



# シリコン素子を継ぐもの

6. ソフトウェア側からみたハードウェアへの要望

## 生涯パートナーとしての計算機

元田 浩 川口 敦生

日立製作所基礎研究所

理想の個人用計算機は、どうあるべきだろうか。ユーザと行動をと  
もにし、意思の疎通を図りつつ、さまざまな活動を支援する計算機  
の可能性を探り、記憶(知識)に処理能力の基礎をおく空間指向パラ  
ダイム、計算機とユーザの相互理解に的を絞った説明指向パラダイ  
ム、生涯パートナーとして動作し続けるための超信頼性などの実現  
のための主要技術課題を論ずる。

### 1. はじめに

近年の計算機技術の進展には目を見張るものがある。現在までの進歩を概観すると、大半の努力はプロセッサの高速化と処理形態の並列化を柱とした実質的な処理速度の高速化に向けられてきたように思われる。この傾向がこれからも続くことを想定すると、プロセッサもいずれはピコ秒オーダーの動作をするようになり、アーキテクチャの方も、超並列化への道を邁進し、プロセッサの処理速度向上と相乗し、実質的な演算速度は飛躍的に向上することになるであろう。一方、微細加工技術の進展に伴い、ICの集積度は年々向上し、このまま外挿すれば21世紀はじめには現在の数1000倍のメモリをもった計算機が実現可能になると予測される。

近い将来、非常に高速で大容量な計算機が安価に提供され身近なものになったとき、我々は何を計算機に期待するのであろうか。計算機はただ、高性能であれば、皆喜んで使うのであろうか。多分、今の使い方の延長で考えれば答えはノーであろう。今でも計算機は使い方によっては非常に便利な道具であるが、何とんでも頑固である。頭の回転がいかに速くても、頑固で融通の効かない人とはつき合にくい。計算機に対しても同じことが

いえる。

ありあまる性能を十分発揮しつくすためには、どのようなことがこれからの計算機に求められているのであろうか？ 計算機は単に使う便利な道具から、我々の生活に必要不可欠なものへと変わるであろう。そしてその究極の姿は、揺籠から墓場まで我々に付き添い、ともに成長する「生涯パートナーとしての計算機」となるであろう。本稿ではこのような問題意識から筆者らが計算機の将来について日頃考えていることを、予測し得る必要な技術課題を含め、述べてみたい。

### 2. 計算機から記憶機へ

古くは水銀遅延線から磁気コアなどを経て半導体メモリにいたるまで、電子計算機の歴史の中でメモリは一貫して貴重な資源であった。このため計算機に関する多くの研究がメモリ使用量の抑制を目標あるいは成果の1つに挙げてきた<sup>\*</sup>。その根底には、計算して導けるデータについては、可能な限りそれが必要になった時に計算して求めればよく、高価なメモリに憶えておくのは資源の無駄遣いであるという考えがあった。本来必要なメモ

<sup>\*</sup>特にオペレーティングシステムの研究にこの傾向が強く、UNIX<sup>®</sup>も少量のメモリで動作可能であることを強調している。

りを計算で補うため、当然計算手数が増えプログラムの実行には余分な時間がかかる。こういったことも、高速化の研究に拍車をかけた要因のひとつであろう。また、仮想メモリの考え方は、このような思想の現れともいえる。

計算の重要性は、これからも減ることはことはないであろう。特に、科学計算など計算手順が明確なものは専用プロセッサが使われ、プログラムを書くことと専用プロセッサを作ることが同じになるであろう。そして個別の問題には個別の専用プロセッサが使われるようになるであろう。現在では、計算で予測不可能なものは実規模あるいはその何分の1かの模型を作り、実験を行って性能を予測している。このようなものはすべて計算機シミュレーションにとって代わられるであろう。さらに、実験が極めて難しいものや不可能なもの大半はシミュレーションによって再現され、計算機シミュレーションが新たな知見を得るのに大きく寄与するであろう。しかし、これらは計算機の使われ方のほんの一部に留まり、ほとんどの計算機は我々の生活の一部にとけ込んだ、一般ユーザが日々の業務(すなわち、計算以外の)に使う道具として使われることになるであろう。

このような兆しはすでに、最近のワードプロセッサ、パーソナル・コンピュータ、ワークステーションなどの使われ方に現れている。そこでは、各種のデータベースに代表されるようにいろいろなデータの記憶、加工、それらの再利用、伝達が主要な用途になっている。そして、記憶容量や通信機能が重視され、計算速度そのものは(もちろん重要だが)裏方ともいえる。人工知能の分野では、研究が進むにつれて知識の重要性が認識されてきた。初期の頃は少数の公理と強力な推論能力があれば、世の中の多くの問題は解決できると考えられていた。しかし、その後の研究でこれは間違いであることが分かってきた。頭が良いだけでは不十分で、知っていなければ無力に等しいということは、日常生活においてよく経験するこ

とである。したがって、知的な計算機には大規模な知識ベースが欠かせないということは共通の認識となっている。すなわち、「記憶」しておくことの重要性が明らかになってきた。この考えは先に述べた現在までの計算機には欠けていた視点である。すなわち、これからは「計算機」から「記憶機へ」の道を歩むことになるかと予測される。この原稿を執筆中に、社団法人情報処理学会の「30年後の情報処理」という特集の中で、同じような考えを表明している記事<sup>2)</sup>を偶然見つけ、筆者らも意を強くしている。

常識を計算機に埋め込もうという壮大なプロジェクト(Cyc Project)がアメリカで進んでいる<sup>3)</sup>(本号 p.115参照)。百科辞典を計算機に入れるのであるが、百科辞典を理解するにはさらに百科辞典あるいはそれ以上の知識が必要になる。Cycでは、人間が手作業で入力している。しかし、知識がたまるにつれて、その知識を使用することができるようになるため、入力作業は楽になっていくと報告されている。最終的な知識の量は1億個(片)と見積もられている。1片を100バイトとすると10ギガ(100億)バイトという勘定になる。また、現在、金融業で使われている計算機システムの平均ディスク容量は約30ギガバイトである<sup>4)</sup>。これらは現在の記憶装置から考えれば莫大な量である。しかし、いずれ、たかがギガバイトといえる時代が間違いなく、それも、かなり早く(おそらく5~10年後に)来る。

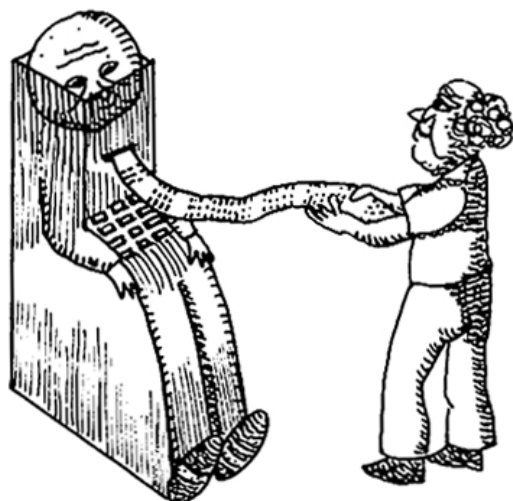
IC技術の進歩、特にダイナミックメモリの集積度は年々向上し続けており、西暦2000年前後には1ギガビット・ダイナミックメモリが実用になると予想されている<sup>5)</sup>。すなわち現在の基準からいえば、事実上メモリ資源を無制限と考えてよい時代が訪れる。そうなれば、Cycの知識をメモリに乗せることが可能になる。しかし、従来の計算機はこのような状況を想定して作られていないため、このままでは、せっかくの大容量メモリも十分活用できるとはいえない。さらに「記憶」しておくという考え方に基づいてプログラムを書くための効果的な手段は一切用意されていない。メモリ容量の制限が事実上なくなればプログラム・ス

\* 家庭の主婦や事務に携わる人々のみを指すのではなく、技術者や研究者も含む。

タイトルも大きく変わるであろう。そこでは何を記憶し、何を計算すべきか、どのデータは何に依存して決められているかなどのきめ細かい管理が重要になり、これらはすべて計算機側で管理されることになる\*。限られた資源を有効に使うことを主眼にしたアルゴリズムはそれほど重要ではなくなり、むしろ時間領域を空間領域でカバーする「空間指向のパラダイム」が出現するであろう。つまり、量から質への変換が実現するであろう。

人間は正確に計算し、探索し、記憶することは苦手である。しかし、極めて優れた連想能力をもっている。そのため、ほとんど考えないで、あるいは、ほんの少し考えただけで、与えられた問題が自分の知識で処理できるかどうか判断できるし、膨大な記憶の糸を手繰りながら所望の知識を効果的に引き出すことができる。計算機の「正確な記憶」能力は強力な武器になる。しかし、我々には計算機がただ正確に記憶しているだけでは膨大な「記録」をうまく処理することは難しい。計算機が我々の連想を補助してくれる機能が不可欠である。そのためには、計算機自身が自分を使っている人(ユーザ、持ち主)のおかれている状況(ここでは「文脈」ということにする)を把握できなければならない。すなわち、文脈の共有が不可欠になってくる。

文脈に沿って的確に連想するには、人間から見て自然な連想を行うためのアルゴリズムと、現在の文脈を得る、あるいは追いかける能力を計算機に付与する必要がある。前者は人工知能の研究の1テーマであり、後者はマンマシン・インタフェースの問題である。ユーザと計算機の相互理解を可能にするには、少なくともこのような機能が必要となる。一般に相互理解には時間がかかる。長年付き合っている人間同士でも新しい発見が常にある。したがって、計算機と人間は一生つき合う方がよい。物心がつき次第、一人一人に計算機を与えれば、個人と共に成長し(学習し、知識を獲得し)、記憶を共有できる生涯パートナーとしての個性をもった計算機が実現する。



### 3. 計算機からユーザへ

計算機は、人間が与えた指示に従ってではあるが、情報処理を行っている。この点、他の道具と大きな違いがあるが、人間が一方向的に計算機に働きかけているという点では、まったく同じである。しかし、「情報」を取り扱っているため、計算機によってはあたかも知性をもっているかのように振舞うものがあり、これが状況によってはユーザを混乱させるものになる\*\*。理由は、簡単である。メッセージがあらかじめプログラムに埋め込まれた固定のもので、ある特定のレベルの人にしか適切でないからである。世の中には「メモリが足りません」といわれて何をすればよいかわかる人も多いが、何もできない人もそれ以上に多い。このように従来の計算機に欠けているのは、自分自身をユーザに分からせようとする姿勢(機能)である。

相互理解には計算機から人間への働きかけが必

\*人工知能の分野では、Truth Maintenance System, Assumption-based TMSなどと呼ばれる知識の整合性の管理の研究が行われており、この方向の歩を示唆している。  
\*\*余談だが某有名ワープロソフトで、文書Aを編集して保存した後、文書Bを読み込もうとすると「作成中の文書が失われますがよろしいですか?」と聞いてくる。これを文書Aが「この世の中から失われる」(実は実行中のワープロソフトのバッファが消去されるのだが)と思ってしまう、何もできなくなってしまったという笑うに笑えない経験を身近に知っている。

要である。自分(計算機)を分かってもらうには自分が分かっていることについて説明するしかない。しかし、ただ説明するだけではちょっと気の効いたメッセージシステム以上にはなれない。説明には適切な質問が不可欠である。質問に対するユーザの答を通して、ユーザが何を考えているかを知ることができ、それがひいてはユーザが理解しやすい説明の種になる。また、説明には自己に関する知識が必要である。自分自身に関する正しい理解なくして、正しい説明はあり得ない。

人工知能の分野では、「理解」という言葉をキーワードとして、さまざまな研究が行われている。この種の研究で必ず問題となるのは、実用規模の実験をしようとすると膨大な知識を入力する必要があり、往々にして計算パワーがネックとなる前に入力作業が障壁になる。計算機自身に自分のことを理解させようとすると、この問題はより深刻になってくる。計算機そのものが複雑な上に、そこで走るアプリケーションは無数にある。それらすべてにわたって十分に理解できなければ、必ず説明に不具合が出てくるのが容易に予想される。

Cycのように常識を埋め込むことももちろん必要であるが、自己理解という観点からは次のような高級言語アプローチが魅力的である。つまり、ある言語でプログラムを書けば、プログラムを書いたことにもなるし、仕様書を書いたことにもなり、さらにそのプログラムを理解するための知識を書いたことにもなるという言語があれば、新しいアプリケーションについてはかなり問題が緩和される。このようなプログラミング言語を仮に説明指向プログラミング言語(自己理解言語)と呼ぶことにしよう。オペレーティングシステムや説明指向プログラミング言語処理系自体もこれで記述すれば、ソフトウェアに関する説明は閉じる。さらにこの言語でハードウェアをも記述できるようにすれば(当然、このためのシリコンコンパイラもこの言語で書いて……)、自己理解のための知識ベースとハードウェア、ソフトウェアを同時に得ることができる。このような言語の芽として関係を

宣言として記述することができる Prolog 処理系が挙げられるが、果たしてこのような夢のプログラミング言語は本当に可能であろうか。そして、この言語で学習を学習するプログラムを書くことができれば、自律的な自己成長も夢ではない。

高級言語アプローチは、すでに無数に存在する既存のアプリケーションには無力である。より現実的なアプローチとして、プログラム理解プログラムが挙げられる。プログラム理解プログラムは入力としてプログラムをとり、そのプログラムに関する知識ベースを出力する。この基盤としては、前述の説明指向プログラミング言語が使えるであろう。まず、既存のアプリケーションを説明指向プログラミング言語で書いたプログラムに等価変換する。もちろん、アプリケーションプログラムが(従来言語を使って)プログラミングされた時点で「説明」に必要なかなりの情報が欠落しているので、変換結果からただちに十分な「説明」が得られることはない。しかし、人間の専門家が後処理をする出発点としては悪くない。後処理で欠落した知識を補うことによって、既存のアプリケーションを説明するのに十分な知識ベースが得られる。さらに、変換後に後処理されたプログラムを多数集め、それらの共通点などから変換知識を抽出し、変換プログラムの知識ベースに加えていけば、やがては変換プログラムが説明に必要な知識を変換時に補っていけるようになるであろう。

#### 4. 知識の共有と成長

従来、人間が自分の知識を他の人に伝えるには、知識を明示的にせよ、非明示的にせよ、「表現」する必要があった。また、他人の知識を自分のものとするにも、「表現」されたものを經由しなければならなかった。「表現」には、表現能力、メディアや時間などのさまざまな資源が必要となる。また、他人に知ってもらうためにはその「表現」を流通させなければならない。そのため、一般に自分から見た知識の発信者は、ごく一部の有名人か身近な人、あるいはマスメディアに限られることとな

る。一方、表現された知識が役に立つためには、それが各自の頭の中で整理され構造化されなければならない。さらに、共通の理解を得るためには、構造化された表現やその処理の形態にも共通するものがなければならない。このような状況で計算機との対話を通して、ユーザと計算機の間に共通の知識が生まれる。

もちろん、人間の記憶ならびに思考のメカニズムが解明されていない以上、知識を完全に共有することは不可能である。過去30年の人工知能の研究で明らかになったことは、人間の知能の少なくとも一部は記号処理で模擬できるという、いわゆる「記号処理仮説」である。これが知能を表現するための必要十分条件であるかは分からない(最近のニューラルネットの研究の流れは、これに否定的な見解の立場)が、少なくとも十分条件であることは人工知能の歴史が示している。

しかし、知識を共有するとは、明示的に表示された記号としての知識を共有するだけでは不十分である。人間の理解の仕方に整合した情報の授受が必要で、記号はその一部に過ぎない。たとえば、拡散方程式を眺めれば粒子の分布が見えてくる、幾何の問題図形を眺めれば補助線が見えてくる、というような思考形態(イメージ推論と呼ぶことにする)を計算機が模擬できることが、非常に重要なことであるように思われる。つまり、実現メカニズムが人間と同じである必要はないが、記号レベルよりもっと深いレベルでの知識の共有を可能にすることができるかどうか大きな課題である。このようなイメージ推論の機能を付与することができれば、いわゆる「木を見て森を見ず」のような近視眼的な思考形態に陥ることなく、大局的のものごとを理解するといった人間の理解の仕方に整合した情報の授受も可能になるであろう。このような機能がいかなるメカニズムで実現されるかは自明ではない。盲目的にニューロ・コンピュータに飛び込むのは危険である。ビットイメージ表現と記号表現とが、記述量最小化基準のような合理的な基準で、自然にバランスし融合するような表現形式が考えられるのではなかろうか。

このような表現形式ならびに処理形式に関し、一定の技術的見通しが得られれば、対話の文脈を追いかけたり、自分について説明したりすることが可能となる。その結果、計算機とユーザ相互の理解を深めることが可能となり、やがては対話を通して、相互に知識を共有できるようになるであろう。別のいい方をすれば、計算機とユーザの対話で話題にのぼった事項が、知識として相互に理解可能な形式で計算機に蓄積されていく。計算機がもつネットワーク機能を用いれば、このような知識を表現しなおすことなく流通させることが可能であり、計算機を通して個人レベルの知識を任意の人とやりとりすることができるようになる。

人間は、一生、学び続け、学んだことを子孫に残す。学習の効果は最初にどれだけ知っているかに依存することが分かっている。つまり、多くを知っているほど多くのことを速く学習できる。多くの研究がゼロからの学習に悲観的な結論を示している。さらに、我々人間も、他の人が知っていることは、独力で学ぶよりも、知っている人から直接学ぶ方がずっと効率が良いこともよく知っている(教育の効果)。先に述べたCycも、このことを反映している。計算機による効果的な学習には保持すべき知識に閾値があり、それを越えなければ学習の効果は期待できない。また、すでに我々が知っていることを再発見してもあまり嬉しくはなく、どうせ計算機で学習するなら、まだ我々の知らないことを学習して欲しい。それでCYCは我々がすでに知っているこの世の常識を手で入力しているのである。

計算機とユーザで知識を共有することが可能になり、かつ計算機同士がネットワークで直接、知識を伝達出来るようになれば、学習能力の飛躍的向上が期待される。計算機の処理速度は速い。それに24時間稼働可能である。さらに、種となる知識の量も閾値を越えている(もちろんゼロから出発する必要はない。このようなシステムが可能になる頃には、共有可能な知識ベースが利用可能であろう)。本人が知らない間に学習し、計算機から本人へのフィードバックが起り、相互に啓蒙

しあいながら成長する。ユーザは計算機が学習したことをすべて知る必要はない。必要なときに、分身としての計算機に聞けばよい。また、長く付き合っているため、計算機は持ち主の性格に同化する。つまり、持ち主と同じ個性をもつようになる。一を聞いて百を知るようになり、いちいち細かいことをいう必要はなくなる。いわゆる枯れた使い方ができるようになる。人間、誰しも年をとれば、計算機は苦手となる。しかし、その心配は不要であり、計算機の方が持ち主の能力の変化に適応してくれる。そのうち顔色を見て行動するようになるかもしれない。

## 5. 計生100年

知識の共有は、体験の共有でもある。また、体験の共有が相互理解を加速し、効果的な文脈の追跡を可能にし、新たな知識の獲得を容易にする。人間同士でも行動を共にすることにより、その過程あるいはその後において対話がはずむことはよくある。これは行動を共にすることによって、文脈の追跡(あるいは状況の把握)が行動と同時に進行し、相互作用の経過が必要でなくなるためである。実際、「同じ釜の飯を食った仲間」という言葉で表されるように、行動の共有は人間の世界では極めて重要な相互理解の方法である。したがって、生涯パートナーとしての計算機はその持ち主と体験を共有し続けることが極めて重要である。人間の場合、幼児から若年にかけての体験が、人格その他諸々の個性の形成に重要な意味をもつので、持ち主が生まれた時に計算機を与え、それを常に所持し、使い続ければ、体験の共有としては十分であろう。

このような計算機の使い方では、信頼性が大きな鍵を握る。計算機の停止はすなわち体験あるいは時間共有の停止であり、相互理解の妨げとなる。端的にいうと、このような計算機は70~100年の間、十分な機能を保って動き続ける必要がある。そしてその間、持ち主のことを理解しようとし続ける。我々は、いまだかつてこのように長時間安定して動作し続けた電子計算機をもったことはな

い。将来、計算機科学はこのような計算機を実現可能としてくれるであろうか。つまり、いずれ「人生100年」に加え「計生100年」の時代になる。

持ち主の寿命と同じくらい長く動作できる性質のことを、仮に超信頼性と呼ぶことにしよう。超信頼性自体は、必ずしも単一の計算機で達成される必要はない。主眼は、あくまで持ち主との体験の共有が途切れないことである。超信頼性は、さまざまな機能を通じて実現され得る。1つは無停止性である。これは文字通り、体験の共有に空白期間を作らぬよう、決して動作が停止しないことである。無停止性については、不完全ながらフォールトトレラントシステムとして商用化されている。しかし、現状では定期的なメンテナンスを要しており、決して十分とはいえない。

無停止性と表裏一体をなすのが、自己修復性(機能)である。とくに、いま議論している計算機は、万人が所有することを前提にしており、かつ極めて複雑\*なものとなるため、前述のフォールトトレラントシステムのような人手によるメンテナンス作業は不可能であろう。ハードウェアの修復には内蔵の微小ロボットを使う方法や、ハードウェアに十分な冗長性を持たせておき、不良部分を捨てていく方法などが考えられる。人間の体内活動に重要なヒントが隠されているように思える。一方、ソフトウェアはどうであろうか。ソフトウェアの自己デバッグや、バグはゼロにできないという立場に立つにしても、十分低い確率でしかハングアップしないようなソフトウェアの開発法が従来にもまして重要になってくる。前述した説明指向プログラミング言語が解決の糸口を与えてくれるかもしれない。

また、数10年も使っていれば、その間に大きな技術革新が何度もなされるであろう。そのようなときに自動的に最先端技術を取り込むような自己改良性を考慮した設計になっていなければならない。現在でさえ、計算機世代ごとの互換性は大きな問題である。数10年にわたって互換性を維持し、

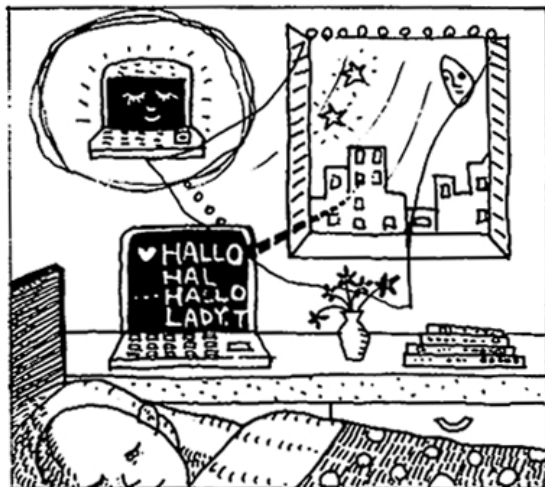
\*少なくとも人一人、一生分の記憶容量を持ち、持ち主のことを理解しようとする高機能なものである。

あるいは非互換性を克服するような技術を開発することは極めて重要である。人間の進歩は遅いが、計算機の進歩は速い。もし、ハードウェア・アーキテクチャを劇的に、あるいは100歩譲って、わずかにでも変えねばならないような状況が来た時に、どう対処すべきか、知識の伝承という点からも、非常に大きな課題である。

以上のような諸問題を解決できたとしても、破壊的な局面を回避するために計算機内に、蓄積された「記憶」のバックアップをとっておく必要もあろう。巨大なバックアップセンタを作り、そこに各人の計算機が自分で勝手にバックアップする方法が容易に思いつくが、計算機の数を考えると、この方法はおそらく不可能であろう。むしろ、各人の計算機同士に相互に十分な冗長性を持たせ、バックアップを広く分散させておくことになるであろう。持ち主を「理解」している計算機を通じて、個人レベルの有用な知識を直接交換し、知識を永久利用へと進化させる方向はすでに述べた。このような知識交換と冗長分散にもとづくバックアップには、当然大容量かつ場所を選ばない通信経路が要求されるが、我々はすでにこの雛型をパソコン+モデム+携帯電話にみている。

## 6. おわりに

以上、技術的裏付けが薄いことは承知の上で主観を中心に、我々が理想とする計算機の未来像について述べてきた。一言でいえば、分身として生涯をとらえさせる計算機である。この中で、技術的課題に関連するキーワードを列挙すると、超大容量化と超高速化を前提に、空間指向パラダイム(記憶、連想、学習、知識獲得、イメージ推論)、説明指向パラダイム(説明、質問、自己理解、相互理解)、超信頼性(無停止、自己修復、自己改良、超大容量通信経路)の3点に要約されよう。これらの技術課題に対し、本特集の他の記事が解決の糸口を与えてくれるかどうかは分からない。どれにも共通の課題なのかもしれない。もし、超信頼性が達成されれば、そのうち「〇〇家に代々使えてきた計算機一族」なる言葉が出ることも夢ではな



くなるであろう。

最後に、このような計算機システムが社会に与える影響は、計り知れないものになることに言及しておきたい。社会のインフラストラクチャに計算機が組み込まれ、人間生活の必需品、それも社会生活を営む上での知識の媒体として使われるようになると、それが社会の変革の制約にもなる。つまり、このような計算機システムに社会そのものが同化せざるを得なくなるかもしれない。知識の共有を真剣に考えないと、そこでは持てる者と持たざる者の差は拡大する一方になる。知識が安く手に入るようになり、人類全員が知っている知識を最大限に享受できるような社会が実現するかどうかは、どれだけ真剣にこの問題に取り組むかにかかっている。

## 文献

- 1) Ritchie DM and Thompson K: The UNIX Time-Sharing System, The Bell System Technical Journal, Vol.57, No.6, 1978
- 2) 牧之内顕文: 情報処理機械から情報記憶機械へ, 情報処理, Vol.32, No.1, pp. 4-7, 1991
- 3) Lenat DB, Guha RV, Pittman K, Pratt D, Shepherd M: CYC: Toward programs with common sense, Communications of the ACM, Vol. 33, No.8, pp.30-49, 1990
- 4) 第7回汎用コンピュータ・ユーザ・センサス, 日経コンピュータ, No.235, 1990.
- 5) 集積規模の拡大に限界はないが、微細化は加速度的に難しくなる, 日経エレクトロニクス, No.491, 1990.