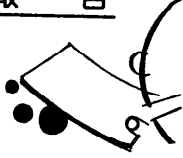


報 告



パネル討論会

第5世代コンピュータ・スーパーコンピュータが商用コンピュータに与えるインパクトについて†

パネリスト

鈴木 則久¹⁾, 林 弘²⁾, 竹内 郁雄³⁾
山本 昌弘⁴⁾, 司会 相磯 秀夫⁵⁾

司会 皆さんこんにちは。それではパネル討論始めさせていただきます。このパネル討論の課題は第5世代コンピュータ、あるいはスーパーコンピュータのようなものが、商用コンピュータにどんなインパクトを与えているか、という題で仰せつかりました。この課題は飯塚委員長並びに坂村、大島幹事のもとで設定していただいたものです。

最初に壇上におられますパネリストの皆さん方をご紹介します。

1番左側は東京大学工学部計数工学科助教授でおられます鈴木則久先生。鈴木先生は昭和44年、東京大学応用物理、物理工学科をご卒業になり、昭和46年東京大学の電子工学で修士の学位をお取りになりました。その後、米国のスタンフォード大学に留学をされまして、昭和50年にスタンフォード大学計算機科学科で工学博士(Ph. D.)の学位をお取りになっています。その年から2年間カーネギーメロン大学、コンピュータ・サイエンス学科の助教授をお務めになり、昭和52年から約5年間、ゼロックスのパロアルト・リサーチ・センターで研究をされております。その後、昭和57年に東京大学におもどりになりまして、現在助教授をお務めになっております。

そのお隣りは富士通研究所の林弘さん。林さんは昭和42年、東京大学の電気工学科をご卒業後、富士通研究所にお務めになっております。現在はリスプマシンなど高級言語マシン、あるいは並列推論マシンの研究を担当されております。ICOTの研究などにもずいぶん分与されております。

3人目の方はNTT 武蔵野通信研究所からおいでの竹内郁雄さん。竹内さんは昭和44年、東京大学理学部数学科をご卒業になり、昭和46年に大学院で修士の学位をお取りになりました。その後、武蔵野通信研究所基礎研究部にお入りになり、主に人工知能用のシステム・ソフトウェアの研究をされました。現在は情報通信基礎研究部に所属をされておられます。

私の隣りにおられます4人目のパネリストは、日本電気の中央研究所にご勤務の山本昌弘さん。山本さんは大阪大学工学部精密工学科をご卒業になり、日本電気中央研究所にお入りになり、高級言語マシンの研究開発に従事されています。現在は、C&Cシステム研究所・コンピュータ・システム研究部に所属され、知識処理関連の研究、知識処理マシン、知識処理用の高級言語、あるいは知識処理応用などを研究されております。以上が壇上におられますパネリストの方のご略歴であります。

私は司会を務めます慶応大学、相磯秀夫です。昭和30年に慶応大学を出まして、32年に修士、そして大阪大学工学部を経て通産省の電気試験所、今の電子技術総合研究所に入りました。昭和46年に慶応大学のほうにもどり、現在に至っております。

きょうのパネルは、もう、十分時間をとっていただきましたから、気楽に、大いに皆さんの意見の交換をしたいと考えております。ですから、壇上におられる方は、メインのパネリストで、会場の全員がパネリストだと思っております。全員参加のパネル討論だと思っておりますので、どうぞ皆さんも気楽に、壇上におられるパネリストにひけを取らずに、ご意見を述べていただきたいと考えております。

パネルのやり方といたしましては、やはりご準備いただきました4人の方に、それぞれ20分ぐらいずつ、

†日時 昭和59年11月28日の「アキテクチャワークショップ インジャパン '84シンポジウム」

場所 機械振興会館大ホール(地下2階)

1) 東大, 2) 富士通研, 3) 武通研, 4) 日電, 5) 慶大

まずご自分の見解を述べていただきまして、それをベースにしてご質問、あるいは意見の交換ということにしたいと思ってます。それでは席の順ということで、鈴木先生からお願いします。

鈴木 パネルの題は、スーパーコンピュータおよび第5世代コンピュータが商用コンピュータにどのような影響を与えるかということとして、なかなかむずかしい題だったんですけど、ぼくがあまり知らないほうから言いますと、スーパーコンピュータのプロジェクトについては一切知りません。私の知る限り、スーパーコンピュータというのは大規模科学計算に使われ、核ですとか、構造解析など政府関係が使うんだらうと思っています。最も問題なのは、貿易摩擦かなということでお茶をにごしまして、もっと興味のある第5世代コンピュータはどうなるかということですね。

第5世代コンピュータの定義は、1つは ICOT の開発するコンピュータと、もう1つは人工知能用のコンピュータですね。まず ICOT の開発するコンピュータからお話しましょう。ICOT が今まで開発したのは、データベースマシン、PSI と、それから第2期に作られるであろうパラレルマシンというのが私が知ってる限りです。委員会に属してないので、どういう仕事をしているか知らないんですけど、いろんな資料ですとか会議で聞いた情報を集めると次のようになります。

まずデータベースマシン。私は根本的にアーキテクチャが間違っていたんじゃないかと思っています。話に聞くとところによると大変遅い。メインフレームと比べて100分の1ほどのスピードだそうです。あるメーカの方、部長さんから2年前に呼ばれて、うちの会社は第5世代計算機でなにをやったらよろしいでしょうか、と聞かれました。データベースマシンだけはやめておいたほうがいいと言いました。それが正解だったんじゃないかと思っています。

私は2年ほどリレショナル・データベースのソフトウェアをこつこつ作りまして、経験がかなりありますが、要するに現在のアーキテクチャ屋さんは、Sortとか、Joinのハードウェアをつけて速くしようとするんですけども、本当のデータベースのボトルネックは、ストリングの Compare ですとか、Tuple をアクセスしますと、それを1つのページの中で動かしたりガーベッジ・コレクションしなければならぬ。そういうメモリマネジが問題になりますね。要するに I/O よりも、CPU の中にデータを読み込んでからページ

内のアクセスがボトルネックになっている。ここを速くしなければ、いくら Join を速くしてもシステムの性能には一切ひびかないということですね。と言うことはメインフレームを持ってきて、そのマイクロコードを書き替えるとかかなり速いシステムができる。また大規模キャッシュを持ってきて、ページ内の計算をする。データベース・システムをご研究になったらわかると思うんですけど、データベースの重要なデータ構造には Tuple というものがある。この Tuple が各レコード・データを含んでいるのですが、それにはユニークな名前が付いている。またレコードの中のデータは非常に間接参照が多いんで、そのアクセスを速くするためのキャッシュを作るとかいうのが一番いい。Join を速くするというのはまだまだ当分先のことなんで、だから商用計算機としても、Join をハードウェアでやるデータベースマシンが出るということは当分先であろうと思います。

次は PSI ですが、これはいろいろな評価がありまして、特にアメリカ人なんかに言わせると、PSI を見るとまた新しい LISP マシンができたということで、アーキテクチャ的には Symbolics 3600 にそっくりですね。PSI が他のマシンと違うところは、初めて本格的なハードウェア・メーカが作ったこと。ということは今までの LISP マシンはベンチャーか Xerox というコンピュータ・メーカじゃないメーカが作っていたんで、信頼性に問題がありました。

しかし PSI の場合はスピード問題ですね。PSI は今、30 K LIPS だそうです。マイクロコードを書き替えれば、もっと速くなる可能性もあると思いますけれども、これは 68020 上の Prolog よりも、かなり遅くなるんじゃないかということが懸念されますね。David Warren が会社を作って、Quintas Prolog というのを売り出しましたが、これが非常に速いんです。ソフトウェアというのは1人の天才が作ったものが、世界中に簡単にいきわたります。ところがハードウェアというのは、非常にたくさんの人がコピーを作らなきゃならないんで、そういうことはあり得ない。Warren の作った Prolog は、今 68010 の上で 20K LIPS で動いています。それから DEC で出そうとする Micro VAX-II、これは VAX 780 と同程度のスピードですが、この上だと 30K LIPS まで出る。68020 がもうすぐ出ますが、これは 68010 の4倍までいくそうです。これが出てくると、Quintas Propog は、いっぺんに、80 K LIPS になります。そういうわけで、

PSI が 68020 より良い点はちょっと見付からない。

一番明るい点は ESP と、SIMPOS です。これは今 PSI の上でしか動かないので、ESP を使おうという人は PSI を買わなければならない。

次はパラレルマシンがものになって、商用マシンが出てくるという問題です。いろいろ有効であるという論文が出てるんですけども、それを見るとシラミツブシ問題についてのみ、これは非常に速くできる。たとえば 8 Queens プログラムで、全部の解を求めるのに、シングルマシンより速くできる。しかし今、人工知能で問題なのは、まず 1 つなにか解を求めてくださいというのが問題でして、そういうことになるとパラレルマシンを持ってきても、シングル CPU より速くならない。Prolog の場合は、1 つだけの解を求めるときには、パラレル・コンピュータを使っても、せいぜい 4 台ぐらいが性能向上限度で、これ以上並列性を高めても速くはならないだろうと言われてますから、これは第 2 期の非常にむずかしい課題になるんじゃないかと思えます。ですから ICOT の開発するコンピュータ、特に PSI が世の中で大いに使われるかということについては、かなり疑問が多い。値段も高いし、スピードも遅いということですね。

しかし人工知能用コンピュータに関すると、1985 年は人工知能用ワークステーションの大躍進の年だと思えます。そのまずはしりとして、1984 年の後半に 2 つの新しいコンピュータが出てきた。1 つは Texas Instruments 社が新しい LISP マシン (Explorer) を開発しました。これは Symbolics 3600 の半分の手軽な速度ですけど、3分の1以下の値段ということで MIT が Explorer を 400 台買ったということ。

それから Tektronix 社の 4404 というのが出ました。これは Smalltalk 専用です。CPU は 68010 ですからなんでも動くんですけど、Smalltalk がもたなくなってまして、Xerox 1100 の 2 倍の速度で 4分の1の値段ということですよ。

85 年はこの路線上で、大メーカー、DEC とか IBM が人工知能用コンピュータは非常に大きなマーケットになると思って出てくると思います。特にパソコンで本格的に動く LISP、Prolog とか Smalltalk が出てくる。現在パソコン・ブームは 1 段落しましたけれども、今後は人工知能用コンピュータ・ブームというのが 1985 年からおこると思えます。

なぜこうなるかと言うと、1 つはソフトウェアの技術的進歩と、もう 1 つは、68000 ファミリのマイクロ

プロセッサの性能が爆発的にのびるだろうということです。

私はこの三言語の中で Smalltalk に最も詳しいんで、速度比較をします。いろいろ Smalltalk のシステムが作られました。最初にできたのは Dorado と Dolphin で、Dorado は、ECL で作られた。マイクロコードはかなり昔に書かれたもので、今これを書き替えれば、この倍は少なくとも出ると思うんですけど、Smalltalk のバイトコードを 1 秒間に 25 万個実行できます。

一方日本で手に入る Dolphin は、毎秒 2 万 5000 バイトコード実行します。ところが最近いろいろな所で 68010 上でいろいろ苦労して作りまして、Dorado にも迫る勢いのもが出てきている。50 N ワークステーション上で Xerox 社の Peter Deutsch が作ったのが、1 秒間に 10 万バイトコード実行し、68010 で現在の Dolphin の倍の速度を出す。

Tektronix 4404 というのが最近発表になりましたが、仕事は 1980 年からやってる。5 年かかったわけですね。しかし、こういうソフトウェアには 5 年もかけても大丈夫だということです。68000 のように寿命の長いアーキテクチャの上で動くソフトウェアというのは、いくら長くかけても大丈夫です。ハードウェアというのはどんどん変わるから、あまりマイクロコードを最適化していても始まらないんですけども、68000 上で非常に苦労していいソフトウェアを書いておいておくと、日立とか Motorola 社とかが 68000 のスピードをどんどん上げてくれます。ということで、黙ってただけで 2 倍、4 倍とどんどん速くなる。

われわれの所で菊 32 というのを作ってまして、現在 1 秒間に 4 万バイトコード実行できます。これは 32 ビットのオブジェクト・ポインタを持ったインタプリタとしては世界で一番速い。Tektronix 4404 はいまだ 16 ビットです。ということで 68000 という汎用マイクロプロセッサにソフトウェアを書くという仕事、こういうように専用マシンを作る仕事とかなり厳しい競争になってきています。特に値段の安いマシンについては汎用機がいい。また、ソフトウェアの数から言っても、汎用機がまだ非常にいいだろうと思えます。

このようにたくさん人工知能ワークステーションが出てきますと、どれを選ぶのかと選択に迷われるわけですね。選択となると、値段と速度とソフトウェアの数ということになります。ソフトウェアの数がどのく

ら出てくるか、たとえば人工知能用語で今、主に言われているのは LISP, Prolog, Smalltalk ですが、たとえば Symbolics 3600 だと LISP しか動かない。

一方 Tektronix になると、その上に UNIX がついてくる。UNIX は、今まで人工知能ではそんなに使われていないかもしれないけれども、これから差が出てくると思うのは UNIX の上のソフトウェアの量というのは莫大なわけです。特に基本的な計算、リストプロセッシングですとか、ネットワークだとかいろいろなソフトウェアが揃ってる。Tektronix 4404 というのは非常に短期間で開発して出したんですけど、そういうときにネットワークは Smalltalk で書いてない。みな UNIX で書いてあるパッケージを買ってきて乗せただけでできてしまう。また Smalltalk の上から UNIX を呼べますし、Prolog の上からも読めるということで、UNIX の重要性が非常に増してくる。一方 PSI は Prolog だけ、SUN2 は LISP も Prolog も Smalltalk も UNIX もあるということで、こういうところから見ても Symbolics 3600 とか、Dolphin とか、PSI は、商用計算機としては、かなり厳しい環境になるだろうと思います。

結論を言いますと人工知能システムというのは、これからいろいろな分野で重要になります。皆さん大いにソフトウェアを開発してください。しかしながら、使うシステムとしては高速に実行するときはメインフレーム、中低速になると 68000 ベースのワークステーションだとかパソコンが主流になるだろうということです。

司会 どうもありがとうございました。第 5 世代コンピュータの研究開発には 2 つのポイントがある。1 つは ICOT で開発をしているコンピュータ、それからもう 1 つは一般的に解釈した AI コンピュータ、その 2 つに関して、鈴木先生のご見解がありました。ICOT の研究は 1 つの成果ではあるけれども、回りの技術の進歩が急速で、その辺をよく考える必要があると指摘されています。Prolog のユーザがどのくらいあるかというのも 1 つの問題点になっているように思われます。もっと一般的には、AI の分野というのは大変重要で相当いろいろなコンピュータが、特に米国から AI マシンという形で出てくると予想されています。

しかしいろいろ考えてみると、高速用マシンはメインフレーム、中低速用は M68000 のような、割合に将来性のあるアーキテクチャの上で作った方が賢明のよ

うです。ソフトウェアはハードウェア上で日に日に早くなってくるという事実を意識して、ソフトウェアにむしろ時間をかけるべきではないだろうか。特に AI のアプリケーションをよく考えて、むしろソフトウェアの研究ということに大いに時間をかけたほうがいいんじゃないか、というようなご意見だったように思われます。

さて、次に林さんをお願いいたします。

林 今回、相磯先生のほうからパネリストとして指名されて、一番困ったのはこのテーマの名前です。第 5 世代コンピュータ、あるいはスーパーコンピュータが与えるインパクトということなんですが、土台ができてないものをどうやってそれが言えるか、もともとできるかどうか分からないときにインパクトがあり得るだろうかというのが、基本的な私の感じですが。

第 5 世代コンピュータのイメージということで私自身、あるいは他の人もある程度言ってる部分というのはかなりあるわけです。大きく分けて言うと 3 つあるようです。1 つは全体システムということで、これは先程の鈴木先生の言われた人工知能システムとか、それとからむと思いますが、いわゆる現在できてないところをうまく処理するソフトウェア能力です。それから ICOT の PSI 関係を直接やっておられる人が特に言われるプログラム開発環境の問題です。

それからあとはハードウェアの問題に関しては、一つは基本的に高速化ではないかという気がします。ここではこれについて、私の考えてるところを少しアーキテクチャ的に述べてみたいと思います。

高速化ということに関して、今回のワークショップでも一般的だった話ですが、並列化すればスピードが上がると皆さん簡単におっしゃるわけなんです。じゃあ本当に並列化できるかどうかということになるとデータ、特に画像とかそういうデータについてもともと並列化すれば速度が上がるという分野は当然であると思うんですが、それ以外の所で本当に並列化できるかどうかということが、はっきりと実証されてない。ただ、できたらいいなということ、できるということと勘違いされてる部分があるんじゃないかと私は思ってます。富士通の大型コンピュータの性能の推移を見るとリレーから現在は LSI までいってるんですが、これはゲート当たりの遷移時間でいくと 10^7 倍ぐらい、簡単に上がってるわけです。片方はミリ秒で、片方はピコ秒までいってるわけです。ところが、この 10 年間を見ますと、システム

の性能は10倍も上がっていないわけです。富士通のマシンで見ますとM190というのが1975年くらいに出荷されました。M380が1983年あたりに出荷されました。この間大体10年なのですが、性能は1桁向上していません。

そうすると今並列で、いろいろ皆さん討論されているのは、最低でも10倍、もっといけば100倍とか、1000倍という値が対象となっている。本当に100倍、1000倍を出せるのかというのが私の疑問です。

これが、今の第5世代コンピュータの1つの問題です。ですから今日のテーマというのも、第5世代コンピュータが与えるインパクトじゃなくて、第5世代コンピュータをどうやって作るかというのが一番問題ではないかと思っています。

あと専用マシンということで、鈴木先生も言われたんですが、少しコメントしたいと思います。

私、この5年間LISPマシン「FACOM- α 」を富士通研究所で研究してきましたが、その間の経緯と、それに基づくデータで少しお話してみたいと思います。

表-1はわれわれが作り出したLISPマシンFACOM- α のハードウェアの使用頻度でございます。

このデータをもとにして、 α と汎用機のインタプリタにおける性能比較をしました。結果的に、汎用機に比べて α の速度が6倍向上しました。

ご存じのように、汎用機では当然コンパイラが存在します。コンパイラで比較したら、専用機の場合、当然条件はきびしくなるわけです。インタプリタでせいぜい6倍だったものが、コンパイラで比較するとどうなるか、つまり6倍より下ということは目に見えているわけです。

こういうことは先程から鈴木先生も言われているんですが、本当に専用機というのは必要かどうか、と言うか、生きのびられるかどうか、つまり本当に将来的に残り得るか、十分検討すべき項目だと思います。

また補足ということになると思いますが、表-2はPrologとLISPの性能を α の上で比較したときのデータです。右のほうががLISPの性能です。LISPでNREVERSE 30とQUICKSORTを動かしました。左のほうはPrologのデータです。

かっこの中はコンパイラの性能です。NREVERSEに関してみると、 α 上のPrologのインタプリタだと、40msのLISPのインタプリタだとこれが41msです。 α はもともとPrologというのを意識してませんでLISP専用で作ったマシンです。ところがマイクロコードを細工すれば似たような性能が出る。Prologというのは従来LISPに比べて非常に遅いと言われてたんですが、ハードウェアでインプリメントするとそうでもない。

このデータから2つ言えると思います。1つは専用機と言っても、本当に記号処理専用機としてアーキテクチャがあり得るか。つまりどういう言語を持ってきても、それなりに性能は出るし、アーキテクチャとは

表-1 ハードウェアの使用頻度

		SORT-100	TARAI-3	TPU-1
各ハードウェア使用頻度	スタックアクセス	0.42	0.41	0.46
	スタックポインタ自動増減	0.24	0.25	0.29
	データタイプ判定	0.09	0.10	0.07
	分岐 ※1	0.19	0.17	0.17
	メモリアクセス	0.14	0.15	0.12
全走行ステップ数		4200000	270000	3900000
実行時間 (μ s)		710000	44000	640000
平均命令実行時間 (μ s)		0.17	0.16	0.16

※1 無条件分岐および条件分岐成功のステップ数
インタプリタ

表-2 PROLOG/LISP 性能

インタプリタ
単位 ms (コンパイラ)

	PROLOG			LISP		
	NREV 30	QSORT 50	8 Q	NREV 30	QSORT 50	8 Q
FACOM α	α -PROLOG			UTILISP		
	40 (16)	58 (18)	7,561 (2,829)	41 (9)	51 (11)	4,028 (1,570)
M180 IIAD	PROLOG/KR			UTILISP		
	344	433	48,808	34	44 (4)	3,286 (1,011)
VAX 11/780	C-PROLOG			FRANZLISP		
	330	502	114,400	684 (106)	918 (130)	80,160 (52,520)
DEC 2060	DEC 10-PROLOG					
	191 (9.5)	235 (14.7)	—			

あまり関係なしに性能が出るのではないか。また、Lisp と Prolog というのは結構性能的に似たところに行くし、従来から言われているように Prolog は必ずしも遅くない。

これが一般的にあてはまるかどうか、まだちょっとわかりませんが、私としてはこういうような、全体的な第5世代からみているいろいろな問題があるのではないかと考えています。これが実際の商用機に対して、どういうインパクトを与えるかというのは、はっきりよくわかりません。以上です。

司会 ありがとうございます。第5世代コンピュータの研究の意義は大きく分けて3つある。1つはシステム全体から見れば今までのコンピュータが得意であったところを、どうやって改善するか。もう1つはソフトウェアの生産性を上げるという意味もあって、プログラミング環境を整えるというような観点。3つめは処理効率を上げるというような目的があるのではないだろうかとのことです。特に林さんの立場では、過去5年間のリスプマシンのご研究から見て高速化という観点を示していただいたわけですが、意外と専用マシンでもスピードが思うように上がらないというようなお話だったように聞こえました。また、Prolog 自身が、どのような意味を持つか、そしてそれがこれからどのように商用コンピュータにインパクトを与えるかは未知数というようなお話であったと思います。

それでは竹内さんをお願いいたします。

竹内 これは聞くところによりますとアーキテクチャのワークショップです。私のごとき人間がこういう所にこのこと出てくるのは、まことにおかしい話です。恥をかけたということだと思いますが、幸い恥をかくのにはなれておりますから、ちょっとここでアホなことを言わせていただきます。

実はこの題名に非常に悩みました。これを聞いたときに、回りの人に、「インパクトある？」と聞いたら、ほとんど全員「ない」と言いました。私も「ない」と思ったんですが、なんとかしてあることにしなければならぬと思ったのが、このパネルの私に対するインパクトでありました。いろいろ考えたんですけども、やっぱりどうもなさそうですね。なさそうというのは、今、Prolog とか Lisp とか、Smalltalk とかにたくさんの方が飛びついていますが、最終需要として本当に実体のあるものに皆が飛びついてるのかどうか不安だからです。

私も実はそういう関係のことをやってるわけであり

ますが、本音は囲碁のプログラムさえ作れば良いと思ってるので、商用になるかどうかあまり関係ないわけです。人工知能というものに向けてなにか工作機械みたいなものがどんどん作られて、それが一見売れ筋風なので人が騒いでいるわけですが、工作機械はしよせん工作機械でして、末端の消費製品、つまり、チョコレートとかプラモデルとかができないと、最終的な需要と結びつきません。この点が今だに心配です。

そうはいつでも人工知能言語など不要、メインフレームで Fortran でやってしまえば終わりという結論が出てしまうのは悲しい。そうならないためやっぱりがんばらないといけないと思うんですが。

ところで、日本の研究のソサイエティでは、システム作りばかりやって、最終的なアプリケーションに優秀な人がかからないような気がします。

アメリカはネットワークが盛んなせいもあるでしょうが、1つ文化の種ができます。それが広がるのが速いですし、必ずアプリケーション指向の人がいまして、人の作った土俵の上で仕事してくれます。どうも日本ではこれが弱い。あいつの作ったものの上では絶対仕事をしないといった感じのセクショナリズムが盛んでありまして、どうも調子悪いですね。いろいろな研究の努力が相互に無関係に分散されて、まさに分散処理になっています。分散処理の実効が上がらないのと同じで、どうも日本全体のアウトプットが少ないという気がしています。またこのセクショナリズムの話とは別に、縦の階層でのつながりの悪さがあります。デバイス、アーキテクチャ、ソフトウェア、なにかそのあいだにも一種のセクショナリズムが匂います。私の印象からするとアーキテクチャ屋さんが自律的に動いて、ソフトウェア屋さんの都合を考えずに、いろいろおもしろいものをたくさん作っている。こんな感じですね。ですから全体でつながりのあるシステムが一向にできてこない。

予稿の中に、今はアーキテクチャのルネッサンスと書いたんですが、これは裏を返しますと、第5世代コンピュータというのが話題になったお陰で、今まで失業しかけていたアーキテクチャ屋さんが、また職を持ち始めたということです。しかし、そのアーキテクチャ屋さんに、強力なソフトウェア屋さんがなかなかついてないような気がします。

ですから、まとまったシステムがなかなかできない。世の中にインパクトを与えるという意味では、紙の上に書いただけのアーキテクチャじゃ全然話になり

ませんし、なにか動いてランプがちかちか動くだけでもだめです。なにか入れたらおもしろい結果が出るとか、どうもこれ使えそうだとか、コストも安そうだとか言えるところまでこないと商用という意味ではインパクトは与えられないと思うんですが、なかなかそこまでの努力がなされていないような気がします。研究者の方はどんどん目移りするもんですから、1つのマシンの上で10年くらい正座をして仕事をするという人がなかなかいません。

私は前からコンピュータというのは、1つの文化だと思っています。だからコミュニティができれば絶対だめだと思うんですね。たとえば IBM 文化、DEC 文化、UNIX 文化、Lisp 文化、どれをとってもそれを支えるコミュニティがあります。Prolog 文化というのもできつつあるんでしょうかね。文化の成長にとってその文化のコミュニティ、つまり母体が社会から無視し得なくなるくらいの厚みと大きくなるということが重要だと思います。アーキテックチャー屋さんにはソフトウェア屋と組んでコミュニティを作ろうという感覚が、少し欠けてるんじゃないかと僭越ながら思ったりしています。

先程専用機と、汎用機の話がありました。われわれはまだ正式には発表していませんが、ELIS という専用マシンを開発しています。その上に TAO という、Lisp と Smalltalk と Prolog、さらに Fortran、ついでに C も取り入れた気遣いみたいな言語を乗せようとしています。それを見る限り、先程林さんのほうから6倍という話がありましたけれども、試算では10倍上がったと思っております。汎用機のほうでは、Cray の上に Lisp を作っている人間が研究室におります。どれくらい速い Lisp ができるか、非常に興味深いんですけど、CAD とか、ものすごい応用を Lisp で書きたいと思ったら Cray の上でやることになるかもしれない。

われわれの専用機のほうは、性能もさることながらコストダウンを1番気にしております。CPU の VLSI 化が第1目標です。回路規模は16K ゲートそこそこですから、現在のテクノロジーでそんなにむずかしいものではないと思います。他の専用マシンは、もっとゲート数が大きいですが、われわれのは非常に単純で、いわゆる専用ハードウェアというものはほとんどついていません。かえてそれがよかったのかもしれない。その代りマイクロプログラムは非常に大きくて、単一言語と言うには大きすぎる感じの言語ですけ

ど、単一言語のためのファームウェアとしてはいまのところ一番大きいんじゃないかと思っています。

現在は23K ワードですが、今年度末ぐらいには、大体32K ワード目一杯ぐらいまでいきます。これを最終的にはLSIにしてWCSもROMにできれば価格は数百万円以下。数百万円ではまだ高いですけどマルチユーザがですから1人当たりによれば、うんと安くなります。たとえば4人で同時に使っても現在のシンボリックスと同じくらいの応答速度は出ますから、これくらいのコストダウンには十分意味があると思います。

勝手なことばかり言いましたけれども、名ざしでご批判くださればお答えしたいと思います。

司会 どうもありがとうございました。竹内さんはアーキテクトの立場というより、むしろAIのアプリケーションのお立場でお話いただいたんですが、竹内さんの回りでは研究開発プロジェクトの直接的なインパクトはほとんどなさそうだと見えています。それよりも、われわれにとって大変耳の痛いいくつかの問題点を指摘されました。日本の社会というものはどうも、技術屋さんがシステム作りはやるけれども、最終的なアプリケーションまでなかなかやらない。そのアプリケーションを仕上げるということが大変重要なんだと主張されています。これは最初の鈴木先生のお話に似てるんですが、やはりそこを突かなければだめだというお話でした。

もう一点は、どうも日本の社会というのは他人の土俵の上で仕事をするということは避ける傾向にある。特にアーキテックチャーの分野では、ソフトウェア技術者の意見をどうもあまり聞いてないのではないかということをおっしゃっていました。したがって全体のつながりが非常に弱い、だから仕事が浮き上がって、まとまったものにならないのではないだろうか。

やはりコンピュータというのは、確固たる文化になっていくので、そのためにはそれを支えるコミュニティというのが非常に重要で、コミュニティ作りが将来の日本にとっては課題で、アーキテクトは特にその辺が欠けてるといってご忠告がありました。

それから、実際にはAIの立場で専用マシンの開発も重要であるが、もう1つのアプローチとしては、商用スーパーコンピュータの上でLISPを動かすなんてことも1つの考えではないだろうかと指摘されています。まあ、いろいろの立場があって、それらがどんなふうによろしくもって展開をしていくかということだ

と思います。

最後に山本さん、お願いいたします。

山本 日本電気の山本です。私も、林さんと同じようにタイトルに非常に困りまして、第5世代コンピュータじゃなくて、一般的な推論マシンが商用コンピュータの中でどういうインパクトがあるかと、というタイトルで話を進めたいと思います。

私の立場としては、まず結論からいきますと商用コンピュータの中で推論マシンは、かなり重要なウェイトを占めて、将来は多くの部分で用いられるだろうと、また、用いられるようにと期待を持っております。

私は今まで専用マシンの研究をやってきたわけですが、なかなか専用マシンの道はむずかしいことを認識しております。しかしながら、Prologをはじめとする数式処理マシンの場合には、専用マシンとしての道は、かなりあるんじゃないかと、また、ありがたいという気持ちを持っています。

図-1は、将来のコンピュータのイメージを示しています。推論マシンが並んでいて、うしろに従来型のコンピュータがある。人間から見ると、計算機というのは推論マシンが見え、そのうしろに、必要な処理に応じてスーパーコン、シュミレーションマシン、データベースマシンなどがならんでいる。処理の大部分は推論マシンで行われ、それを必要とする仕事人間から要求がきたときには、その仕事をこれらのマシンにディスパッチしてやってもらうわけです。

つぎにマシン構成ですが、鈴木先生のお話ですと、32ビットの汎用マイコンがあれば、大体かたがつくというような意見であったような気がしますが、私は32ビットのマイコンぐらいじゃ推論処理は困難でもう1桁ぐらい、別の言い方をしますと、今の大型コンピュータの性能に近い推論処理が可能な推論マシン

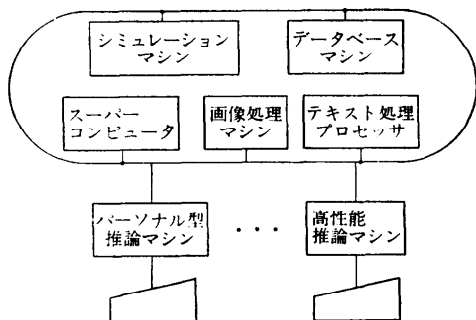


図-1

がほしいと考えてます

また、モトローラの M68000 やインテルのマイコンですと、現在でも 1 万円前後ですから将来はかなり安くなり、私どもメーカーにとって、月産数十万個ぐらい売っても、あまり儲からないことになります。もっと儲からないと困るわけで、これらよりインテリジェンスを持った専用の推論コンピュータとして高価格のものを売りたいと思います。そして、これらはいろんなアプリケーションをやる中で出てくると思います。

初めにもどって、われわれがやらないといけない仕事を分析すると、大きく 2 つに分けられます。第一は手続的なもので、別の視点でみると定型的なものといえます。また、この範ちゅうのものは主に数値が主体の作業になるでしょう。もちろん非数値的なものも一部入るでしょうけれども。

もう 1 つの範ちゅうは非手続的と言うか、必ずしも手続として表現されてるんじゃないで、時々刻々やり方がいろいろ変わるもの。いわゆる非定型的な業務であって、それは、その中身を見ますと数値ではなくて非数値処理といえます。

この 2 つの分野に分けた場合に、実際にわれわれが日常やってる仕事を見ますと、もちろん人によって違うわけですが、第二のほうが圧倒的に多くて、第一のほうは非常に少ないだろうといえます。従来の計算機は第一を主体として、早く計算することに使われていた。しかし第二のほうはそうじゃなくて、どうやるかというやり方を考える手段として使われるものだと思うんですね。この第一と第二の割合の具体例が今年の 6 月号の Spectrum にでていましたので紹介します。

図-2 は専門家の作業を分類したもので、横軸が年代で、縦軸は作業の割合を示しています。内容はざっと見ていただければわかると思いますが、サービスとか分配、あるいは判断とか調査とか学習などが非定型的なものです。

この図からもわかるように、これからずっと将来を見ますと、かなり非定型的なものが多くなってくると予想されます。

さらに、この中の判断の部分について細かく分析しますと、この図のように分類がされておりまして、たとえば直感的作業などはコンピュータでどれくらいできるか、非常にむずかしいところだと思うんですけど、それを

手続き型処理
従来型コンピュータ

非手続き型処理
第5世代コンピュータ

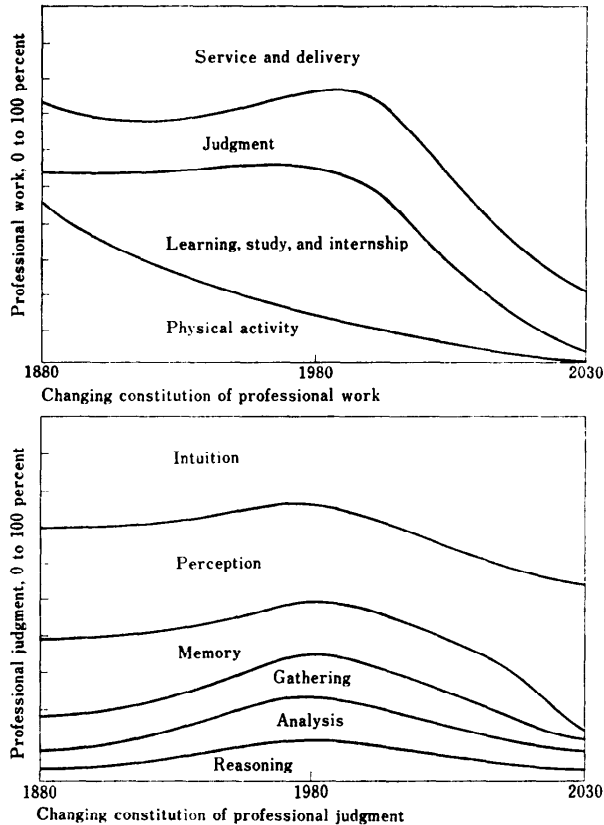


図-2

助けるようなコンピュータの役割はあると思います。

この他、記憶、データ収集、解析など理詰めの仕事をやっていくような仕事がありますが、これから非常に必要になってきます。

これらを機能的に分析してみますと、3つの機能を実現する必要があるだろうと考えられます。1つはいろいろな状況とか個々のデータの中から、ルールとか方式、アルゴリズムなどを定式化していくことです。

2番目は、定式化された方式やルールを検証することおよび、そういうルールやアルゴリズムを早く実行することです。

3番目は設定されたルールを調べたり蓄えたり、また、新しい要求があったときに蓄積されているルール、手法を提示し援用する機能です。

このような機能の分析をもとに設定したり将来のコンピュータのイメージが図-1です。これは一般的には、人間の大部分の仕事というのは非定型的な作業がほとんどである。その中のある部分は定型的な作業だ

ということです。

また、処理の中身から見ますと、大部分は非数値の処理で一部が数値処理です。

それを具体的に実現しようとした場合には、1番最初にお見せした図-1のコンピュータシステムというものが考えられるだろうということです。こういう形が理想的じゃないかなと思います。したがって、人間から見た場合には推論系のエンジンが前面に並んでおり、大部分の仕事はここで行われる。そして、必要に応じて、そのうしろに並んでいる従来型の数値処理計算プロセッサが起動される。したがって、手続型処理を行う従来コンピュータは小さく、非手続型処理を行う推論マシンのほうが徐々に大きくなっていくでしょう。そしてこちらの推論マシンとは、パーソナル型の推論マシンと非常に高性能な推論マシンがそれぞれ必要だと思います。これは今の情報処理分野のコンピュータで、マイクロコンピュータからスーパーコンピュータまでるように、推論マシンとして見た場合に

も、パーソナル型のものから超高速推論マシンまで必要だということです。そして、パーソナル型のものとしては、第5世代プロジェクトで開発中のPSIが対応します。そして高性能推論マシンとしては、私どもが現在第5世代プロジェクトの一環で開発中のものが対応します。これは冒頭申し上げましたように今の32ビットマイコンより1桁ぐらい速い推論の専用マシンです。

次に、その高性能推論マシンの概要を紹介します。このマシンはPSIのバックエンドプロセッサとして動作します。性能としては、100から150K LIPSを目標にしていますが、机上評価では250から300K LIPSぐらいはいくだろうと予想しています。メモリ容量は64MB備える予定です。

ハードウェアの特徴としては、速い素子を使いマシンサイクルを速くすることがポイントになっています。

次に、アーキテクチャの特徴を示します。第一はパイプライン処理、第二はプリフェッチング、これは汎用の超大型計算機で行われているものに近いプリフェッチングを行うことにしています。

第三は、Prologを実行するには、数個のスタックが必要になるため、これらの制御に必要な制御情報を蓄えるレジスタファイルを内部に持たせています。

第四は、メモリ管理プロセッサとパターンマッチング処理ハードウェアを備えることです。

ファームウェアの特徴として、1命令は80ビットで3個の並列処理を行えるようにしています。

このようなアーキテクチャ、ハードウェアの特徴によって、250K LIPSぐらいは達成できる見通しです。以上です。

司会 どうもありがとうございました。山本さんはやはりICOTのプロジェクトにはかなりご関係がございまして、そのお立場ということもあるんですが、むしろ第5世代コンピュータの推論マシンというところに話題をしばって、それがこれからのコンピュータにどのような影響を与えるか、という見解を述べていただきました。

将来のコンピュータのシステム構成というのが、やはり今までのコンピュータではちょっと不得意な非手続的、非数値的、あるいは非定型的な処理という場面には、やはり推論マシンのようなものがある有用ではないだろうか。一般のデータ処理に関しては、スーパーマシンや普通のコンピュータが結合したシステムになって

くるんじゃないかというお話でした。

しかし、実際は現在、たとえばICOTでやられている推論マシン、逐次推論マシンのようなものではなくて、せめてその1桁以上高速のもの、そしてやはりチップではM68000なんかよりもさらに1桁機能の高いものにしてみたいというお話だったように窺われます。

以上それぞれのパネリストの見解を示していただきました。今までのお話に関連して、どうぞ皆さんにかご質問がありましたら遠慮なく、気楽に、全員参加の形でやっていただきたいと思います。

曾我 感銘深いお話で感激していますが、今回の主題のインパクトに関しまして、先程竹内さんからインパクトはないというようなことを言われたんですけども、実は私、非常にあるんだというふうに考えております。むしろすでにインパクトは与えてしまってるんだと言ったらいいのかもしれない。現在この討論会が開かれてること自体、インパクトの上に乗ってるわけですから、しかもこの第5世代のプロジェクトが発表される前に、これだけ人工知能とかPrologとか言う人口がいたかと言うと、そういうことはなかったと思うんです。ただ1つの独立した研究分野として存在してただけで、こんなに世の中が騒ぐことはなかった。ものすごいインパクトじゃないかと思います。ですから、極端なことを言いますと、現在の第5世代コンピュータのプロジェクトがそのまま消えてしまったとしても、このインパクトはそのままずっと残ってるんじゃないかと思います。ですから第5世代コンピュータがもしこれで失敗しますと、逆のインパクトを与えることになりますので、是非これは成功させていただきたいというふうに考えてるわけなんです。

それからコンピュータを作る場合に並列処理して高速化するというふうによく言われるんですけども、こういう考え方は、ちょっと横のほうに置いてもいいんじゃないかというふうに考えてるわけです。と申しますのは、コンピュータの処理というのは大体並列処理も直列処理もあるので、直列処理もできるコンピュータを作るのが自然じゃないかと考えてるわけです。ですからその結果として高速になるんじゃないかと考えております。私自身並列処理の研究をしてる関係上、並列処理にすれば速くなるというふうによく言うんですが、これは速くなるというのは、計算機が速くなるということとして、処理が速くなるということじゃないわけです。計算機が速くなることと、処理

が速くなることは、どういうふうに違うかと申しますと処理というのは、並列にできない処理を一生懸命、計算機で並列処理しようとしたって速くならないわけです。計算機はいくらでも速くできますけど、処理そのものに並列性がなければ、速くならないんです。ですから一体何が速くなるかをはっきりしなければなりませんと思います。

それからもう1つ、私自身は高級言語マシンはあまり性能が上がらないだろうという偏見を持っています。高級言語マシンというのは言語があり、実際に動作してる計算機というのは2進法で走っているわけです。ですからどうしても高級言語から2進法のところまで翻訳する過程が必要です。それをいちいち翻訳してますから高速にならないわけです。もし高速にしたいんだったら低級言語マシンを作るべきだと思います。もし高速言語マシンを作るのであれば2進法じゃなくて、高級言語マシン数で動くものを作らないことには速くならないんじゃないかと考えております。

司会 ありがとうございます。大変おもしろいご意見です。極端に解釈しますと、もう十分インパクトを与えているので第5世代コンピュータプロジェクトはおしまいにしてもよいのではという冗談がよくいわれています。それくらいいろいろなインパクトを与えて、AIの研究も盛んになってきているんじゃないでしょうか。しかし、やはりそれにつけても言い出しっぱですからプロジェクトは是非成功させてほしいですね。

それから並列処理でも直接処理も効率よくできるということが、大前提ではないだろうかという意見です。また、コンピュータは、並列マシンによってそれ自身は速くなるんだけど、処理自身は必ずしも並列と同じ程度に速くならない。それから高級言語マシンというのはセマンティックギャップが大き過ぎてうまくいかない。むしろ、もし作るなら低級言語マシンをベースにしたほうがいいんじゃないかというご意見だったと思います。

さて、パネリストの皆さんにかご意見ございましたらどうぞ、

竹内 現在インパクトを与えたことは十分承知のわけなんですけど、そういうインパクトはなくて、もっと、その10年後の本当の世の中に売ってる計算機に対するインパクトという意味で私は解釈しました。今多くの人がわっと騒いでるんですが、曾我さんがおっしゃったように、それがボジャラないことをひたすら

願っております。

鈴木 やはりインパクトとしては、人工知能ワークステーションという、ああいう画面上で絵が、言葉と混ぜて使えて、非常に高速なコンピュータというのは今急にのびてるわけですね。しかしあれは ICOT が考え出したんじゃないくて、ゼロックスや、シンボリックスが考えたんであって、ICOT はちょうどいいときに他人の尻馬に乗ってわっといったんだと思います。

それからコンピュータの速度ですが、どちらがいいか。要するにわれわれソフト屋から言うと、どんなコンピュータを持ってきても、時間さえくれれば非常にいいコンパイラは書けます。68000 がいいというのは、あれは非常に安定したものが長くあるんで、しょっちゅう書き替えている必要はない。マイクロコードの場合、問題なのは非常に寿命が短いわけですね。Dolphin とか Dorado も、われわれがマイクロコードを書き替えれば、5倍にはなると思います。けども、それをやる気はないのは、やったとしても寿命がない。ですから高級言語マシンだろうがビットマシンだろうが、安定したハードウェアを持ってきてくれて、それが非常にインパクトがあり、これはずっと長く生きると言うのであれば多くの人がいいソフトを書くでしょうね。

山本 1番最初のインパクトの話ですが、私はインパクトがあると思ってます。そのインパクトの範囲を AI とか、知識処理とかいう言葉で表現されているとすると、使いやすいコンピュータ、誰でも使えるコンピュータ、AI の技術を組み込んだ高機能インタフェースなどにすでにインパクトが出てきてるんじゃないかと、私は考えております。先程から紹介のありましたワークステーションは、1つの具体例だと思っております。そんなに急に変わるものではないし、急に変わったら使う人は困ってしまうんじゃないかというような気もしてますから、技術は徐々に変わっていくべきものですし、この変化はすでに始まっているんじゃないかと私は思います。

それから高級言語マシンについてなんですが、これまで RISC と CISC の議論は国内、国外で行われていますが、今こちらにおられるアーキテクトの立場からみますと RISC は全然おもしろくないんじゃないかという気がしています。先程曾我さんがご指摘になりました、たかだか1桁。しかし、1桁も上があれば結構いいんじゃないかと思うんですね。十分効を奏してるだろうという意味で、私はもっと、特に知識処理の場

合ですと言語としては、かなり高いレベルのものが処理対象となりますから、大いにやるべきなんじゃないかなという観点で見えています。

林 並列処理により、計算機は速くできるが、処理から見れば全然違うという意見がありました。しかし、計算機というのは、あくまでも使わなければどうしようもないので、アイドルでいくら動いていても、それだけではなんのために存在するか、よくわからないわけです。そうしますと、そういう議論というのは、もともとあり得なくて、処理が速くならないものは、いくら計算機を速くしてもしょうがないと思います。

それから高級言語マシンということで山本さんも言われたんですが、私少し見解を別にします。と言うのは、高級言語マシンで、たとえば速くしようとするときどこを工夫するかと言いますと、究極にはパイプラインでの実行になります。パイプラインを実行しようとするとき、1つのマシン命令が重いと、パイプラインなんてほとんど不可能なわけです。そうすると性能が出ない。じゃ性能をかせぐためにどうするかと言うと、先程曾我さんが言われたような低級言語マシンと言うんですか、これが基本的には重要になります。今の RISC マシンというのは、そういう概念だと私は思っています。あれはなぜ速くなるかと言うと、パイプラインがきれいにでき、1命令という単位で処理ができるからです。確かにコンパイラとしては非常に負担がかかると思うんですが、速度的にはかなりかせげるんじゃないかと思えます。パイプラインもやろうと思えばさらに段数が上がると思えますから、そういう意味で、いわゆる言語を処理する方向としては、そういう方面もあるんじゃないかと、ぼくは思っています。

司会 いかがですかほかに。

高島 今、商用コンピュータに与えるインパクトがあるとかないかという話ですが、商用コンピュータに与えるインパクトですから、竹内さんが言ったなものもない、というのはある意味では正しいわけです。第5世代コンピュータが、コンピュータのフィールドになんらかのインパクトを与えたかという意味においては、たとえば1900何年からの商用コンピュータにインパクトを与えるか、とかあるいは与えないか、とかいえばまだわかるのかもしれない。

少なくとも大きいインパクトを与えたということとはね。そのインパクトも非常に間接的なインパクトで、今そのことによって人工知能に関心が集まって

Prolog が走るとか Smalltalk が走る。ワークステーションはどれがいいとか、あれを使ってみたくてということで、きわめて間接的には現在の商用コンピュータにもインパクトを与えると言えば与えたんじゃないでしょうか。

それから非常に大きいプロジェクトで開発したコンピュータは売れるはずないと思うんですね。研究用ですからね。いろんな試みやってるわけですよ。そして商用として売るためには、今度は商品としてのコスト・パフォーマンス考えなければいけないわけで、そうすると研究用としてなにもかも入ったものが、そのまま売れるはずがないです。過去のいろんな研究プロジェクトを見てもその通りですね。だけどそういうことは別に、ただそのプロジェクトで開発したコンピュータが、たとえば研究とか製造とかユーザとか、あらゆる階層になんらかのインパクトを与えればそのプロジェクトは成功だと思っています。その商用コンピュータに直接的なインパクトを与えようが与えまいが、なんらかのインパクトを与えれば、ぼくはプロジェクトは成功であると思う。だから第5世代コンピュータもインパクトが与えようが、与えまいが、かかわりなく一懸命やっていただきたいと思うわけです。

司会 ありがとうございます。やはりそれぞれの定義がきちっとしてませんから、異なったお立場で皆さんがご意見を述べてるので、若干食い違いがあるかと思えます。このパネル討論は早急に結論を出すことが目的じゃありませんので、その辺の食い違いがあるということ、わきまえていただければ、まずはいいんじゃないかと思えます。いかがでしょうか。そのほかに。

岡田 いろいろ参考になるお話を聞かせていただいて、ありがとうございます。第5世代というのは、非常に大きなプロジェクトで、いろんな側面から見られると思うんです。そういう意味で初めて少し定義がほしかったと思ってるんですけど。私の解釈では、たとえば ICOT の測所長も言われてるように、1つの解釈としては Prolog をアセンブラ言語として使う計算機である。問題はその上に構築されるいろんな知識情報処理のソフトウェアがありますけど、そういったものを含んだものが第5世代計算機ではないかと私は解釈してるんですが、そう言った場合だと、先程の Prolog が普及しないとか言った話がありますけれども、この Prolog というのはアセンブラ言語ですから、スペシャリストが使えばいいわけですね。これは

別にそんなに広がらなくてもいいんじゃないかという気がします。

それから鈴木先生が、プロセッサが決まってないというソフトウェアはできないんじゃないかというお話ですが、たとえば言語レベルで統一することによって、かなりの程度まではいけるんじゃないかと思えます。たとえば論理プログラミングでも、標準的なものができて、それが広がれば少しは安心して組めるような状況になるんじゃないかというような気がいたします。それで、われわれアーキテクチャ屋が集まっているわけですけど、アーキテクチャに関しては、たとえばある程度確とした見通しはあるんじゃないか、というような気がするんですけれども、先程の、その上に構築される知識情報処理システムがたとえば10年後に、どれくらいものが可能であるかということ、竹内さんと鈴木先生に教えていただければ幸いです。

司会 Prolog というのは、ある意味ではアセンブラ言語で、スペシャリストが使えばいいんじゃないかという話ですが、この辺について鈴木先生ご意見は。

鈴木 別に意見はないです。

司会 それでは、10年後には知識情報処理システムというのは、どの程度のところまできてるのでしょうか。

鈴木 むずかしい質問ですね。アプリケーションは非常にむずかしいんだと思うんですね。アプリケーションと言うと、もろに個人の発想がきいてくるわけですね。ない分野のものを作り出そうと。コンピュータを速くしようというのは、すでにコンピュータというものがあって、与えられてそれを速くするというので、思考形態としては Convergent ですね。256 K ビットの次は1メガビット、1メガビットの次は4メガビット、その次は16メガビット、同じようにやっばり100 K LIPS の次は500 K LIPS で、その次は1メガLIPS と、これは知能指向として深くつきつめてずっと考えるような人には適任の仕事ですね。

アプリケーションを考えると、これはまったく別でして、ダイバジェントに考えないといけない。要するになにか新しい使い方を考えるという。アーキテクチャ屋とか、コンパイラ屋だとかいう人には不得手な分野でして、芸術家とか心理学者が考えたほうがいいんじゃないかと思うんですね。

それはさておいて、どういうアプリケーションが出るかということ、日本人には不得手な分野かもしれ

ません。日本人というのはコンパジェントに思考する人間ですね。ですから S/370 があると、S/370 より速いコンピュータを作るのは日本人はすごく得意です。Prolog だって世界で1番速く、メモリだって世界で1番密度の高いのを作る。ダイバジェントな思考は日本人は弱いで、日本の今の教育とか風土とかにも問題があるのでしょう。

竹内 Prolog がプロフェッショナル向きのアセンブラというのは宣伝されている言い回しとは大分違う気がしますが、アセンブラであることは事実です。しかし Prolog で書いたものを、他の言語に直すのが困難であるという意味では非常に高級なわけです。今までの言語ですと、なんで書いても、こいつはフォートランに移植しろとか、Cに移植しろとか、なんとかできたわけです。しかし、Prolog で本当に大きいアプリケーションが書かれたときに、それを他の言語に移す気がするかと言うと、まずしない、そうなったときは、Prolog の勝利ですね。逆に本格的アプリケーションが作られなければポシヤ。Prolog は問題とする対象がパターンであるときに、局所的に非常にいいんですけど、全体を Prolog だけで書くというのは、全部アセンブラで書くのと同じでやはりどうも疲れるところがあるような気がします。また、プロフェッショナルというのは、最終的にコストパフォーマンスのことを必ず考えますから、遅いとだめですね。それから安くないとだめです。開発コストまで含めて全部トータルで考えて Prolog で書くのが速くて安いということになるのかもしれないですけど、いったん製品として出てしまった場合、同じようなことをするのに、Cで書いたほうが Prolog よりも5倍速ければ、誰だってそっちのほうを買うわけですね。ですから商売にするソフトウェアを作ろうという人は速度のことを考えるはずで、速度のことを考えるとアルゴリズムにももの考えるということですから、どうやってメモリを節約し、どうやって N^2 を、 $N \log N$ にするか。そういうことを一生懸命考えてプログラム作るわけですね。Prolog ではそれが、やりにくいような気がしています。それをさぼれるというのが、Prolog の長所であるという前宣伝と裏腹ですが……。だから本当のプロフェッショナルが Prolog を使うかは疑問です。気遣いみたいなプロフェッショナルがいて、Prolog で大規模なソフトウェアを作ってしまうと、それ以後 Prolog は本当に確固たる地位を築くんだという気がします。いちかばちかということ

です。

鈴木 アプリケーションの話なんですけども、トップダウンでいって一体なにが必要かを考えます。私の生活を考えますと、たとえば講義をしたり、なにか日程を組んだり、論文を書いたり、プログラムを書いたり、本を書いたり、原稿を書いたり、いっぺんにパラレルにいろんな仕事をしています。ですからほしいものは、自分の仕事に必要な書類や資料はどこにあるかということとは考えなくてよくて、どこまで書き残してあって、ほかの仕事はどこまでやってあるかということとを全部記憶してくれる非常に頭のいい秘書みたいなことをやってくれて、それがどこからでもアクセスできるものがほしいですね。ぼくのオフィスの中に1つのワークステーションがあって、そこに行かないとなんでもできないと言っちゃこまります。だからネットワークが非常に必要になりますね。それからグラフィックスが非常に必要になるとか、まあ、推論とかデータベースというのは、ほとんど外に出てきてくれなくていいと。だから私自身 Prolog でプログラムを書く必要はないでしょう。だからなぜ ICOT が Prolog に固執するかわからないんですね。一般ユーザが、Prolog で書くことはまずあり得ない。Prolog はなにに優れてるかと言うとリレーション・データベース・クエリぐらいですか。論理型というのはやっぱり、論理型を第5世代がやると言ったんで、ひっ込みがつかないんで、いつまでたっても Prolog をやると言ってるだけぐらいしか理由がない。だから Prolog というのは第5世代のキャッチフレーズだけで、もうほとんど意味はないと思いますね。

司会 いかがでしょうか。なにかご意見。

白川 第5世代のことにしましては、どっちかと言うと私は週刊誌で読んでるくらいの知識しかありません。そういう知識で申し上げるんで申し訳ないんですけども、鈴木先生のほうからおっしゃっていただきましたが、第5世代コンピュータは、やはり道具を作ってるということであるということです。私どものような者から考えますと、その道具を使ってで上がる最終製品のほうに興味があるわけです。たとえば、自動翻訳などがキャッチフレーズにあったかと思いますが、最終的にたとえば、鞆とか、できれば手帳ぐらいの大きさの機械があって、全然わからない言葉をしゃべられると、リアルタイムに日本語になって聞こえてきて、こちらがなにか日本語でしゃべると、それが相手のほうにちゃんと確実に伝わる、そういうようなも

のでできることが目標にあると思うんです。そういうところから Convergent に考えていくということが必要なんだと思います。

それからまたなぜ、そういう論理的な思考をするものが期待を持たれるかと言うと、世の中にはやはり解決しなければいけないけれども、なかなか解決のつかない問題がずい分あって、それは単に価格計算が速くなればすむと言うような問題じゃないような、たとえば先程おっしゃったスーパーコンを開発すると、貿易摩擦が増大するかもしれない。貿易摩擦を解消するにはどうしたらいいかと言うと、これはやはりなにかに論理型の計算機があるんじゃないかということになって、そちらのほうで解決していただきたいというような気がします。あるいは、いろいろ世界の緊張を緩和する方法などいろんな問題が世の中にあると思います。そういう問題を解決するために必要なコンピュータということで、ある程度 Convergent に考えていただいたほうが、でき上がった成果が、たとえば自動翻訳機械で、もし先程申しましたようなのができ、1台2万円ぐらいですと、8台ぐらい買まして7台を世界中に小包で送り、それが衛星通信で自動的につながるようになってますと、ペンフレンドの代りに、自動翻訳機フレンドが世界中にできることになります。そういうことになると非常に楽しいですね。そういう機械はきっと何千万台も売れるだろうし、国連でも採用するだろうし、大変商用機としてインパクトの大きいものになると思うんですが、そういう発想もやっていただきたいと思うんですけれども。

翁長 いろいろお聞きして、非常に示唆に富んだお話だったんですが、要するにコンピュータの、ソフトウェアの文化が問題であって、アメリカからソフトウェアに関して UNIX とか、C 言語とか、Smalltalk というのは、なにか波源がアメリカのほうにあって、そして世界に広がりつつあるように思います。中途はんばに完成したぐらいで出てきて、ライセンスを売って広がっていく過程を見ていると、なにか文化の伝播が原始的で、今でも10年ぐらいのあいだに広がっていくような感じがします。竹内さんが文化という言葉が使われたの、なにか文化伝播のメカニズムを見るような思いがするんですが、また、鈴木さんが商用機には、資本主義の原理が非常にシビアに働いてるといわれましたが、美学とか、直感とか、あるいは論理学によって組み立てていこうとするようなものではソフトウェアは作れないんじゃないのかというような気もし

ます。つまりアメリカから伝播していく原動力はなんなのか、アメリカ人の天才的な才能がそれなのか、彼らがクリエイトするものがロジックで出てくるんじゃないかと、天才のひらめきによって出てくるのか。なんでアメリカからそういうものが生まれ出てくるのか、それをちょっとお聞きしたいんですけども。日本で第5世代のコンピュータが大きなプロジェクトとして注目されていますが、その中で働いてる人の個人の顔は見えません。第5世代という名前はあるけれども、なんとかさんが働いていてというような感じはちょっとしない。ところがアメリカからやってくるのは、皆ある個人が出てきて彼がやったんだという感じで語られてるに対して、第5世代プロジェクトは顔がなくて、のっぺらとしてるような感じがしている。だから逆に商売で案外勝つかも知れない。確か私が聞いたところによると、Prolog が採用されたのは、政治的な理由によると。つまり Lisp じゃ困ると、通産省の役人が何かおっしゃったというようなことも聞いております。これはコンピュータサイエンスに対するインパクトであって、商売をめざしたものじゃないんじゃないか。商売は商売で日本はうまくやるんじゃないかと思います。つまり文化は作らなかつたけれども商売には勝つのではないだろうか。ただ第5世代プロジェクトは、日本がなにかコンピュータサイエンスに対して貢献するんだというようなところでどうもあつたんじゃないかとも思います。しかし Prolog によってショック・ウェーブを発生させたんだけど、残念ながら個人の顔が見えないところが、ちょっと気になるところなんですけれども、どうしてアメリカからソフトウェアのいいものが、ここ数年のあいだに出てきたかということでご意見をお聞かせ願いたいと思うんですけども。

司会 今のご質問なんですけれどもビジネスコンピュータへのインパクトはさて置いて、なぜアメリカのほうから新しい文化が出てくるのか。これは先程鈴木先生と竹内さんからご指摘がありました。日本はアプリケーションまで徹底的にやらない。この辺に原因の一つがあるような気がします。それから、是非これから一生懸命やろうよということを提唱されたと思うんですが、それをうまくやるためには、どうしたらいいかも含めて、なぜ米国から素晴らしいクリエイティブなものが出てくるんだらうか、そうするためには日本はどうしたらいいんだらうか。まずアメリカのことをよくご存じの鈴木先生からご意見をお聞かせ下

さい。

鈴木 去年のチューリング賞を取った UNIX を作ったトンプソンとリッチの受賞講演で、これはビットにも翻訳されていますが、そこでベル研とゼロックスの PARC が最近の15年間の大きなソフトウェアの仕事をした唯一の所で、IBM からは出てこなかったといっています。それはやっぱり、ああいういいソフトウェアを作る環境としては商売のことを考えてはだめだ、と言ってるのが非常に印象的ですね。ゼロックスって会社は商売は非常に下手な会社だけど、ソフトウェア研究は非常に優れてました。Deutsch が、さっき非常に速いを作ったと言いましたけれども、あれはまだ商品になってないし、Tektronix よりずっと速いんですが、彼自身は商品にしようという意志はまったくない。それからベル研もそうですね。UNIX を作って、世界の OS マーケットを席巻したわけです。ベルでは、トランジスタと UNIX がベルで作られた2大発明だそうです。レーザを抜いたわけですね。それほど高く評価されているのに UNIX ではほとんどうけていないし、ATT がコンピュータマーケットに入ったけれども、うまくいかないわけです。だから、コンピュータ・ソフトウェアでいいものを作ろうと思ったら商売のことはあまり考えない方がいい、ということかもしれないですね。だから ICOT をあまり商売商売と責めないほうがいいでしょう。ICOT の研究活動の問題点としては、当面 Prolog でいかなきゃならなかったこと。自分のところの発明でなくて、ヨーロッパで発明されたのを持ってきたが、途中から、Prolog では3年以内ではシステムが書けないということでオブジェクト指向に乗り替えたということですね。それと、UNIX とか Smalltalk の開発を見てると、ごく少数の天才がこつこつ始めて、それを周りで急に盛りたてて出てきた。ICOT みたいに、最初から大プランができてやると少数の天才の働く余地はなくなり、画期的なソフトウェアは生まれないんじゃないですかね。これがゼロックスの PARC と ICOT の違いです。

司会 ありがとうございます。ビジネスから離れば離れるほどいい文化ができるといわれた。竹内先生どうでしょうか。

竹内 大体言い尽くされているような気がします。最後に鈴木先生がおっしゃいましたが、周りが盛り立てるといのがないですね。第5世代の人にしても周りからずいぶん盛り下げられていますね。やっぱりこ

れは日本人の実にみごとに体質なのでしょう。私もよくわかりませんが、これを改善しない限りどうしようもないんじゃないですかね。

林 アメリカで、そういうソフトウェア文化が非常に発展していると言うか、生まれてくるという1番のポイントは、やはり計算機をよく使うからではないかと思えます。日本人は先程言われましたがシステムを作ることばかりをやって使わないんじゃないか。ですからアプリケーションとか、そういう方面からの自分の本当の感じるどころというのは、わからないわけです。いくらそういうマシンでも作るばかりではしょうがなくて、やっぱり使うのがベースにあって、それからではないかと思っております。

山本 私も、先程の竹内さんの意見に近いのかもしれない。日本人は特にアメリカから入ってきたものはいいように思う傾向があります。日本で生まれたものでも、そんなに差があるんじゃないかと、多勢の人が積極的に取り組んで使っていけば良くなり、アメリカのものと同様に成長すると思えます。どちらかと言うと、よそのものはよく見えるけれども、日本でいいものが出てなかなか使わないところがあるんじゃないかなという気がします。そこをもう少し見直す必要があるんじゃないかと思えます。

司会 なにかございますか。どうぞ。

坂村 日本人とアメリカ人がどう違うかと言うと、やっぱり日本人は和食文化で長続きしないで、すぐやめちゃう性格を持っていると思うんですよね。それに比べてアメリカ人は割としつこいと思うんですよ。日本なんかだと大学でもすぐ論文が何編だとか、ささいなことを言いますが、むこうの優秀な人たちは、ある意味で論文などまったく書かない人がいますよね。ただやった仕事は、非常にしつこく何年でもやるからいいものができます。日本は少しほり出すのが早すぎるんじゃないかと思うんですね。人工知能だって、1度日本で昔わっと騒がれて、すぐ評価しちゃう、だめだと言ってやめちゃうわけですね。欧米では、ずっとしつこくやっている。最近だとエキスパートシステムなどがビジネスになるようになってきたというのは、そのしつこきの成果で、そのしつこきを出すために、日本の大学なり研究所なんかは、研究に対しての考え方を少し変えないといけないと思うんですね。ですから Smalltalk などで驚くことは新しい情報を知るのに、なにを見たらいいのかというと、ACM の論文でもないし、IEEE の論文でもなくて、雑誌 "Byte"

をまず読むことなんです。Byte なんて日本で言ったらなにに相当するのかな、トランジスタ技術とか、ビットとかでしょ。そんな雑誌にああいうおもしろいものが出ちゃうこと自体がまったく驚きであり、すごいことだと思います。

それからこのシンポジウムを企画した者としてちょっとインパクトということに対して言いたいんですけど、私の感想ですが商業的にインパクトを与えるものというのは、やっぱり素人がわからないものはだめですね。ですから大体ここ数年でインパクトを与えるものはスーパーコンピュータ、これはまあ、インパクトはあると思うんですね。現に今いろんな場所でスーパーコンピュータ作ってる理由がなんであるかと言うことは誰でもわかるわけですね。もう1つインパクトを与えているのは、やはりワークステーションだと思います。あれもやっぱり素人が見たときに、見て違うということはわかると思います。だけど第5世代コンピュータはわからなくても、インパクトを与えています。なんでインパクトを与えるかと言うと、やっぱりマスコミュニケーションの力ではないかと思えます。芸能ゴシップに限らず、この手のたぐいのものであのチャンネルを使えば、どんなものでもみんな名前ぐらいは知るようになるんですね。それはすごいことだと思います。だからそのくらい日本というのは、世界と通信ネットワークで結ばれてて、10年前、20年前と比べものにならないくらい、情報の伝わる速度が速くなってるということを実感します。

次にちょっと専用と汎用という話が出てたんですが一言、言いたいんですが、専用コンピュータは重要だと思います。専用コンピュータを作るので1番優れている人は、コンピュータ・アーキテクトではなくて応用を持てる人たちなんですね。そういう人たちが、自由にそういうものを作れるようになれば必ず汎用コンピュータより能力あるものになると思うんです。ところが日本でそういうコンピュータを作るのに、応用分野の人たちが乗り出してくることはまずなくて、アーキテクトがやるから十分よいものにならない。もっと両者が組んでやらないと汎用コンピュータのほうがよくなっちゃう。最近ワークステーションの開発が盛んですが、それで使うディスプレイ・コントローラ専用のグラフィック・プロセッサを使う気になれません。それを使うとソフトウェアの形態が非常に崩れてしまうからです。専用プロセッサを使うと、見かけ上安くなりますが、2年たったときにスピードが

上がるかというとその保証はありません。ただたとえM 68000を使っておけば、2年たつとスピードが2倍になるんですね。8 MHz が出て、次16 MHz、32 MHz が出ると、ソフトウェアのほうはなんにもしなくてプロセッサを取り替えれば速くなる。専用マシンを作るのはものすごく大変ですから汎用でいいものを作らなくちゃいけないと思います。その点は少し考え直さないといけません。最後に、日本とアメリカの違いでもう一言、この中にかなり研究的に力を持つての方がいたら、研究はみみちくやっけてほしくないですね。アメリカ行って感じるの、やっぱり先端の研究をして、影響を与えるようなことやるには最高の設備を使ってやろうと言われてます。設備的な差というのは、鈴木先生なんかもよくご存知かと思いますが、アメリカ行ったら驚きますね。

司会 ありがとうございます。いかかでしょうか。鈴木先生コメントありますか、特に日本の大学は、という話もあったんですが、もう少し日本の大学はこうあったほうがいいんじゃないかという、ご意見がありますか。

鈴木 日本の大学で1番頭が痛いのは働き手がないことですね。私も苦勞して学生育てるんですけどもちょうど教えて一人前になった頃に、みんな就職してしまう。だから、これは企業も金の卵になる前に早くつぶしてしまってるんじゃないかと思ひます。もう3年いれば、もっとすばらしい研究者になったと思うのが皆やめていく。

和田 アプリケーションと言ひますか、コンピュータを応用したいという立場から、なにかいい話はないかと思ひてここにやってきました。私はLSIの製造過程のところの仕事をやっていますけれども、先月アメリカに参りまして、テスト・コーフェレンスという所で話を聞きました。テスト技術に関して、人工知能の応用というのはどうかという話が、今年初めてテスト・コーフェレンスで人工知能のパネルで取り上げられました。そこでパネルの人たちの言っている話は、結局言語が使ひにくく、従来の自分たちが使ってきた言語を拡張してやっったほうがよほどやりやすいということ強く言ったと思ひます。私は研究所の中でLSIの製造工程の仕事をやっていますが、LSIの製造工程というのはノウハウのかたまりであるというふうに見えます。工程が200以上になるような、非常に複雑な過程は、結局は数値化できないノウハウという形が残ってしまひます。ユーザは、なんでノウハウという形

で、技術を自分たち1人の知識にしておくのかという、ノウハウというのは、数値に表して見ると、他の人たちから、そんなものは理論になっていないのじゃないかと、ばかにされるからです。ですから、やはりノウハウは公共の場所に出にくい性格のもので、ユーザとしては、自分たちでこっそり使えるような場所でコンピュータ上にインプリメントできますと、かなり役立つだろうと思ひわけです。そういう点で、先程からお話がございましたようなアプリケーションの特にユーザ・フレンドリなシステムを開発していただくということを是非お願いしたいと思ひております。

司会 ありがとうございます。コメントとして承っております。

曾我 私大学で働いてるものですが、どうして日本からいい研究が出ないかということをもいつも考えています。実際日本では、もうちょっと研究者は失敗すべきじゃないかと思ひてるんです。一般に失敗というと、小さい失敗はかまわないんですけど、大きな失敗が許されないような雰囲気を持っています。大きな研究をするんだったら99ぐらい失敗して1つぐらい成功するぐらいの研究をやってもいいと思ひます。

特にそれができるのが大学でして、大学はかなり自由がききますので、大きな研究をして、ほとんど失敗だということがあってもいいんじゃないかと思ひてるわけなんです。それからもう1つ最初の話にもどっちゃうんですけど、計算機を速くすることと、処理を速くすることは違うというふうに言ったんですが、これはアーキテクチャをやっておりますと、特に汎用コンピュータのアーキテクチャを作った場合ですけれども、計算機が速くなるということはわかるんですけども、それが処理まで速くなるかどうかということは、ソフトウェアをやっている方とアプリケーションをやっている方に聞いてみないとわからないわけですね。ですからどうしても先程申されたように、いろんな方と組んでやっていくということが必要じゃないかと思ひております。

司会 ありがとうございます。失敗を恐れずに挑戦しましょうとの提案です。ただ失敗した場合になぜ失敗したかということを確認して、将来の研究に活かせばそれで十分ではないかというわけですね。それに付けても、いろんな方と一緒に研究をして評価をするということが重要なんではないだろうかというご意見だったと思ひます。

と言うわけでそろそろ時間がきてしまひました。大

体核心に迫ると、こういうパネル討論はおしまいになるというのがきまりです。きょうは2時間20分もとっていただいたんですが、やはりおもしろくなったところでおしまい。ですからこの続きは次の機会のパネル討論でやっていただきたいと思います。なおきょうは、ちょっと残念に思いますのは、ICOTに深い関係を持つ方があまりおられなかったということです。

第5世代コンピュータの分野に関してはこれから更

にディスカッションの花が咲くだろうと思いますから、それに期待したいと思います。

きょうはパネリストの皆様方に、お忙しいところを貴重なご意見を頂戴ありがとうございました。またご討論に参加していただきましたフロアの皆様方にも感謝を申し上げます。拍手をもって終わりたいと思います。ありがとうございました。(拍手)