESENTATION
IN
IC DESIGN

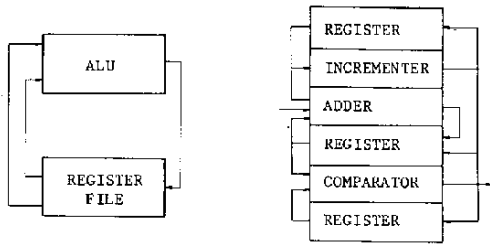
1. ARCHITECTURE
2. LOGIC
3. CIRCUIT
4. TOPOLOGICAL NETWORK
5. MASK LAYOUT

MACPITTS TARGET ARCHITECTURE



CONTROL STRUCTURE IMPLEMENTED AS FINITE STATE MACHINE
DATA STORED AND MODIFIED IN DATA-PATH UNIT
INDIVIDUAL COMPONENTS CONNECTED USING A CHANNEL ROUTER

DATA PATH GENERATION

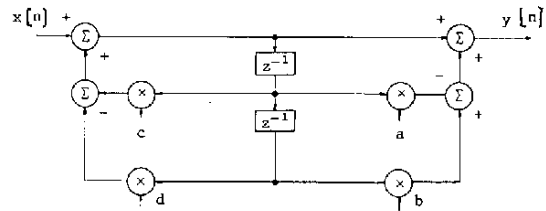


CONVENTIONAL APPROACH

MACPITTS APPROACH

SOUND ORDER RECURSIVE BI-QUADRATIC FILTER

$$H(z) = \frac{1 - az^{-1} + bz^{-2}}{1 - cz^{-1} + dz^{-2}}$$



4 multipliers
4 adders

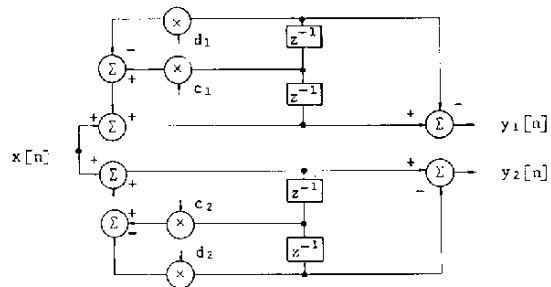
2 registers
6 I/O pins

THEMES THROUGHOUT

1. REPRESENTATIONS KEYED TO PARTICULAR DESIGN FOCI
2. TRANSFORMATIONS BETWEEN REPRESENTATIONS WHICH PRESERVE ALGORITHMIC INTENT.
3. USER-ORIENTED INTERACTIVE LANGUAGES WITH APPROPRIATE DEBUGGING AND GRAPHICS.
4. TOOLS TO DEAL WITH COMPLEXITY, INCLUDING FACILITIES TO NATURALLY HANDLE MODULARITY AND HIERARCHY.

FILTER BANK SEGMENT

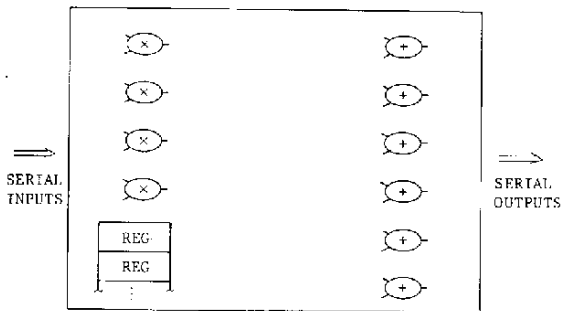
$$H(z) = \frac{1 - z^{-2}}{1 - cz^{-1} + dz^{-2}}$$



4 multipliers
6 adders

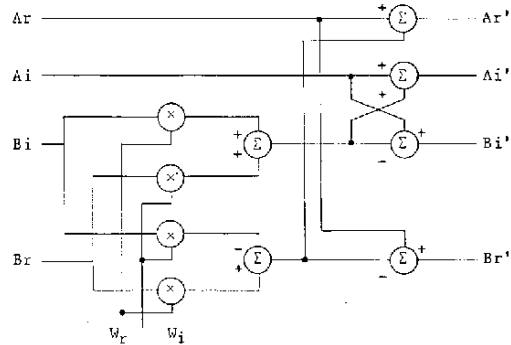
4 registers
7 I/O pins

BASIC CELL CONFIGURATION

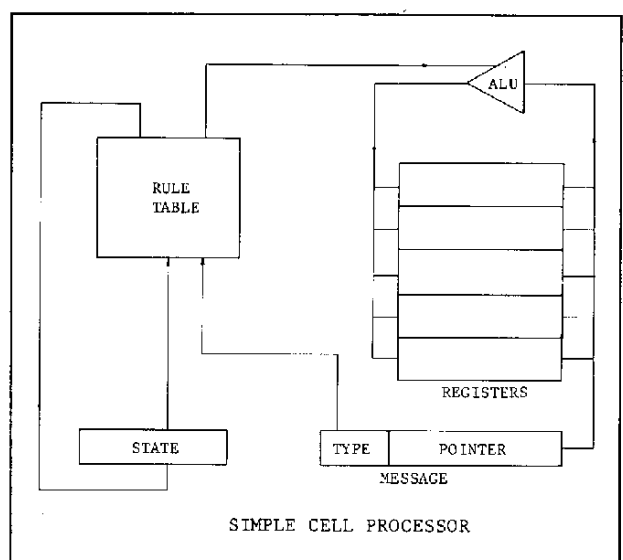
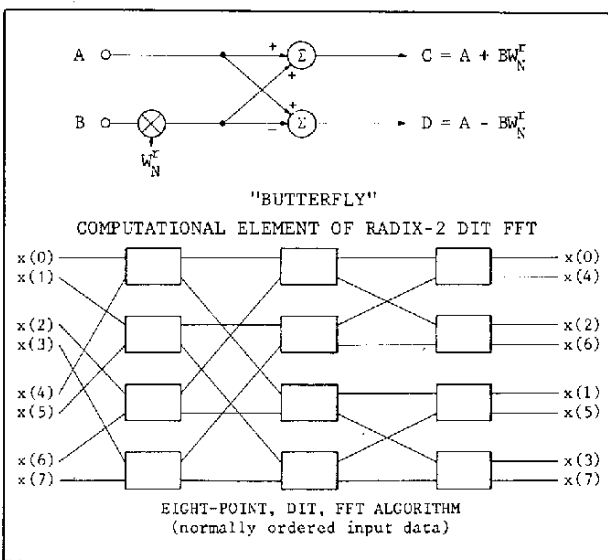
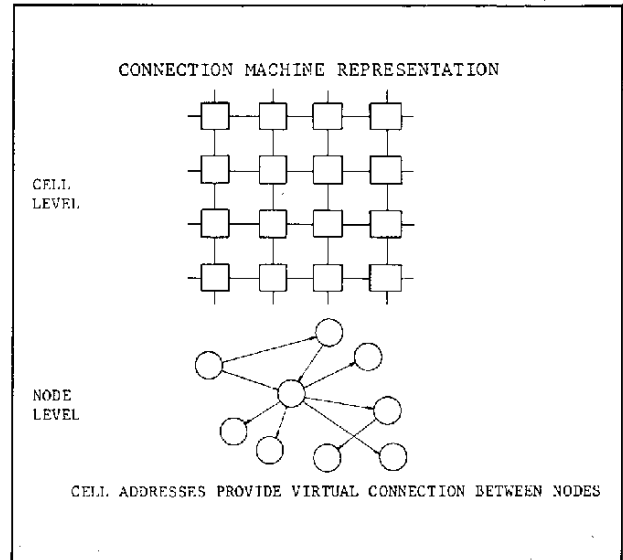
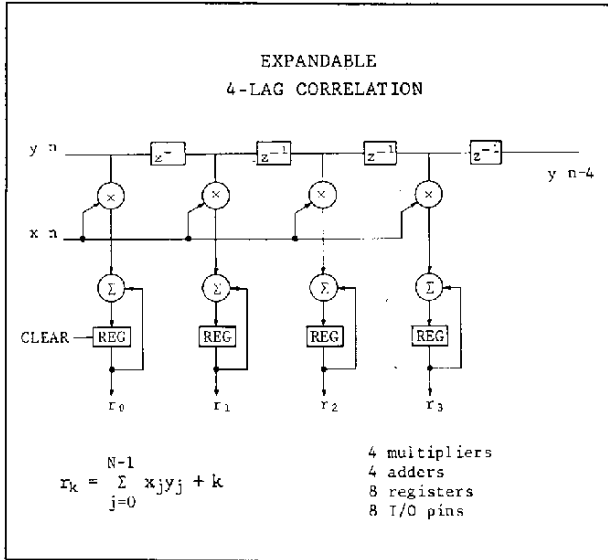


- 4 16 x 16 multipliers
- 6 16+ bit adder/subtractors
- Small number of 16+ bit registers
- Serial arithmetic (10-20 MHz clocks)
- Serial I/O

RADIX 2 DECIMATION IN TIME FFT COMPUTATIONAL ELEMENT



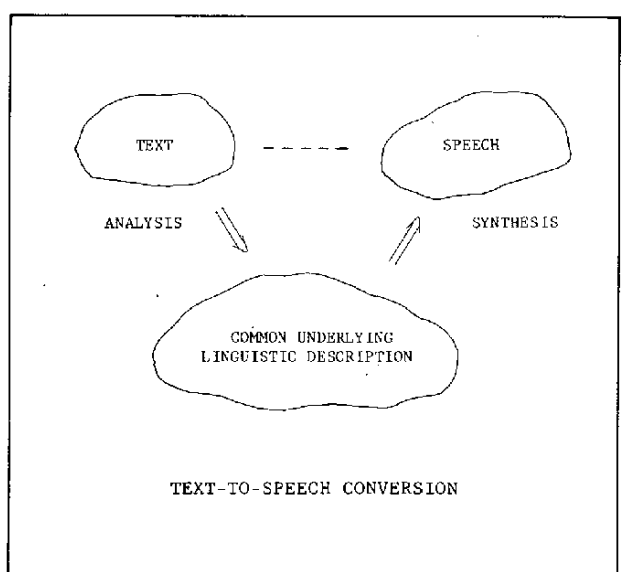
4 multipliers 6 adders 10 I/O pins



CONCURRENT MANIPULATION OF KNOWLEDGE IN SEMANTIC NETWORKS

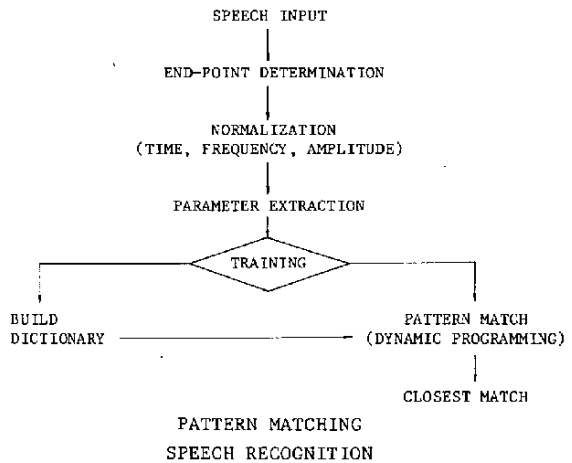
A FEW SIMPLE OPERATIONS:

- DEDUCE FACTS FROM SEMANTIC INHERITANCE NETWORKS
- MATCH PATTERNS AGAINST ASSERTIONS OR PRODUCTIONS
- SORT A SET ACCORDING TO SOME PARAMETER
- SEARCH GRAPHS FOR SUB-GRAPHS WITH A SPECIFIED STRUCTURE



TEXT ANALYSIS:

1. MORPHOLOGICAL ANALYSIS
FORMALLY + FORM + AL + LY
2. LETTER-TO-SOUND RULES
JAPANESE
3. PARSING
(COMPUTER SCIENTISTS) (LIKE) (SUSHI.)
(IN THE EVENING.)
4. STRESS RULES
ENGINEER, HUMANITY
5. MORPHOPHONEMIC RULES
PLURAL, PAST, PALATALIZATION
6. LOW-LEVEL PHONETIC ADJUSTMENTS
FLAPPING, CENTRALIZATION, NASALIZATION



SPEECH SYNTHESIS:

1. TEMPORAL FRAMEWORK
PAUSES, VOWEL DURATION, CLUSTERS
STRESS
2. PITCH CONTOUR
DECLARATIVES, QUESTIONS
SEGMENTAL EFFECTS, MODALITY
3. PHONEME TARGETS;
20 PARAMETERS PER PHONEME
4. CONTEXTUAL SMOOTHING
SIMULATES COARTICULATION
5. PARAMETER-TO-COEFFICIENT CONVERSION
FORMAT FOR SYNTHESIZER
6. DIGITAL SYNTHESIS
USES VOCAL TRACT MODEL.

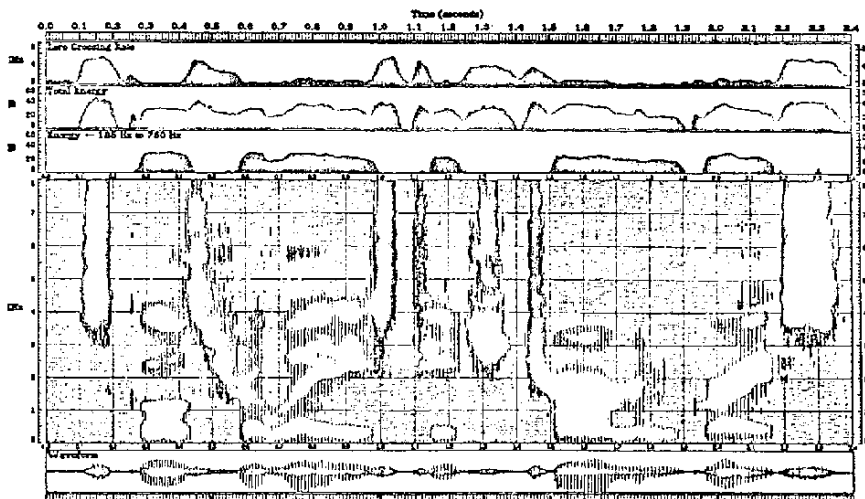
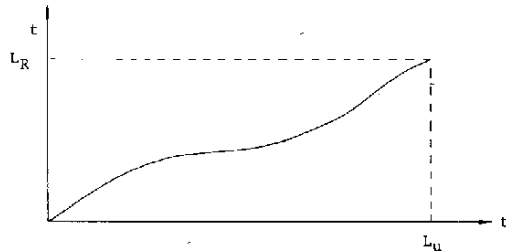
DYNAMIC TIME WARPING

$$d_{ij} = \sum_{n=1}^p (R_{jn} - U_{in})^2$$

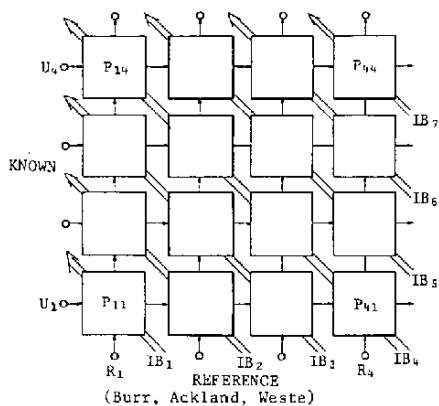
$$D_{ij} = d_{ij} + \min^*(D_{i-1,j}, D_{i,j-1}, D_{i-1,j-1})$$

$$0 < i < L_R \quad 0 < j < L_U$$

*subject to slope constraint

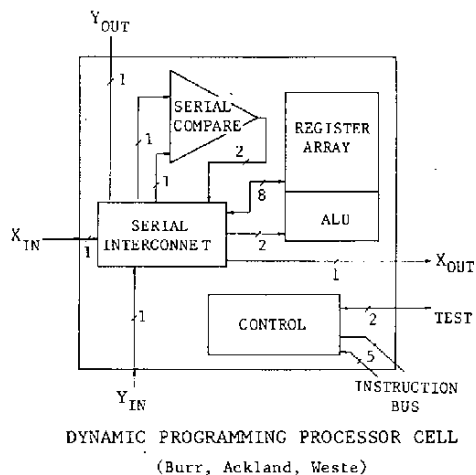


4 x 4
DYNAMIC PROGRAMMING PROCESSOR ARRAY



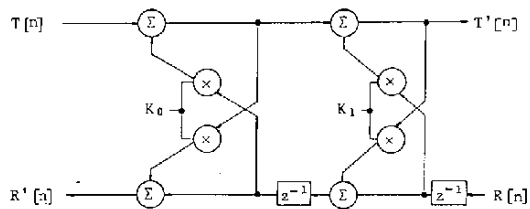
INTEGRATION
OF
PHONETIC CUES
FOR
VOICING

- use (NOUN & VERB)
- rapid/rabid 16 correlates!
- voiced/unvoiced/silence distinction
maximum likelihood over
6 parameters
- stop closure time vs. rate of
articulation.



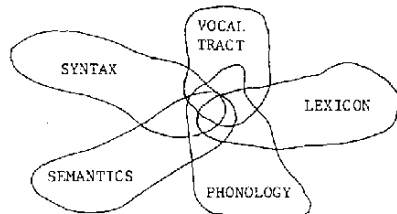
DYNAMIC PROGRAMMING PROCESSOR CELL
(Burr, Ackland, Weste)

EXPANDABLE
FOURTH ORDER LATTICE FILTER



- 4 multipliers
- 4 adders
- 2 registers
- 6 I/O pins

LANGUAGE
AS A SET OF
MUTUALLY INTERACTING CONSTRAINTS

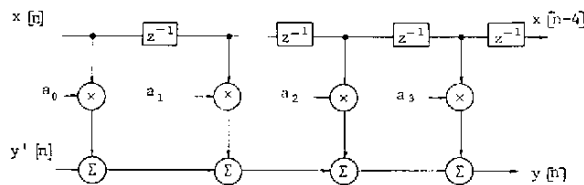


- EACH SYSTEM HAS A SUBSTANTIAL INTERNAL THEORY
- BUT ALSO A SIGNIFICANT INTERACTION WITH OTHER CONSTRAINT SYSTEMS
- SYNTAX & PHONOLOGY
NOUN/VERB AMBIGUITIES: "INCLINE"
- SYNTAX & SEMANTICS
"HE DROVE DOWN THE STREET IN THE CAR"
- MORPHOLOGY & PHONOLOGY
PLURAL, PAST, PALATALIZATION

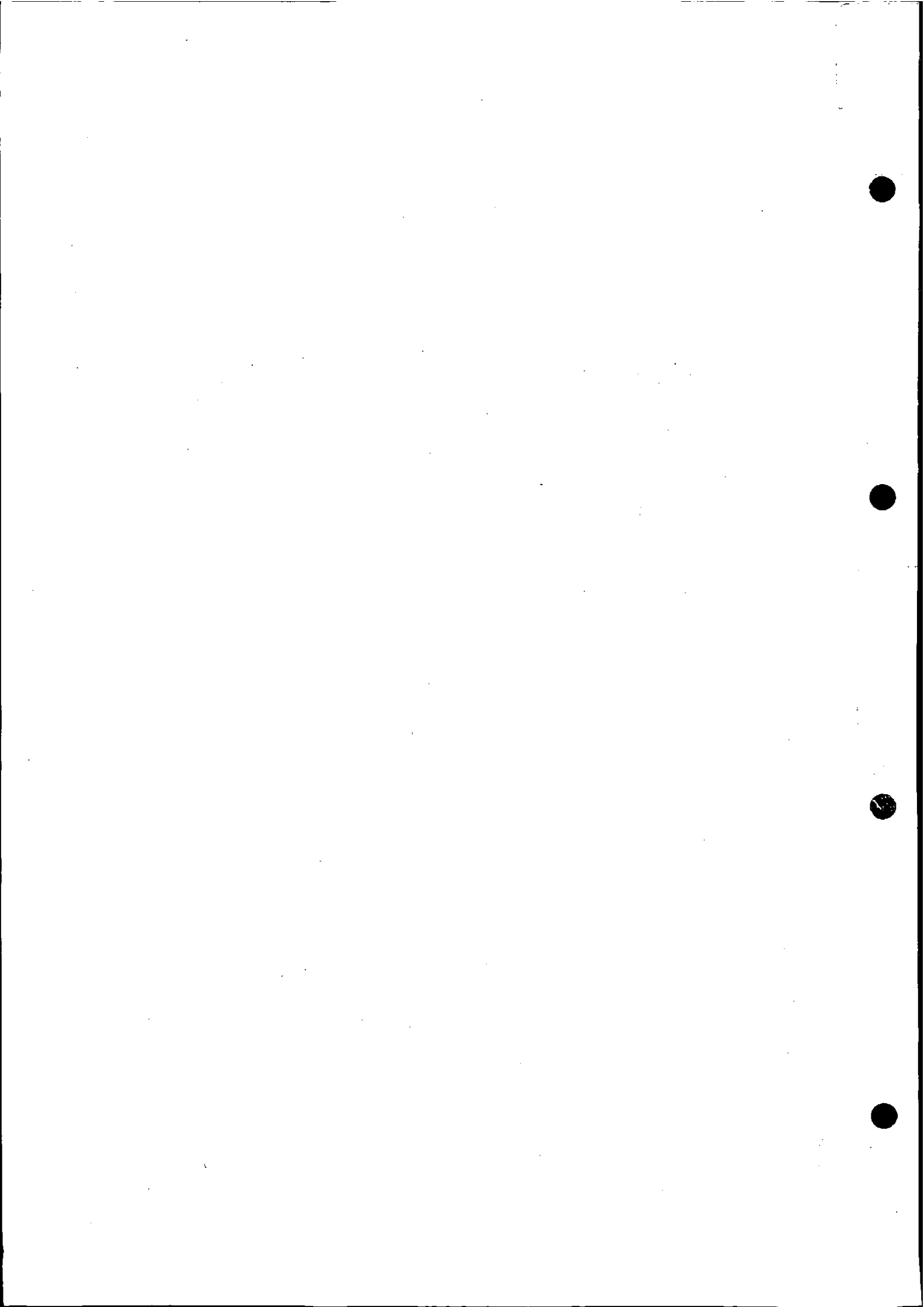
EXPANDABLE FIR FILTER

$$N = 4$$

$$H(z) = \sum_{j=0}^3 a_j z^{-j}$$



- 4 multipliers
- 4 adders
- 4 registers
- 8 I/O pins



—— 禁無断轉載 ——

昭和 57 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 3-5-8

機械振興会館内

TEL (434) 8211 (代表)

印刷所 山陽株式会社

TEL (591) 0248

