第5世代コンピューター研究・開発の現状と展開

横井俊夫[†]

第5世代コンピュータープロジェクトの目標と基本的アプローチを概観し、研究・開発の現状と今後の展開を紹介する。視点を中心部に絞り、大きく6つの研究・開発の分野について述べる。分野は言語、すなわち核言語（第0版）、核言語（第1版）、知識表現言語、自然言語、論理型プログラム言語、数学言語の6つを手掛りに分類する。

1. ま え が き

言葉・文字、印刷術、電気通信技術、そしてコンピューターと、人類はその永い歴史をかけ、自らの情報処理能力を強化、拡大する手立てを次々と生み出してきた。そして、情報処理能力の飛躍は、様々の葛藤を経ながらも、次には、社会文化の飛躍へと結びついていった。

これから10年ないし20年をかけて、まだ別々の仕組みの上に成り立っている情報処理の道具やシステムが、それぞれの良さを残しつつ、その本質的な部分で深くかかわり合い、融合し、統合化された新しい情報処理システムへと脱皮を開始する。これが、昨今喧伝される高度情報化社会への胎動である。このようなコンテクストの中に、第5世代コンピューターも位置付けられる。統合化された新しい情報処理システムの主役はコンピューターである。

この主役を演じるうするためには、現在のコンピューター技術にもう一段大きな飛躍が必要である。書物が持つ手に素直になじみ、紙面の構成や、ページをめくる感覚が、我々に強い知的共感を呼び起こすように、コンピューターも素直に人間の情報処理感覚になじみ、親しみを呼び起こすものへと成長しなければならぬ。この飛躍への企てが、第5世代コンピュータープロジェクトである。

以下にこの企てを、現状と今後の展開を中心に紹介する。2章では、本プロジェクトの目標とするところ

と、基本的なアプローチの方法を簡単に述べる。3章では、“言語”をキーワードにして、6つの代表的な研究・開発の分野につき詳述する。

2. 目標とアプローチ

本プロジェクトの目標とするところは、第5世代という、現在のコンピューター技術とは明瞭に一線を画しうだけの、新しい世代を作成しうるに足るコンピューター技術を、ソフトウェア、ハードウェア、そしてその裏付けとなる理論までを含めて総合的に研究・開発することである。

それらの技術の妥当性・有効性を実証するためのいくつものプロトタイプシステムを具体的に作成しては見せるが、設定されたある目標値を満たすシステムを作って目標が達成されたという性質のプロジェクトではない。したがって、表1に掲げる目標像にしても、目標システムというより、研究・開発される様々

表1 目標像

総合機能	知識情報処理
基本応用	エキスパートシステム 機械翻訳システム 知的VLSI-CADシステム など
インタフェース	音声理解機能、図形・画像理解機能、 自然言語理解機能
基礎システム	知的プログラミングシステム
基本機能	知識ベース機能、推論・問題解決機能
プログラム言語 (核言語)	論理型言語
アーキテクチャー	高級言語マシン、関係データベースマシン 超並列処理マシン（推論マシン・知識ベースマシン）
素子	VLSI技術

[†] 電子技術総合研究所（現在、財団法人新世代コンピュータ技術開発機構に出向中）

“The Current Status and Future Trends of Fifth Generation Computer System Project” by Toshio Yokoi (Institute for New Generation Computer Technology, Tokyo)

の技術が納まるべき枠組ともいうべきものである。

この枠組を埋め尽すだけの技術と、それを具体化してみせる様々のシステムが研究・開発され、それらに基づいて種々の商品となるコンピューターが1990年代にいくつも登場する。これがプロジェクトの波及的な成果である。

より高踏的な立場に立って、とかく応用技術に終始してきた日本の技術開発の立場に基礎的・独創的技術開発の風土を作り出し、単に一国の産業の利に益するのではなく、人類の共通資産ともいうべき創造的な先端技術に積極的に投資すること、これも本プロジェクトを律する基本精神であり、目標である。

枠組としての目標像(表1)を詳述する。第5世代コンピューターの総合的な機能は知識情報処理である。処理の基本単位が、データやプログラムではなく、それらがもう一段高位に統合化された“知識”というものに移る。数値処理でなく、記号処理を土台に組直された知識処理のシステムとなる。

基本技術への具体的な経験を得、その有用性を実証するものとして、代表的な応用が基本応用として試みられる。現在、世の中で実用化が叫ばれているエキスパートシステムも、整った内部メカニズムの上に、もう一段高度な機能を発揮しうるものが、適切な分野を選んで試みられる。機械翻訳システムも意味処理を踏えたものが取り上げられる。特定の深い推論を駆使したり、多量の知識の検索を必要とする例題として、VLSI設計や論理設計等のCADシステムが試みられる。

人間とコンピューターとのインタフェースは、人間が人間に対する時のインタフェースに極力近づけるのが原則である。言葉による話し合い、図、表を用いての説明、人間の知覚に素直になじむマンマシンインタフェースが必要である。ただし、音声処理、図形・画像処理については、ローレベルの処理に多くの努力を割くことはせず、むしろ記号的な処理への結びつけの部分に、すなわち理解メカニズムの実現に重点を置く。

この記号的な処理への媒介として重要なのが、自然言語(日本語)処理である。これに関しては、高機能な構文解析技術の確立は当然として、本格的な意味処理、さらに文脈理解と総合的な技術の研究・開発を目標にする。自然言語処理は、出来上ったシステムでは人間とのインタフェースに位置することになるが、研究・開発の過程ではシステムの基本機能にかかわる重要な役割を担う。

知識情報処理の典型的応用でもあり、研究・開発の

支援システムともなり、しかも基本機能に深くかかわるものとして知的プログラミングシステムがある。基礎システムとしてその重要性を強調してある。現在まだ実用化されるに至っていないプログラム仕様記述・検証技術を様々の観点から取り上げ、この技術を中心にして、統合的な知的と呼ぶにふさわしいプログラミングシステムの開発を目指す。ソフトウェア危機論議への沈着な取組みが展開される。

基本機能は、知識を蓄わえる知識ベースと、様々な問い掛けに答える推論・問題解決のメカニズムである。基本機能は、一般性を重んずれば低い能力となり、特殊化すれば狭いながらも高い能力を得るという人工知能の原則に、いかに適格に折り合うかというのが鍵である。最も一般的なものとして、論理型言語(Prologで代表される)を設定し、高機能だが特殊なものとして、“定理を証明する”、“プログラムを検証する”、“データベースの整合性を保証する”、“自然言語を理解する”、“音声・図形画像を理解する”、“英語を日本語に翻訳する”、“VLSIを設計する”等々が試みられる。

プログラム言語は、それぞれの用途に向け様々なものが用意されるが、すべてが論理型言語という基本スタイルを何らかの形で継承する。それらの言語の中で基本となるのは、ソフトウェアシステムの研究・開発に用いられるシステム記述言語と、ハードウェアとのインタフェースを司る核言語(機械語)である。

核言語は、体系立った計算メカニズムの上に設定され、個々のアーキテクチャー設計の過程で微調整される。システム記述言語は、適切な文法上の飾り付けが施され、オペレーティングシステムの提供する機能等を取込んだ高レベルの言語となる。

ハードウェアとしては、推論マシン、知識ベースマシンと呼ばれるものが目標となる。それらのマシンを基本的に支えるアーキテクチャーは、高級言語マシンアーキテクチャー、関係データベースマシンアーキテクチャー、超並列処理マシンアーキテクチャーの3つである。高級言語マシンアーキテクチャーの軸に沿って現状技術を素直に延長したところに、逐次型推論マシンが設定される。このマシンのシステムアーキテクチャーとして考慮されるべきは、パーソナル化の観点である。

関係データベースマシンアーキテクチャーの軸を中心に、他の2つのアーキテクチャーを加味して知識ベースマシンが設定されるが、関係データベースマシンそのものとしての高機能化(高並列処理化)、またはパーソナル化という観点からの追及が、具体的なアプロ

一チとしては重要である。

超並列処理マシンアーキテクチャの軸を中心に、並列型推論マシンが設定される。汎用の並列処理マシンという難課題ではあるが、この際重要なのは、単なる処理速度の向上だけではなく、高信頼性の実現、新しいハードウェアの構成技法の実現等の観点も、等しく考慮されねばならない。

本プロジェクトでは、新しい素子技術の開発は目標ではない。むしろ、まだ未開拓のシリコンベースのVLSI技術を十分に使い切ることが課題である。逐次型推論マシンのワンチップ化、並列型推論マシンや知識ベースマシンを構成する様々な専用チップの試作を通じて、VLSIアーキテクチャの追及が具体化される。J-J素子、GaAs素子や新機能素子の研究・開発は、他プロジェクトで行われており、それら成果の活用は当然予定されている。

以上の目標像に向かってどうアプローチするか。アプローチの基本ストーリーを説明する(図1)。

知識を蓄え、理解し、処理するメカニズムを計算機に植え付けるのであるが、人間が扱っている知識には様々なものがある。その中で、第5世代コンピュータとして取り上げ得るのは、人間がすでに体系立ったものとして整理しているものである。知識を表現し、扱う言語系が用意されるか、知識に対する教科書や手引書がきちんと用意されるかによって、知識は体系立ったものとなる。そのような言語の代表として自

然言語(日本語、英語)、数学言語(論理式、数式)、仕様プログラム記述言語、知識表現言語が取り上げられる。

自然言語は、人間が知識を記号化して表現する最も一般的な言語であり、数学言語は、関係概念や集合概念や数概念等、やはりすべての知識に共通する基本概念を表現する言語である。これら2つの言語は、計算機の出現以前に長い時間をかけて形成されてきたものであり、人間の知的機能の反映である。仕様・プログラム記述言語は、計算機の出現にともなって生み出されたもので、手順的な知識の表現を司る言語である。

教科書や手引書という形で整理されている様々の専門分野の知識を整理し、計算機処理可能にする言語が知識表現言語である。これらの言語を処理・理解するメカニズムを計算機に持たせることが、知識処理をする計算機を実現する基本ストーリーである。

これらの言語は、互いに深くかかわり合っている。したがって、処理・理解するメカニズムも高い共通性を持つ。また、極力共通のメカニズムへと持ち込むのが、研究・開発の基本姿勢である。その共通のメカニズムの第1次近似ともいえるべきものとして、論理型言語(Prologによって代表される)を設定する。この論理型プログラム言語を用い、先の4つの言語の処理・理解システムを、段階を踏みながら様々の側面から実験を繰り返しつつ実現していく。あるレベルの処理・理解システムが得られると、それを土台にして、より具体的な応用例に対し実験システムを組み上げ、得られた緒機能の有効性を検証する。この論理型言語、4つの言語の処理システム、応用システムとこれらにわたる様々な実験サイクルを経ながら、一步一步最終目標に向かって収斂させていく。

論理型言語(ハードウェアとのインタフェースとしては、核言語と呼ばれる)を効率良く実行する環境を提供するという観点に沿って、アーキテクチャの研究・開発のストーリーが組まれる。パーソナル型高級(論理型)言語マシンとしての逐次型推論マシンの実現がその第一歩であり、次に高機能関係データベースマシンの実現がそれに続く。

それらのマシンは、プロジェクト前期中に基本ソフトウェアも含め実用規模のものが研究・開発され、プロジェクト中期からは非常多数台の逐次型推論マシンと計算機ネットワークという土台の上に、プロジェクトの本格的推進のためのインフラストラクチャーを上げる。

研究・開発ツールとしての利用経験を積み重ねなが

図1 アプローチの構図

ら、逐次型推論マシンや関係データベースマシンは改良、拡張され、高機能化や小型化等の方向に沿って自立した歩みが続けることになる。

超並列処理マシン実現に向けても、何段階かのステップが踏まれる。接続される要素プロセッサの台数、要素プロセッサの基本メカニズムの2つの要因で段階が切り分けられる。接続台数が少なく、基本メカニズムが逐次型推論マシンに近いものは、プロジェクト中期中には実用規模のものが実現される。接続台数が非常に多く、基本メカニズムも一新したもののについては、様々のシミュレーションを通じ実験が繰り返され、理論的な研究とも並行して、プロジェクト全期間を通じ、今までになかった新しいコンピューターアーキテクチャーを目指し、着実な研究・開発が展開される。

3. 現状と今後の展開

目標とアプローチの筋書きに沿って、多数の研究・開発課題が設定され、進められている。これらすべてを等しく紹介するというのは本稿の目的ではない。第5世代コンピューターの骨組と今後の展開の基調に焦点を合せ、中心部分にしぼって紹介を行う。研究開発の分野を大きく6つに分け、言語を視野の中心に置いて説明する。ハードウェアよりソフトウェアに重点を置き、具体的な応用に関する紹介は大きく割愛する。

応用システムの研究・開発による具体的な検証と経験の蓄積の重要性はいうまでもなく、これらに関しては別の機会、あるいは他の紹介を参照されたい。なお、以下の6分野の説明は言語にポイントを置き、その周辺をめぐる形で説明を行う。研究・開発課題をすべて網羅したものではないことを注意されたい。

3.1 核言語（第0版）と記号処理

核言語は、ソフトウェアとハードウェアのインタフェースを規定するもので、第5世代コンピューターの機械語というべきものである。アーキテクチャー研究の出発点であると同時に、実際のソフトウェア作成に用いられる様々のプログラム言語が、コンパイルされる対象言語となる。そこで、プロジェクトの中核となる言語の意味からこの名前がつけられた。核言語は、プロジェクトの前期（昭57～59年度）、中期（昭60～63年度）、後期（昭64～66年度）のそれぞれのスタート時点で、第0版、第1版、第2版が設定される。

これらの3つの版ともに、論理型言語を母体とするが、その基本計算メカニズムは大きく異なる。第0版は逐次計算モデルに基づき、第1版、第2版は並列計

算モデルに基づく。第2版は、第1版に比べ基本メカニズムに大きな差異はなく、第1版の使用試験により有効性の明らかになったものを組み込むという形でまとめられる。現在のところ知識表現向きの機能等が想定されている。

核言語第0版、通称 KL-0 は、前期の出発点、すなわちプロジェクトの出発点ということで、既存の言語やシステムを参考に、できるだけ安全側に見込むという形で設計された。新しい高レベルの機能を追うより、むしろ整理した低いレベルの記述能力を中心にして汎用性を失なわぬようにすることに重点が置かれた。

Edinburgh 版 Prolog 等の言語仕様と実現技術を参考に、また広くは Lisp 等の記号処理用言語の経験も加味して設計が行われた。入出力やデータベース操作や、処理系への指示等、オペレーティングシステムにからむものは除き、現存 Prolog をサブセットとして含み、次のような機能拡張が行われている。

① 遅延評価（バインドフック）や拡張カット等の制御機能の拡張と整理。

② ベクタ、文字列等のデータ構造の拡張と整理

これらの拡張によって、処理効率をも考慮に入れうる高い記号処理能力を持つ言語が設定された。

この KL-0 を機械語とする逐次型推論マシン（表2）が設計、製作されている。このマシンのハードウェア部は、PSI (Personal Sequential Inference Machine) と呼ばれ、KL-0 向きのマイクロプログラム制御方式のパーソナル型コンピューターで、図2にハードウェア構成を、表3にハードウェア諸元を示す。昨年12月に1号機が完成し、今年2月に2号機、3月までにファームウェア（約13K）が完成し、5月末に4台が製作され、さらに9月には約10台が製作され、今年度中には研究・開発の要所に配置される予定である。

逐次型推論マシンのソフトウェア部は、SIMPOS (SIM Programming and Operating System) と呼ばれ、パーソナル型の良好なプログラム作成環境を提供すべく設計されている。図3に SIMPOS の構成図を示す。昨年度より設計を開始し、現在、スーパーバイザ

表2 逐次型推論マシン

基礎システム	SIMPOS, ESP
基本機能	
プログラム言語 (核言語)	KL-0
アーキテクチャー	SIM (PSI) (逐次型推論マシン)

図 2 ハードウェア構成 (逐次型推論マシン)

表 3 ハードウェア諸元 (逐次型推論マシン)

項 目		仕 様	
方 式	実 行 方 式	論理型命令の直接実行方式	
	語 表 現	40ビット/語(タグ部8ビット, データ部32ビット)	
	構 造 体 表 現	structure sharing方式	
	タ グ 種 類	64種	
	組 込 述 語 数	約150種	
	ガーベージコレクション方式	マークビットによるスライディングコンパクション方式	
ハ ー ド ウ ェ ア	主 記 憶	データアクセス幅	40ビット (タグ部8ビット, データ部32ビット)
		アクセスタイム	1.2 μ s/4 words
		サイクルタイム	1.2 μ s/4 words
		アクセス方式	ニブルモードによる語直列アクセス
		容 量	最大16Mwords
		エラー検出・訂正方式	7ビットECCによる1ビット訂正, 2ビット検出
	C P U	制 御 方 式	マイクロプログラム制御
		制御記憶容量	WCS : 64 bits $\times$ 16 kwords
		マシンサイクル	200 ns
		実 行 速 度	約30kLIPS
		内部データ幅	データバス32ビット, タグバス8ビット
		レジスタファイル	1 kwords

図 3 ソフトウェア構成 (逐次型推論マシン)

一部、カーネル部の実機上でのデバッグが完了したところである。他の部分は、ほぼコーディングが終り、Dec 2060 上のシミュレーター上でデバックを行っている。

さらに新しい追加機能の開発を並行させつつ、本年 10 月には 0.5 版ともいべきものを完成し、本年度末には第 1 版を完成させ、広く配布する予定である。

SIMPOS は、単一言語向、パーソナル型ということで、従来の大型機の実オペレーティングシステムの発想とは大いに異なる。1 つの言語 (ESP) を中心にプログラミングシステムからオペレーティングシステム

まで、さらにはハードウェア部に至るまで、一貫した思想のもとに、均整のとれた透明なシステムとして構成されている。

また、従来の大型機のシステムがハードウェア・資源の管理に多大の労力を割くものであったが、SIMPOS は、安くなったハードウェアより、より高価であるべき利用者（の頭脳）という資源をいかににより良く利用するかという観点に立っている。したがって、ハードウェア資源の管理は簡素簡明になり、プログラミングシステム、ウインドシステム等、利用者とのインタフェースにかかわる部分に新しい機能が盛り込まれている。また、日本語ベースのシステムとなっていることも重要な点である。

もうひとつの SIMPOS の特徴は、その構成法である。Smalltalk, Flavor System (Lisp マシン: Symbolics 社製等) などで試みられているオブジェクト指向に基づく構成法を採用した。今後のソフトウェア構成法の一大指針となると予想されるオブジェクト指向を論理型言語の上に本格的に採用したのは、この SIMPOS がはじめてであり、その成果のほどが注目されている。

現在のところ、コーディング量も約 1/3 位に短縮され、非常に見通しの良いモジュール化が可能になり、多人数による短期間での大規模なソフトウェアを容易にするとの実感を得ている。

もちろん、オブジェクト指向の難点である実行速度の低下という問題は、現時点では解決されたという状態ではない。しかし、ファームウェアでの機能強化やコンパイラの最適化によって、満足すべき程度に達しうるものと考えている。

論理型言語と SIMPOS の開発とのもうひとつ重要な観点は、論理型言語の汎用性の実証という意味合いである。論理型言語は、ユニフィケーションによる記号処理、非決定処理、非手続型、etc の言語であるといわれる。SIMPOS のようなシステムプログラムは、これらの特色をそれほど必要としない。もちろん論理型言語的パラダイムにのっとりた構成や機能という特色を有してはいるが、SIMPOS のほとんどのプログラムは、ユニフィケーションを単なる引数の受渡しだけに用い、決定的で手続的である。すなわち現在広く用いられている汎用言語の機能で満たされるものである。

引数受渡しは、ユニフィケーションの特殊な場合として含まれる。論理型言語では、決定的は非決定的の、手続的は非手続的のそれぞれ特殊な場合として含まれる。したがって、SIMPOS がシステムプログラ

ムとして十分に成功すれば、論理型言語が、従来の汎用言語を特殊な場合として含むうことを、すなわち新しい、より高度な汎用言語であることを実証できることになる。

このような SIMPOS の試みを支え、オブジェクト指向の機能を具体化しているのがシステム記述言語である ESP (Extended Self-contained Prolog) である。ESP は、シミュレーターとクロスコンパイラが、Dec 2060 上の Prolog を用いてインプリメントされ、これらは、SIMPOS の開発に不可欠のツールとなっている。

ESP は、多重継承の高度なオブジェクト指向機能を持ち、併せて高機能なマクロ機能も持つが、言語の基本メカニズムは一般の Prolog と同様に、ユニフィケーションと後戻りである。ただし、オブジェクトの状態は副作用を用いるため、論理型言語としての純粋性は若干損われるが、副作用の範囲をオブジェクトの状態だけに限定し、プログラムの見通しの良さを保証している。

ESP のシミュレーターやコンパイラをはじめ、マイクロプログラム用アセンブラやリンカ、KL-0 コンパイラなど SIM の開発ツールの多くが、Dec 2060 上の Prolog を用いて作成されており、それぞれかなりの量のプログラムながら、短期間に開発され、充分な有用性を提供していることは特記すべきことである。また、この SIM に並行して、より高速で大容量のバックエンド型の推論マシンや、SIM に接続される図形・画像用や日本語処理用の高機能入・出力機器も開発されており、本年度の終り頃には詳しい報告が得られる予想である。

SIM は非常に多数台製作され、計算機ネットワークで結ばれ、研究・開発環境を構成し、SIM 自身の改良・拡張も様々な形で続けられる。また、SIM の開発で得られた経験や成果の蓄積は、今後の数々の難テーマへの取組みに重要な役割を果たすであろう。

3.2 核言語（第 1 版）と並行処理

核言語第 1 版、通称 KL-1 は、並列計算を基本メカニズムとする論理型言語である。並列アルゴリズム、並列プログラム言語、(超)並列アーキテクチャーと、並列処理は、計算機科学を研究する者にとって、永年の課題であった。

様々の試みにもかかわらず、現実にとらえ得たものは、科学技術計算用スーパーコンピュータや画像処理プロセッサや FFT など、得られた計算に対し、その特殊性に着目したもの、あるいはオペレーティングシステムのプロセスの概念や多重プロセッサシステム

表 4 並列型推論マシン

プログラム言語 (核言語)	KL-1 (concurrent Prolog, pure Prolog)
アーキテクチャ	PIM (並列型推論マシン)

や分散処理システムなど、一般的とはいえ、逐次計算の固まりが複数あるというほどのものが得られただけであった。

人間の頭脳が、何らかの形の並列計算に根ざしていると予想されるだけに、計算の本質にかかわる一般的な並列計算メカニズムの実現は、多くの研究者を虜にする十分な魅力を持ち、いくつもの無鉄砲な試みが行われた。結果は無残であった。試みの理由付けとして処理速度の向上、そして時には商用計算機への対抗を標榜しただけに、素子・ハードウェア技術の進歩を利用し、パイプライン制御など、こまごまとした工夫を積み重ね、強引に成長しつづける商用計算機の側からは嘲笑をもって迎えられた。

努力は報われるものである。並列処理への道筋が、ひとつの明瞭な方向付けが、試行錯誤の果てに見えはじめた。関数型言語や論理型言語などの新しいプログラム表現の定式化と、それらに対する並列計算モデルの研究、VLSI の特性を活用するアーキテクチャの研究やハードウェアをふんだんに利用できる環境、これらの上に次のような性質を持った新しいマシンを想定することができるようになった。

- ① 新しい計算表現：多数の、実効的には無限の微小なプロセスの上に計算を組立て、透明度の高いアルゴリズムを実現する。例えば、状態概念を副作用なしに実現するなどである。
- ② 高速処理：(従来の研究がこれのみに注目してきたのは改められるべきである)。
- ③ 高信頼性：故障箇所を切り離し、再構成していくメカニズムを実現する。
- ④ 新しいハードウェア構成法：数種類の基本ブロック(チップ)を組合せることにより、任意の処理速度・容量を自由に得られるような構成法を実現する。

現在、論理型言語の上に並列処理記述機能をもたせた、非常にすぐれた言語である Concurrent Prolog や Pure Prolog を対象に、PIM (Parallel Inference Machine) の名のもとに、いくつかの代表的な方式を選定し、シミュレーションによる実験が行われている(表4)。

3.3 知識表現言語と知識ベース

知識は、知識表現言語で記述され、知識ベースに蓄えられる。知識ベースは、様々な問いかけに、蓄えられた知識を利用し答え、知識の追加や削除にともなう整合性の管理や、さらには知識の獲得のメカニズムも持つようになる。これらのメカニズム自身も知識に関する知識として表現され、知識ベースに蓄えられる。

知識表現の研究は、知識の if-then ルール形式に注目したプロダクションシステムと、知識のモジュール化に注目したフレームシステムの大きく2つの流れがある。プログラム表現の世界にもプログラムを if-then ルールで表現する論理型言語やその特殊型としての関係データに注目する関係データベースシステム、そして新しいモジュール化の方向を示すオブジェクト指向言語がある。

これらの動きが結び合わさるところに、新しい知識表現言語と知識ベースシステムを描くことができる。しかし、この分野の研究の仕方として、最も一般的な知識のあるべき姿を追求するという設定には、充分な注意が必要である。知識は対象分野によって、またどのレベルをどう利用するかによって、表現の仕方にバラエティーが生ずる。このバラエティーを適格に体験することを通じて、少しずつ一般性へと接近できる。したがって、適切な応用例を数多く手掛けることが必須である。しかし、いずれにしろ出発点としての一般性の高い知識表現の枠組が必要である。

そのひとつが知識プログラミング言語という考え方である。これは、知識表現としての構成要素は基本的なものにとどめ、プログラムとしての記述力を付加することによって、表現の一般性と処理能力の高さを求める方法である。もうひとつが、特定の形の知識に着目するという考え方である。一般的ではあるが、ある形の知識を非常に大量に扱う場合で、大量のファクト(事実:関係データと同義)や極く単純なルール(規則)を高速処理するという観点からの関係データベースの高度化という進め方である。実際の CAD 等では、多量のファクトの効率良い扱いが大切な技術となる。

知識プログラミング言語に関しては、MANDALA

表 5 知識表現と知識ベース

基本機能	KAISER (知識ベース・管理システム)
プログラム言語 (核言語)	関係代数, KL-0
アーキテクチャ	DELTA (+SIM) (関係データベース・マシン)
基礎システム	MANDALA
基本機能	(知識プログラミング・システム)

というシステムが提案され、基本仕様の決定、実験システムの試作が行われ、記述実験が行われている（表5）。関係データベースに関しては、メモリー階層を含めた総合的なシステムとして、DELTA という名のもとに第1版のハードウェア部が完成した。並行して、そのソフトウェア部となる高機能な管理システム KAISER (Knowledge Acquisition Oriented Information SupplIEr) を開発中である（表5）。

3.4 自然言語と会話

自然言語（日本語、英語）は、人間が知識表現のために用いる最も一般的な言語であり、知能解明への最大の手掛りである。この自然言語理解のメカニズムを構文、意味、文脈にわたって総合的な観点から解明し、システムとして具体化する。理解のメカニズムは、会話のメカニズムをその本質的部分に持ち、会話のメカニズムは、第5世代コンピューターシステムの基本メカニズムのひとつである。自然言語処理は、最終成果としては、多くはインタフェース部に位置付けられるが、プロジェクト全体としては、知識情報処理システムの中核に深くかかわる重要な役割を果たすものである。

自然言語の良さは、その高い自由度である。この自由度が、人間の思考に不必要な束縛を強いることなく、適切なガイドラインを与えてくれる。したがって、自然言語を扱うシステムは、この高い自由度を十分に生かし、多様さを多様さとして扱うメカニズムを持っていなければならない。言語現象すべてとはいえないが、ある範囲、あるレベルに限定すれば、この自由度を許容しうるだけのコンピューターの諸技術が整備されつつある。

自然言語処理を本格的な技術に育て上げるためには、言語学等からのしっかりとした理論的な枠組を得ること、辞書や例文集等の言語データを整備すること、様々な実験システムを試作する実験ツール・環境を用意することなどが必要である。理論的な枠組として、Gazdar の GPSG や Bresnan & Kaplan の LFG などの文法理論、Montague のモンテギュー文法や Barwise の状況意味論などの意味理論等、よりアルゴリズム的な精密な理論が用意されてきている。

言語データに関しては、国立国語研究所による整備や出版社や新聞社等で、それぞれの業務のコンピューター化に伴うデータの蓄積が行われている。しかし、著作権上の制約が非常に強く、また、必ずしも利用に適切な形態になっていない。広く公開できる言語データの整備を急がねばならない。

実験ツール・環境に関しては、先に紹介した SIM が

表 6 自然言語

インタフェース	BUP (文法・意味記述システム)
基礎システム	
基本機能	電子化辞書・構造化辞書

十分なワードプロセッサ的な日本語処理機能を持ち、様々なユーティリティーを備えるものとして設計されている。

意味記述に素直に結びつく文法記述のシステムとして、BUP (Bottom-Up Parser) が開発され、辞書や例文集等、基礎言語データの整備と並行して、意味辞書の実験が行われている（表6）。具体的な応用システムとしてはいろいろあるが、やはり代表的となるのは機械翻訳であろう。このテーマに関しては、基礎的な準備を行っているが、まだ陽に明示する段階でないと考えている。

3.5 論理型プログラム言語と知的プログラミング

論理型プログラム言語を土台に、ソフトウェアの作成・保守・改良の全過程における生産性の本格的向上を達成すべく、プログラムの作成過程、プログラムの利用・保守の過程を秩序立った枠組の上にとらえ、適切に支援するシステムを実現する。プログラミングの本質的メカニズムの解明に取組み、新しい理論的土台の整備を行いつつシステムへと具体化していくアプローチと、まずは利用可能なツールとしてのプログラミングシステムを実現し、活用しつつ高機能化していくアプローチとがとられる。

本質的メカニズムに取組むアプローチでは、プログラム・仕様記述言語、プログラム、プログラミング、開発プロジェクト等々の論理を定式化し、それに基づくプログラムの変換・検証・合成のメカニズムを具体化するという手順で進められる。この中でも、仕様記述と検証が要である。

仕様記述言語としては、自然言語に基づくもの、形式的仕様記述に基づくもの、プログラム言語そのものに基づくものが考えられるが、まず自然言語に基づくものに大きな比重を置き進めつつある。

検証に関しても、いくつかのレベルを設定しなければならないが、次の分野で述べる証明支援システムの成果が大きく貢献してくれるものと期待している。

システムの目標像は、Scott 等の提唱する推論的プログラミングシステム (Inferential Programming System) である。具体的ツールからのアプローチでは、先に述べた SIM が、プログラム開発用ワークステーションとして設計されており、この SIM のプロ

表 7 論理型言語と知的プログラミング

基礎システム	仕様記述システム
基本機能	証明システム

プログラミングシステムの高機能化，計算機ネットワークを介しての共同作業環境の整備という手順で進められる。

現在，自然言語（日本語，英語）に基づく仕様記述言語とその処理システムの試作と論理型言語への Boyer-Moore 型の定理証明システムの試作を行っている（表 7）。また，階層型モジュラー化機能を持った言語の開発と，それをを用いてのシステムソフトウェアの開発，高機能なプログラミングシステムの開発が SIM 上で行われている（表 2）。

3.6 数学言語と証明

数学の典型的な分野を対象に，数学的知識に対する推論・問題解決のメカニズムを解明し，証明作業を支援するシステムを実現する。定理証明や数式処理の成果の上に，さらに理論的成果を積み上げ，帰納的推論等の高次推論のメカニズムの具体化を図る。上述の知的プログラミングシステムの研究・開発と一体化して進められるもので，数学者等多くの理論家の協力を必要とする分野である。

現在，対象分野として，線形代数，記号演算，総合微分幾何を選び出し，理論的な研究や，第 1 版の試作

をスタートしたところである。

4. む す び

第 5 世代コンピュータープロジェクトの現状と展開を，その中心部に重点を置いて紹介した。人間の知的機能は決して簡単なプログラムによって実現されることはない。相当大規模なプログラムとデータが用意され，はじめて有用な知的機能を具体化することができる。現在のコンピューターは，通常の応用に対しても，ソフトウェア危機といわれかねないのが現状である。そこで，大きく土台を移し，すなわち基本となるソフトウェアシステム，プログラム言語，アーキテクチャーを一新し，この大規模プログラムの負担に耐えられるようにする。そして併せて，現在のコンピューターのもつソフトウェアの問題に根本的な解決をもたらす。これが第 5 世代コンピューターの技術面での基本精神である。

なお参考文献は，大部にわたるためほとんど割愛した。現在，約 40 のテクニカルレポートと，やはり約 40 のテクニカルメモが出版され，研究・開発の成果を広く世に問うている。

（昭和 59 年 5 月 29 日受付）

〔参 考 文 献〕

- 1) 廣瀬健，瀧一博：“第 5 世代コンピュータの文化（モノド・ブックス 26）”，海鳴社，p. 72（1984）
- 2) 瀧一博，廣瀬健：“第 5 世代コンピュータの計画（モノド・ブックス 27）”，海鳴社，p. 86（1984）