

第5世代コンピュータへのアプローチ

電子技術総合研究所 瀧 一博
Kazuhiro Fuchi

はじめに

現在通産省で検討されている「第5世代コンピュータ開発計画」のポイントは

- (i) 知識情報処理を指向
- (ii) それを支えるマシン群(推論マシン, 知識ベースマシン等)を構築
- (iii) それらの上に、問題解決機能, 知的会証機能, 知識ベース機能等を実現
- (iv) 代表的な応用例を選び、その有効性を実証

ということになると思われる。

90年代の情報処理を目指すものとして、これが(国家プロジェクトとして)適切なゴール設定か否か、各種の観点からさまざまな議論がありうるであろう。

これが90年代のニーズに沿うものであるか、それはどういうニーズであるか、それにどの程度の広さとレベルで応えるものであるか、他にもっと大事な問題はないのか、その検討も必要であり、まに行われている。

一方、それとともに、また、その前提としても重要なのは、その技術的な実現可能性であろう。どの程度の機能のものが、どの程度の期間と努力によって可能になるであろうか。それは、情報処理とコンピュータの革新をもたらすものであるか、それに対する断定的な答は不可能であるにしても、その見通しを支える状況証拠はいくつと出てきていると思われる。

とはいへ、このような目標設定が「野心的」であることは大方に指摘されるところであろう。その方向性について大方の同意が得られるにしても、

- (i) 情報処理の各分野での強力な基礎的研究の展開
- (ii) さまざまな試作と実験のくりがえしによるフィードバック
- (iii) それらを支える、研究開発のための環境条件

が不可欠の前提というべきであろう。

I. 統一的イメージの可能性——諸研究分野の相互関連の進展

上に述べた目標は、単なる願望によってではなく、これまでの情報処理研究の進展自体によって支えられるべきものであろう。少なくともこの10年、情報処理の各研究分野で大きな進展があったと見られる。それぞれに魅力的な成果があったが、それとともに注目すべきは、独自の動機に発したと思われる多くのものが相互に深く関連し合ってきているという事実である。このような進展が今後ますます進むと見れば、その先に統一的なイメージを想定することが可能であろう。

情報処理の(基礎的)研究分野として、たとえば、

- (i) 人工知能研究, 知識工学
- (ii) ソフトウェア工学(プログラミング方法論, 関係データベース論)
- (iii) アーキテクチャ研究
- (iv) 言語学(生成変換理論, モンテギエ理論)

を考えてみる。それぞれにおける進展とともに、それら相互における交流の深化が最近の顕著な現象である。

(一例) Dijkstraのアソシオン¹⁾²⁾

これは(超)並列操作を意識した(超)高級言語である。その上でプログラムの導出(検証)が試みられている。これは1972年にDijkstraによって発想されたもので、現在、Meadと協同でVLSIをベースにした超並列マシンのアーキテクチャの研究が進められているとのことである。この言語は、知識工学の分野でよく使われている「プロダクション・システム」(PS)に酷似している。「我々は、その時点では、同様のプログラム手法が人工知能のグループですでに数年間用いられていたということを知らなかった」¹⁾

(人工知能と知識工学)

70年代の人工知能研究は、「自然言語と知識」という問題意識ですすめられてきたといえよう。70年代の後半には、その手法の応用が注目されるようになり、「知識工学」の提唱があった。

知識工学では「専門知識」の組込みによるコンサルテーションシステムの構築が現在主眼になっている。そのためには専門知識自体の整理が重要であるが、それを支えるのは、知識の表現と利用法、知識の獲得法等である。対象分野としては、医療診断、化学物質の構造決定、油田探査のための信号解析、遺伝子工学の実験計画、コンピュータの故障診断、超LSI設計等、多彩なものが試みられつつある。その考えは将来、教育や経営等へも広がっていくことが予想され、情報処理の新分野となっていくであろう。

他の研究分野との相互関連をみる一つの接点は、人工知能用プログラミング言語である。現在、Lisp, Prolog等が使われ、その上にPSやもつと高水準の知識表現言語が試みられている。それらの言語は、ソフトウェア工学とも関連をきており、またコンピュータの新アーキテクチャの研究とも関連し合ってきている。

(ソフトウェア工学)

70年代において、新しいプログラミング・スタイルの提唱があり、プログラムの検証や合成(変換)の研究が進んだ。それらを基礎づけるものとして、プログラムの意味論も確立されてきた。これらは、今後の情報処理技術の基礎的要素として重要である。

このような流れから出てきた「関数的プログラミング(言語)」は、新アーキテクチャとしてのデータフローマシンやリダクションマシンの考えと深く結びついている。

一方、(関係)データベース論も70年代の大きな流れであった。それ自体として深く研究されてきたが、データベース研究は他分野とも関連をもち始めている。データベースの高度化は、人工知能研究での知識表現の問題に近づいている。また、データベースマシンは、新アーキテクチャの原動力の一つになりつつある。

(アーキテクチャ研究)

新アーキテクチャとして現在注目されているデータフローマシンやリダクションマシン等は、ソフトウェア工学における関数的プログラミングと深く結びついている。それは並列アーキテクチャの流れであるが、一方、これまで提案され、必ずしも活かされていないアイデアもまた、他分野との結びつきという新しい観点から統合的に生きてくるであろう。

(モンテギユ理論)

これは特殊な話題であるが、他分野との関連を見る一例になる。モンテギユ理論は、自然言語の構文論と意味論を統一する枠組みとして、70年代に注目されるようになったものである。言語学の主流である生成変換論とも結びつきを深めつつある。

この理論は、プログラミング言語の意味論の構成とよく似ている。実の所は、同じ思想的基盤から生れたものといえるものである。また、この理論の基礎になっているのは「内包論理」であるが、これは、関数型の高階様相論理である。プログラミング言語意味論への内包論理の適用の研究も出てきている。またデータベース論への利用の研究も出てきている。

様相論理とプログラムあるいはデータベースの関連の深化は最近の一つの傾向でもある。

II 新マシンの可能性と普遍性

いまかりに、各分野での(専用)マシンの可能性を考えてみることにする。そのイメージは、各分野での(プログラミング)言語から得られるであろう。

(i) AIマシン(知識工学マシン)

← Lisp, (Planner, Conniver), PS, Prolog

(ii) ソフトウェア工学マシン

(抽象データ型) ← Simula, CLU, Smalltalk

(形式仕様, 検証, 変換) ← 述語計算, ラムダ計算 (Prolog, Lisp)

(プログラム意味論) ← ラムダ論理, (述語論理) (Lisp, Prolog)

(関係データベース) ← 関係代数, 関係計算 (Prolog)

(iii) 自然言語処理マシン

← ATN, Lingol (Lisp), DCG (Prolog)

これらはそれぞれ全く別のものであるであろうが、たとえば Prolog をとり上げてみる。これは、自然言語の処理(構文解析, 意味処理等)に、少なくとも既存の他言語より適している。これは知識的処理(問題解決, 推論)に適している。ソフトウェア工学において、検証や変換(合成)システムを作るのにもよいであろう。また、プログラミングとデータベース操作の統一的言語の候補としても考えられる。そこで、Prologマシンは、各分野に有用な新マシンと考える出発点になりうるだろう。

(核言語の設定)

「オ5世代コンピュータ計画」では、ハードウェアとソフトウェアのインターフェースとして「核言語」の設定を予定している。それは、Prologを出発点とし関数型言語(Lisp等), 抽象データ型言語等の要素を加味した「論理型言語」になると想定されている。

このような言語の想定は、新しいハードウェアやソフトウェアの開発にとって不自由な制約になるであろうか。ソフトウェアにとって、それはいままでより高水準で使いやすいユーザ向言語のベースになりうるだろう。ハードウェアにとつ

ては、新アーキテクチャを考ふる指導理念になりうるだろう。

新アーキテクチャとして提案されているデータフローマシンやリダクションマシンの考へ、あるいは、これらから出てくるかもしれないアイデアは、その中で自然に所を得てくるものと思われる。

一歩下って逐次型の実現法を考えてみよう。この場合でも、これまで提案され現行のマシンでは必ずしも活かされていない多くの手法（スタック、タグ、ハッシュ、---）が有機的に活用されることになろう。

新マシンの実現可能性を支える技術的な要素の一つは、VLSIの進歩の予測であるが、それとともに必要なことは、それを導く原理である。それを「論理」に求めることは、別に奇異なことではないであろう。いまのコンピュータも論理機械の一種である。ただ、それはきわめて特殊な論理系にもとづいている。それをもっと普通の論理に戻してよい状況が情報処理の各分野から生れてきていると考えられるのである。

Ⅲ 新マシンの有用性 — 知識情報処理へのツール

知識情報処理というような高度の目標をかがげるとすれば、それを支える強力なツールが望まれるのは当然であるが、それとともにその実現可能性も出てきていると考えられる。

(3乗則) ソフトウェアの開発費用は、プログラムのステップ数の3乗に比例し用いる言語の使いやすさ(水準)の3乗に反比例する、という説がある。

これが成立すると、コンピュータアーキテクチャがもつセマンティックギャップについても同様のことが言えるであろう。

ソフトウェア工学におけるいろいろな提案が必ずしも実用に近づかない一因はそこにあるであろう。

たとえば、検証という問題を考えてみる。それ自体の研究の進展も必要であるが、対象言語との距離も大きな問題である。その言語の水準はマシンアーキテクチャの制約を受けざるをえない。中間に(超)高水準言語を設定する(*divide and conquer*)としても、その処理系の作成と実行効率が問題になる。また、検証系自体と作成と実行効率の問題がある。ギャップの大きさはこの場合何乗くらいで効いているのだろうか。

ソフトウェア工学の十分な展開と実施は、知識情報処理の実現の不可欠の前提である。それとともに、その前提の実現のためにも、ハードウェアまで含めた体系の見直しが必要と考えられる。ソフトウェア工学の研究自体がそれを要求しているように思われる。

引用文献

- 1) M. Rem, *The Closure Statement: A Programming Language Construct Allowing Ultra-concurrent Execution*, JACM Vol. 28 No. 2 (April 1981). pp. 393-410
- 2) M. Rem, *Associon: A Program Notation with Tuples instead of Variables*, ACM Tr. Programming Languages and Systems, Vol. 3 No. 3 (July 1981) pp. 251-282



本 PDF ファイルは 1982 年発行の「第 23 回プログラミング・シンポジウム報告集」をスキャンし、項目ごとに整理して、情報処理学会電子図書館「情報学広場」に掲載するものです。

この出版物は情報処理学会への著作権譲渡がなされていませんが、情報処理学会公式 Web サイトに、下記「過去のプログラミング・シンポジウム報告集の利用許諾について」を掲載し、権利者の検索をおこないました。そのうえで同意をいただいたもの、お申し出のなかったものを掲載しています。

https://www.ipsj.or.jp/topics/Past_reports.html

過去のプログラミング・シンポジウム報告集の利用許諾について

情報処理学会発行の出版物著作権は平成 12 年から情報処理学会著作権規程に従い、学会に帰属することになっています。

プログラミング・シンポジウムの報告集は、情報処理学会と設立の事情が異なるため、この改訂がシンポジウム内部で徹底しておらず、情報処理学会の他の出版物が情報学広場 (=情報処理学会電子図書館) で公開されているにも拘らず、古い報告集には公開されていないものが少からずありました。

プログラミング・シンポジウムは昭和 59 年に情報処理学会の一部門になりましたが、それ以前の報告集も含め、この度学会の他の出版物と同様の扱いにしたいと考えます。過去のすべての報告集の論文について、著作権者 (論文を執筆された故人の相続人) を探し出して利用許諾に関する同意を頂くことは困難ですので、一定期間の権利者搜索の努力をしたうえで、著作権者が見つからない場合も論文を情報学広場に掲載させていただきたいと思います。その後、著作権者が発見され、情報学広場への掲載の継続に同意が得られなかった場合には、当該論文については、掲載を停止致します。

この措置にご意見のある方は、プログラミング・シンポジウムの辻尚史運営委員長 (tsuji@math.s.chiba-u.ac.jp) までお申し出ください。

加えて、著作権者について情報をお持ちの方は事務局まで情報をお寄せくださいますようお願い申し上げます。

期間：2020 年 12 月 18 日～2021 年 3 月 19 日

掲載日：2020 年 12 月 18 日

プログラミング・シンポジウム委員会

情報処理学会著作権規程

<https://www.ipsj.or.jp/copyright/ronbun/copyright.html>