第五世代コンピュータ前期計画の成果

1. まえがき

第五世代コンピュータプロジェクト，正式には，電子計算機基礎技術開発プロジェクトは，3年間にわた3前期計画を本年3月に終了して，現在，中期計画初年度の段階にある。これまで欧米追従型に終始してきたわが国のコンピュータ研究開発のパターンを脱して，進路を自力で開拓しなければならないこの行程は，従来の尺度でみれば必ずしもすべてが順調であったということはできないかもしれない。一研究者も告白している1)，‘やっと解く手がかりを得られるかどうか，というような難問に突き当たり，それを避けて別の道から一応の目標に到達した研究も少なくありません。’というのも事実であれば，昨年11月の第2回第五世代コンピュータ国際会議に出席した海外の研究者が評したように，‘ICOTの成果には，まだ本当のブレークスルーといえるようなものは見当らない。’というのも，ブレークスルーとい

1) 近山隆，聴べ！第5世代コンピュータ，ぱそこん5増刊(1982)，pp.180～185.

うことばの意味を主要な難問題に対する決定的な解決とすれば，それはまさに指摘のとおりかもしれない。

しかし，これらのことから直ちに前期計画は十分な成果をあげられないままに終ったとするならば，それは大きな誤りである。第五世代プロジェクトの開始に先立って作成された研究開発計画書[2]に明記されているとおり，「前期の研究開発の重点は，知識情報処理の分野におけるこれまでの研究成果を収集し，評価と再構成を行うとともに，各（研究開発）課題におけるいくつかの候補を絞り，中期に向けての基本（要素）技術開発を行う」ことにある。これを登山にたとえていえば，前期計画の目標はベースキャンプの設営と登はんルートのめどをつけることであるといえる。したがって，この段階では予見される アイスフォールやオーバハングの対策を細かく詰めるより，まず未知の山である ロジック・プログラミングの性質を十分に知ることの方が大切であると考えられる。

これまで理論的な可能性として一部の専門家にしか認

2）日本情報処理開発協会，第5世代のコンピューター研究開発計画（1982.3）P.75

めれていなかったロジック・プログラミングを多面的に展開して具体化し，その広汎な応用可能性を実証できたことは，上述のような前期計画の目標にかなう最大の成果であるということができる。以下では，このような成果の内容を，主要な研究開発課題の分類に従って解説していくことにする。

2. 前期計画とその研究開発課題

第五世代コンピュータプロジェクトは，未知要素が多く，試行錯誤が必要な研究開発であるため，10年間という例外的な長期間が予定され，前期3年，中期4年，後期3年の3段階アプローチをとることになっている（第1図参照）。

前期の目標は，すでに述べたように，第五世代コンピュータを構築するために必要な基本要素技術を確立することである。すなわち，知識情報処理指向の新しい原理にもとづくコンピュータシステムを具体化するために必要な基本技術を，機能別モジュールとしてのシステム要素ごとに開発しようとするものである。このようにモジ

ュール化のアプローチをとる理由は，「分割して征服せよ」という通常の考え方に従うだけではなく，それぞれの要素に独自性を許すことによって，より深い可能性の追及を可能にする狙いがある。

第1図にみられるように，前期の研究開発課題は予想される第5世代コンピュータの構成に合わせて4つのグループにまとめられている。ハードウェアシステムはコンピュータ本体に相当する推論サブシステムと，データベースマシンに相当する知識ベースサブシステムという2つのグループからなり，それぞれ5つずつの課題を含んでいる。ソフトウェアシステムは本来基礎ソフトウェアと応用ソフトウェアで構成されるはずのものであるが，基礎的な部分が未確定な段階で応用システムの本格的検討を行うのは無理と考え，その予備的検討を基礎ソフトウェアの一部とするに止めている。基礎ソフトウェアシステムは，大きく4つの研究開発課題に分けられている。最後のグループ，ソフトウェア開発用パイロットモデルは，中期以降の研

究開発用ツールとしての高性能パーソナルマシンを自己開発しようとするものである。このマシンの役割は，論理型言語にもとづくソフトウェア開発のためのワークステーションとしてのものである に止まらず，これを出発点にしてブートストラッピング的に本格的な開発支援システムへと発展させる狙いをもつものである．

3. 基礎ソフトウェアシステムの成果

基礎ソフトウェアシステムの研究開発課題は，第1図に示されている 4つの要素技術の研究開発と，これらすべてのモジュールのためのプログラム記述の土台となる新言語，すなわち（5G）核言語の研究開発である。第5世代コンピュータの目指す知識情報処理は，現在の手とり足とり式の具体的な操作の系列としてのプログラムでは実現できない高度な情報処理であって，考える力をもった人間のするように利用できる知識を動員して，与えられた課題に関して何をどうすべきかを自分で導き出して実行できるような，いわば推論能力をもつプログラムによって可能になるものである。このような知識ベース

機能と問題解決・推論機能を述語論理の枠組を利用して具体化しようとするのが，プロジェクト前期計画での基本的な考え方である。したがって，5つの課題に対する研究開発は，いずれも，プログラミング言語における Prolog，知識表現における論理形式など既成の論理的手法を出発点とし，それらに高度並列処理，対象指向プログラミングなど次世代プロジェクト独自の選択による機能強化を加えて新しい可能性を引き出す，というかたちで展開された。次2図は，各課題の相互関係と内容のあらましを示したものである。

(1) 核言語はもっとも基本的なシステム記述言語であるとともに，機械語を規定するものとしての役割をもつものである。核言語には逐次型実行を前提にしたKL0の系列と，並列型実行を前提にしたKL1の系列がある。KL0系列は，大型機用に開発されたDEC－10 Prologの言語仕様を，ワークステーションであるソフトウェア開発用パイロットモデルのマシン機能に合わせて合理化したものということができる。すなわち，パイロットモデルの機械

語としてのKL0は，実行効率化，並進実行，ストリング操作など利用頻度が高く，しかも比較的容易に実現できる新機能を追加する一方，プログラム定義データベース管理などのように頻度が少なく，マシン負担の大きい高級機能を削除した仕様になっている．そして，これらの削除をカバーするとともに，モジュール化機能（オブジェクト指向型）とマクロ展開機能を導入してシステム記述力を高めたユーザ言語ESPを別に開発した。この分離によって，効率と記述力という矛盾する要求が高い水準で調和できることになる。ESPの有効性は，後出のSIMPOSの開発を通じて十分確認されたものと考えられる。

KL1系の出発点もPrologであるが，それとの本質的な違いはKL1が並列性を陽に記述する並列プログラミング言語である点である。すなわち第5世代コンピュータは高度並列処理の実現を目指すものであるので，そのプログラミング言語は逐次実行へのコミットが最小であるようなものでなければならない。そのためには，実行単位が細分さ

れていて互の独立性が高く，しかも必要な相互間の通信と同期化の制御が精密に行えるものでなければならない。しかし，そうした性格をもつプログラミング言語にはほとんど先例となるものがなく，開発に当ってはある程度の試行錯誤が必要になる。最初のKL1として開発されたConcurrent Prologは，論理型並列プログラミング言語として世界最初のものであり，AND並列機能，OR並列機能，Set/Stream変換機能，モジュール化機能，メタ推論機能などすぐれた言語仕様をもっている。ところが，この言語に対する処理系の試作を行って検討した結果，OR並列機能のための多環境やread-only anotationの実現が複雑になり，このままでは効率と並列実行に適しないことが明らかになった。そこで，言語仕様に意味論的な規約を加えることによってこれらを不要なものとして生まれたのが，第2代目KL1のGHC（Guarded Horn Logic）である。GHCとConcurrent Prologは言語仕様の大部分が共通であるため，ハードウェア設計など前期の間にConcurrent Prologについて得られた知見は，そのまま

GHCに引継ぐことができる。

(2) 向題解決・推論ソフトモジュールの目的は，帰納的推論，あいまいさを含んだ推論などといった高次の推論機能を実現することであるが，最初からこのレベルを目指すことは無理であるので，前期では専ら演繹的推論プロセスの実行を高性能化する方式を追求することになっている。推論プロセスを高速で実行する方法としては，並列処理の導入，実行制御の精密化，向題の特殊性活用などが考えられる。そこで，これらに対応させて3つの要素技術，すなわち，並列推論機能，メタ推論機能，向題向推論エンジンを抽出し，それらの具体化検討を行った。並列推論機能については，完全な並列性をもつ，いわゆる純粋Prologプログラムに対する並列実行インタプリタをいくつかの方式について実現する，という形での検討を行った。メタ推論機能は，推論そのものについて推論したり，推論のしかたを制御したりするものである。逐次型実行環境下でのメタ推論機能としてはdemo述語というのがよく知られているが，本プロジェクトでは並列実

行環境下でこれと同じ機能を実行するsimulateと呼ばれるシステム述語を開発した。simulate述語の動作内容はその定義の与え方によって変えることができるので，これを使えばユーザが様々な推論法を自由にプログラムに組込むことができるようになる。さらに，こうしたメタ推論が組込まれたプログラムの実行を効率化する工夫として，部分計算法の利用方式を開拓するとともに，数式処理への応用を通してメタ推論の有効性を実証的に検討した。また，問題向推論エンジンは，実用規模の具体的問題を扱うための手法を検討するものであり，分散型問題解決，問題に即した問題解決の具体化を行った。内容的には，将棋の終盤における人間の問題解決活動をモデルにして，コンピュータ上での分散型問題解決の大局的枠組として知識アーキテクチャの構想をまとめ，さらにこれと並行して，実際的問題の代表例として電子回路のためのCADを選び，従来のCAD技術では扱えなかった高品質設計を実現するための方式について試作的検討を行

った。

(3) 知識ベース管理ソフトモジュールは，知識蓄積機能，分散知識源活用機能，知識獲得機能などを完備した知識ベースの実現を究極の目標とするものであるが，前期においてはその要素技術開発を目的にしてつぎの4つの課題について研究を行った。

1. 大規模知識ベースの実現技術
2. 論理にもとづく知識獲得の定式化
3. 知識表現システムの設計
4. 実験的エキスパートシステムの開発

述語論理記述を知識表現の基礎にとり，論理表現と親和性の高い関係データベース技術を用いて，大規模知識ベースの枠組となる実験的データベース管理システム[ソメ]KAISER (Knowledge Acquisition-oriented Information SupplieR) を開発した。このシステムは逐次型推論マシンPSIと大規模関係データベースマシンDeltaによってサポートされるもので，PSI上の内部データベースならびにDelta上の外部データベースを管理する。全体は5

つのモジュール，知識獲得，知識操作，知識蓄積，知識インタフェース，知識対話の各モジュールで構成されている。これらのうち，知識操作と知識蓄積のモジュールは従来の関係データベース管理プログラムに相当するものであり，知識インタフェースモジュールは論理型言語による検索要求を関係データベース操作用言語に翻訳・実行するものである。知識対話モジュールは，利用者との自然な対話を通して必要な情報を提供することを目指すもので，現在は半自然言語文が扱える段階にある。知識獲得モジュールは，知識ベースが新しい知識を獲得したり，それに合わせて内容を更新したりするもので，その基礎になっている考えかが次に述べる『論理にもとづく知識獲得の定式化』である。

有名なピアジェの発生的認識論に従えば，知識獲得の過程には知識同化，知識調整，知識均衡化の3つのプロセスが含まれると考えられる。そこで，知識ベースを論理命題からなる論理的体系とみて，ピアジェのいう同化，

調節などを論理的整合性の文脈で解釈して上記のプロセスを実現しようというのが，ここでの知識獲得の定式化の意味である。このような知識獲得機能の実現にとって本質的なのは，一般の知識の使い方に関する知識，すなわちメタ知識とそれを管理するメタ推論機構である。これらのメタ知識とメタ推論機構を用いて，外から得られた知識のうち，無矛盾に追加できるものだけを知識ベースに取込む知識同化機能，逆に新しい知識を優先し，それに適合するように知識ベース内の知識を修正していく知識調節機能などが実現されている。このような一般の知識の同化や調節に比べて，メタ知識の同化や調節の実現は難しい。知識ベースの中には確実な知識ばかりではなく，あいまいさを含んだ知識（仮定的知識，信念など）も存在する。このとき，メタ知識の追加に対してこれらの知識の間に矛盾が生じないように，あいまいな知識に修正を加える機能（Truth Maintenance 機能）を備える必要がある。KAISERには，こうした機能もある程度用意されている。

知識表現のあり方は，知識情報処理における もっとも重要な問題の一つであり，しかも もっとも見通しの立てにくいものである。人間の扱う知識の形式は多種多様であり，それを単一の表現形式でサポートするのは無理である。それで本プロジェクトでは知識表現システムとして固定的なものを用意しておく代わり，必要なものが自由に定義して使えるような知識プログラミング言語を具えておくことにした。このような言語としては Mandala と CIL の2種類が開発されている。

Mandala は，論理型言語（KL1）をベースにオブジェクト指向，ルール指向，データ指向などのプログラミング技法のための枠組を統合的に盛込んだものということができる。論理型言語を基礎にして考えると，プログラムや知識ベースは命題のかたちをした知識の集まりとみることができる。この知識の集まりをブロック化し，関係のあるものをその関係に従って向連づけ，知識の組織的構造を明示化するための枠組を与えるのが Mandala の

役割である．ブロック化の単位としては，実体（オブジェクト）と世界があり，プログラムの場合でいえば，それぞれ，プロセスとプログラムという概念に相当している．また，ブロック間の関係を表現するものとしては，instance-of，is-a，part-of，manager-ofなどがある．Mandalaに対しては，現在までのところインタプリタとプログラミング支援ツールとしての知識ベースエディタが用意されている。

CIL（Complex Indeterminate based Language）は，もともと問題向知識表現言語として発想されたもので，対話の内容が理解されていく過程を計算機でモデル化するために用いることを意図している。しかし，得られた結果は一般性の高い表現形式になっており，今後広い応用がみられるものと期待されている。この言語の特徴を要約すると，Prologにコンストレイント付きのフレームを強化したものということができる。これは自然言語のための最新の意味理論である状況意味論における3タイプ（個別的な部分を不定項化した残りの共通部分を表わ

すもの）の概念に見合うもので，Prologの記述力を大いに高めるものということができる。

実験的エキスパートシステムを前期で開発する目的は，知識情報処理のためにどんな機能が要求されるかを実証的，体験的に明らかにすることである。このため，日本語校正支援システムと論理設計支援システムの2つについて試作レベルの検討を行い，また，医療費控除相談支援システムとオフィス業務支援システムの2つについて論理設計レベルまでの検討を行った。これらの実用システム開発を通して，論理型言語，対象指向言語，並列型言語の実際的側面について理解を深め，本プロジェクトのアプローチについて確信を得ることができた。

(4) 知的インタフェースソフトモジュールは，人間と柔軟な会話ができる機能をコンピュータにもうることを目的にするもので，現在の機械翻訳システムなどよりは3かに水準の高い言語能力をもつシステムの実現が課題である。このため，従来からの自然言語処理技術を基礎から見直す必要であり，前期においては近年進歩の著しい言語学

の成果を最大限に吸収・活用することに研究開発の主眼点をおいた。また，短期間で内容を深めるため，対象とする言語を当面日本語と英語に限ることにした。

自然言語処理のための道具だてとしては，まず文法と辞書が必要である。文法の枠組としては，最新の文法理論の中からコンピュータでの扱い易さを考えて，LFGとGPSGを候補として選んでいる。これらにもとづく英文法はすでに世の中に相当なものが存在しているが，日本文法については本格的なものは自主開発する必要があり，目下その努力を続けている最中である。また，辞書については，常識と高い知能をもつ人間用のものとは内容・作り方とも本質的に異なるため基礎から考え直す必要がある。このため，前期では本格的な電子化辞書の基本仕様を固めるための小規模な実験試作を中心にした。そして，これらの検討を背景にして，構文解析，意味解析，語用解析，談話理解モデル，文生成機能などの研究開発を行った。

構文解析については，電子技術総合研究所などの指導の下に，論理型言語の特徴を活用したボトム・アップ型の高性能構文解析プログラムである BUP システムを開発した。このシステムは拡張文脈自由文法に属する文法に適用可能で，そのために各文法を定義するのに使う文法記述システムも用意されている。

自然言語の意味を扱うための枠組にはこれまで十分なものがなく，そのため本格的な意味処理を考えることは無謀な試みとみられてきた。しかし，最近の状況意味論の出現によって，その見通しはかなり明かるくなりつつある。状況意味論は目下発展中の理論で日々に新にの状態にあるが，その中核部は意外に安定しており，また，理論の枠組は包括的で単に意味論だけでなく語用論の範囲もカバーできるものである。そこで，これを基礎にとって，そのうえに談話理解，すなわち，文章を読みその内容を理解し質問に応じられるシステムの実験モデルを開発することが考えられる。前期に試作した DUALS システムはこの考え方に従ったもので，LFG による構文解析，

状況意味論にもとづく意味解析と文脈解析の他，問題解決や文生成の機能までをもつものである。ただし，現在の実力は，小学校3年生の国語教科書の範囲内に止っている。因に，知識表現のところで述べたCIL言語は，本システムのために開発されたものである。

(5) 知的プログラミングソフトモジュールは，プログラムの作成をはじめとして，ソフトウェア開発・保守の基本部分を自動化するとともに，その開発から保守にいたる全過程を統合的に管理・支援するシステムの実現を目指すものであるが，前期ではその前駆的段階として，対象指向論理型言語ESPをベースにしたプログラミング・システムの開発と，プログラミング自動化のためのもっとも基礎的な技術であるプログラム検証を中心にしたソフトウェア管理プログラムの研究開発を行った。

ここでのプログラミングシステム開発の目的は，大規模ソフトウェア開発の強力な手法であるモジュラープログラミングを本格的なかたちで実現でき3プログラミン

グ環境を，論理型プログラミングのために，論理型プログラミングによって作成することである。そしてプログラムモジュール化のための中心的機能として用意されたのが，ESPにおける「対象（オブジェクト）」の概念である。このプログラミングシステムは，ソフトウェア開発用高性能パーソナルマシンPSIのためのソフトウェアシステムSIMPOSの一部として実装され，現在実用に供されている。内容としては，記述系（ESPとその処理系），最適化系，作成支援・管理系（デバッガ，ライブラリ，ファイル），編集系・対話系（エディタ，コーディネータなど），交信・入出力系（ウィンドウ，ネットワーク）からなる。デバッガなどの各要素は一種のエキスパートとして動作し，コーディネータを通して利用者の意向をうけ，プログラムを作り上げる。ただし，現状でのエキスパートは知的とはいえない。

つぎに，ソフトウェア管理プログラムは，プログラムの設計，コード化，テスト・デバッグ，修正・改良，保守・管理などプログラムライフサイクルの各段階におけ

る知的な支援機能の実現を目指すもので，各段階ごとの高度な専門知識を組込んだ実験的システムを開発するという形で4つの研究を行った。ソフトウェア開発利用コンサルテーション実験システムは，自然言語によるプログラムの仕様ないし内容記述の可能性を実証的に検討するためのもので，モジュラープログラミングの枠組を使い，自然言語的に記述された仕様を満たすモジュールをライブラリから検索し，どこを修正すれば要求通りの仕様になるかをユーザに自然言語的に説明する，といった機能をもっている。つぎに，階層型論理プログラム検証システムは，与えられた論理プログラムが仕様通りの機能をもつかどうかを確認するためのシステムである。一般にプログラムの仕様は，そのプログラムが満足すべき基本条件をまとめたものである。この条件は論理的な命題として述べることができ，論理プログラムがこの命題を満足するというのは，命題が論理プログラムから論理的に導かれることを意味することになる．したがって，

論理プログラムの場合にはその実行メカニズムの中でプログラム検証が実現できそうにみえるが，実際には仕様記述のためにはプログラムとしては許されていない論理式を認める必要があるなどのため，専用の証明システムを工夫しなければならない。そのようにして実現されたのがこのシステムである。ソフトウェア再利用基礎実験システムは，類似したプログラムを重複して開発する無駄をなくすため，既存のプログラムから利用できる部分を抽出して再利用する技術の確立を目指すものである。必要な機能は，プログラム部品を蓄積・管理するためのデータベース機能，類似プログラム部品検索機能，プログラム部品修正機能，プログラム部品管理最適化機能でそれらの大部分はMandala言語によって実装されている。最後に，イメージによるソフトウェア生産システムは，文章，表，図形からなる通常のプログラム仕様書から直接プログラムの自動作成を可能にするため，仕様書，特にそのイメージ部分の意味理解能力の実現を目標とするもので，予備検討の結果実現性が高いことが判明した。

4. 推論ならびに知識ベースサブシステム

第5世代コンピュータハードウェアの中核部は，推論サブシステムと知識ベースサブシステムによって構成される．現在のコンピュータでいえばこれらは本体とデータベースマシンに相当するものであるが，プロジェクトの最終段階ではこれらは互に融合されたものになりそうというのが現段階での見通しである．しかし，当面は分離したものとして扱うのが便利である．

作業仮設としての推論サブシステムのイメージは，推論機能をもつ超高性能マシンとしてよい．ここで，推論機能をもつという意味は，前節にみた核言語プログラムを実行する能力があるということであり，また，超高性能マシンという意味は，要素プロセッサ1,000台規模の高度並列処理システムを想定してのことである．このような高度並列処理の実現方式としては，目下のところデータフロー方式とリダクション方式が有力であり，他には適当な候補が見当らない．これら2方式をどのようにし

て核言語プログラムの実行に結び付けることができるか，それが推論サブシステム前期研究開発の中心課題である。

知識ベースサブシステムは，推論サブシステムでの処理に必要な知識を蓄積し，要求に即応して関係する知識を検索して提供する機能を期待されるものである。従来のデータベースマシンが一定の形式のデータを直接的な場所指定を行うインデックスと与えられて読み書きするものであるのに対し，知識ベースマシンの場合は，様々な表現形式をもつ知識を内容的な関連性や複雑な条件にもとづいて探し出すといった高度な機能が要求される。また，扱うべきデータ量も桁ちがいの規模になるものと見込まれている。このような状況を考えると，前期の段階でいきなり本格的な知識ベースマシンのモデル設定を行うのに無理があるのは明らかである。そこで，データ表現形式が核言語での基本形式に近いこと，知識ベースで重要になる集合演算機能に強いこと，大容量化ならびに高度並列化に対する適性が高いことなどを判断して，前期でのモデルとしては関係データベースをとることに

した。すなわち，並列関係・知識演算を指向した大容量関係データベースマシンの開発が，知識ベースサブシステム前期研究開発の中心課題である．

(1) 推論サブシステム

核言語プログラムの実行と高度並列処理方式を結ぶためには，核言語プログラムの特性を十分把握する必要がある。しかし，先立つべき核言語が未完であるうえ，その代替として使ったProlog言語にも本格的なプログラムの実績が皆無に近い状態であったため，相当な努力を払ったにもかかわらず，一般性のある特性解析結果をうるのは難しかった。この辺に前期研究開発の限界があるといえる。

核言語などのようなProlog系の論理型言語では，つぎのようなホーン節とよばれる形式の論理式が基本的な文

$$A \leftarrow B, \cdots, C$$

となっている。これは，「Bかつ・・・かつCならばAである」という関係を表わしている。Bから Cまでの部分

が空であれば，それはAが無条件で成立すること，すなわち事実であることを示すことになる．それでその形の文をファクトとよび，それに対してもとの条件文の形のものをルールとよんでいる．また，Aの部分が空のものは，'Bかつ・・・かつCは事実か？'という質問と解釈される．このような言語によるプログラムは，ひとつの質問といくつかのファクトとルールの集合として書かれる．そして，このプログラムを実行するというのは，プログラム中のファクトやルールを使って質問の結論を出すための推論をするという意味になる．この推論はつぎのように進行する．最初，B，・・・，Cという列から出発する．そして，ファクトとルールの意味を'左辺の形のものが現われたら，それを右辺のもので置換えよ'というように読みかえていきの列に適用する．もし，'B←'というファクトが適用されれば列からBが除かれることになる．したがって，こうした適用を続けているとうまく行けば列が消えてしまい，まずければ列が消えないままに適用できるファクトやルールがなくなってしまう

ことになる。前者の場合は「Bかつ・・・かつC」が事実である と証明されたことになり，後者の場合はそれが否定されたことになる。ただし，ここで注意しなければならないのは，ルールの中には左辺が同じのものが複数個ある場合があり，それらのどれをとるかで結果がちがってくることである。場合によってはすべての可能性を網羅しなければない。さらに，以上では説明を簡単にするために省略したが，AとかBは述語論理の命題であるため定数や変数を含んでおり，ファクトやルールの適用においてはこのための手当が必要になる。

いずれにしても，核言語プログラム実行過程の大要は以上の通りであり，そこでの並列処理の可能性は，B，・・・，Cなどへのファクトやルールの適用を各項ごとに並列に行うもの（AND並列），同一の項に対して適用できるルールが2つ以上ある場合，それぞれの適用を並行に進めるもの（OR並列），また，適用などの手続の中味を並列化するものなどである。これらの可能性を最大限に

利用して処理の並列度を上げ，しかも，要素プロセッサ間に必要なコミュニケーションのオーバヘッドが少なくてすむように，各要素プロセッサの機能とそれらの間の結合関係のあり方を工夫するのが，アーキテクチャ設計上の中心課題である。

具体的な構成方式としては，核言語の仕様が流動的であるため多様な要求に応える用意が必要なこと，高度並列処理方式は未開拓の分野で未知要素が多いことから，できるだけ多くの可能性を並行に追求するものとしてつぎの4方式をとりあげることにした。

① リダクション方式

いわゆる記号処理向きの方式で，処理対象の記号列の構造そのものをプログラムとみなし，その各部分構造に対してそれから定まる書き換え操作を実行して記号列を変形（リダクション）するという形で処理を行うものである。うえで説明した「B，…，C」の証明過程は，まさにリダクションそのものである。

この方式にもとづくアーキテクチャは，要素プロセッ

サ64台の場合がソフトウェアシミュレータ上に，同じく8台の場合がハードウェアシミュレータ上に，それぞれ実現されている。これらによるシミュレーションの結果は，この実現方式によってプログラムに内在する並列性が十分引き出しうることを示している。

② データフロー方式

プログラムをプロセッサの実行単位（機械語命令）で分解し，実行準備の整ったものは要素プロセッサの都合がつく限り直ちに実行する方式である。したがって，他の命令の結果を使うものは，それが届き次第実行されることになり，全体としてみると処理は結果（データ）のフローに従って進行することになる。原理的にはもっとも高度な並列化が実現できることになるが，結果の授受など情報転送のオーバヘッドが多くなる弱点がある。この方式については，ソフトウェアシミュレータ上のほかに要素プロセッサ8台の実験システムとして実現されている。

③　完全コピー方式

リダクション方式の一種であるが，AND並列を逐次処理すること，並列処理の単位としてゴールフレームというものを設定するがそれには最初の質問を解くのに必要な環境が完全にコピーされていることなどの特徴をもっている。このためゴールフレームは互に独立に実行が進められるので並列度は高くなるが，大量のデータ転送が必要になるという弱点をもつ。この方式についてもシミュレータ上での実現が行われており，つぎに述べる節単位処理方式と共用で要素プロセッサ16台のハードウェアシミュレータが開発されている。

④　節単位処理方式

AND並列を逐次処理する点では完全コピー方式と同様であるが，OR並列の処理に伴う資源要求の爆発を防ぐため，暇になった要素プロセッサが多忙な要素プロセッサに仕事を要求し，それに応じて多忙なプロセッサは処理中の仕事の一部を切り出して要求元の暇なプロセッサに送るというやり方でプロセッサ割付をする方式である。

(2　知識ベースサブシステム

前期計画における関係データベースマシン（RDBMまたはDeltaとよぶ）開発の具体的な目的は，知識ベース機能をサポートするためのメカニズムを研究する実験台を用意すること，別途開発される逐次型推論マシンと協同して完備したソフトウェア開発ツールを構成することの2つである。

このような動作環境におけるデータベースアクセスの典型的なパターンは，ほぼつぎのようなものになると想定される。推論マシンとのユーザはPrologでプログラムを記述し，その中で外部データベースへの問合せを行う。すると推論マシンは，ユーザの問合せ内容とPrologプログラムとからRDBMへの問合せプランを作成し，それにもとづいてRDBMコマンドを生成，適当なインタフェース（含LAN）を介してRDBMに送る。RDBMはコマンドを解析し，検索コマンドについてはデータベースを検索してデータを読出し，インタフェースを介して推論マシンに送る。

このようにして行われるPrologプログラムから関係データベースへのアクセスの特徴をまとめると，

① リレーション内の属性（アトリビュート）に対しては均等的なアクセスが行われる。

② リレーションへの問合せはselection，join，unionなどの関係代数演算の形になり，しかも演算負荷の重いjoinやunionの割合が高い。

③ リレーションを構成する属性の数は，あまり多くないケースが多い。

関係データベースマシンの性能に大きく影響する内部スキーマには，タプル型格納方式とアトリビュート型格納方式がある。従来の関係データベースマシンではタプル型が使われてきたが，上記のような論理型プログラミング環境下ではアトリビュート型が適すると判断されたため，RDBMではその方式を採用している。そして，アトリビュート型の欠点である属性情報の拡散化に対抗するため2段階クラスタリングを採用するなどの工夫を行っている。

その他のRDBMの特徴をまとめるとつぎの通りである。

① 機能分散型マルチプロセッサ構成

実験機としての機能拡張性，処理性能向上のため，5プロセッサによる機能分散構成とした。

② 関係代数型コマンドインタフェース

推論マシンとの論理的コマンドインタフェースとして関係代数型コマンドを採用した。また，データ転送形式としてはタプル型をとっている。

③ 関係代数演算専用エンジン

RDBMの中核部である関係データベースエンジンは，パイプライン2ウェイマージソート方式による12段ソートセルと関係代数演算処理用マージャで構成されている。4台のエンジンが並列動作できる。

④ 階層構造メモリ

半導体メモリ（最大128MB）と可動ヘッドディスク（最大20GB）とからなる階層構造メモリ方式を採り入れている。

⑤　統計情報収集機能

実験機としての役割から，性能情報や各種統計情報の収集機能を備えている．

5. ソフトウェア開発用パイロットモデル

本格的な論理型プログラミングを効率的にサポートできるワークステーションとして，実用に耐えるハードウェア，ソフトウェアをもつものを前期3年間のうちに完成するのがこの研究開発課題の目的である．

ほとんど未経験に近い言語をベースにしたシステムを短期間で確実に実現するために，安全な選択を基調とした開発方針をとり，逐次処理を前提にしたシステム設計を行った．このため，本システムは逐次型推論マシン（SIM）と名付けられている．システムの性能目標は，後期末の水準においても十分本格的なパーソナルマシンとして通用することを目途に，現在の大型機DEC－2060上でDEC－10 Prologプログラムを走らせ場合と同程度以上になるものとした．また，記憶容量も同じ考え方で16MW相当を目指すことにした．また，ソフトウェアシステム

としては，最近のスーパーソナルコンピュータで重要視されるようになってきた高度なプログラミングシステムの機能を，論理プログラミングのために実現するものとした。そして，それをサポートするためのOSは，対象指向の概念に徹した構造をもち，ウィンドウ，ネットワーク，ファイルなどの機能に優れたものにすることにした。さらに，システム記述における論理プログラミングの有効性を実証的に検討することを目的にして，システム全体を前述のESP言語で記述するものとした。

(1) 逐次型推論マシンソフトウェアシステム

逐次型推論マシンSIMのソフトウェアシステムはプログラミングシステムとOSを包含するという意味でSIMPOS（SIM Programming and Operating System）とよばれている。SIMPOS設計の基本方針はつぎの5項である。

① 一様なフレームワークにもとづいたシステム設計

マシンアーキテクチャから言語システム，OS，プログラミングシステムまでを，Prolog系の論理プログラ

ミングの枠組にもとづいて統一的に構成する。

② マルチウィンドウ機能を利用した対話システム

マウスやキーボードと組合されたマルチウィンドウシステムは，知的システムではきわめて有効な対話手段であり今後のワークステーションでは必須である。

③ データベース機能

Prologと関係データベースは互の順応性が高いので，データベース機能を十分に駆使した新しいプログラミングシステムおよびOSを開発する。

④ 日本語処理

コンピュータは，すべての人が自国語で使用できるものであるべきである。わが国で使うものには日本語が欠かせない。

⑤ 開かれた構造をもつシステム

大規模なプログラムの作成には，できる限り既存のソフトウェアモジュールを再利用するのが望ましい。これをサポートする機能を用意する必要がある。

SIMPOSの全体構成は図　のとおりである。システム全

体はハードウェアを囲んで4つの層をなし，内側の3つの層，カーネル，スーパバイザ，入出力がOSを構成し，最上層がプログラミングシステムに相当している。プログラミングシステムは，前述した基礎ソフトウェアシステムの知的プログラミングソフトモジュールの一部として開発されており，そこに概要が述べられている。

OSのうちのカーネル層はハードウェア資源の管理に当たるもので，プロセッサ管理はマルチプロセス環境を実現し，メモリ管理は記憶領域管理とゴミ集めを担当，デバイス管理は入出力装置の制御を行う。つぎに，スーパバイザ層は，対象指向概念のためのオブジェクト，ストリーム，プールなどの管理，プロセス間通信や実行環境のサポートなどプログラム実行のための基本機能を提供する。また，入出力層は正式には入出力メディアサブシステムとよばれるもので，外部世界との情報交換をサポートするものである。

SIMPOSの現時点での規模は，ESP言語で約12万行分と

なっている。

(2) 逐次型推論マシンハードウェアシステム

逐次型推論マシンのハードウェアには，基本ハードウェアシステム（基本部という）と拡張ハードウェアシステム（拡張部という）の2種類がある。基本部はパーソナルユースの逐次型推論マシンPSI（Personal Sequential Inference Machine）とローカルエリアネットワークで構成されるもので，早期に実用化してソフトウェア開発を促進させる狙いをもつものである。拡張部はPSIの後置マシンとなる高速プロセッサCHI（Cooperative High-performance Sequential Inference Machine）と図形画像入出力装置などの専用装置を含むもので，高性能化，高機能化を主眼として比較的長い開発期間をあてることにしたものである。

PSIとCHIは，共にタグアーキテクチャとマイクロプログラム制御方式を採用しているが，構造上の共通化を図るということは特に意識されてはいない。PSIでは入出力制御やOSサポートなどの機能が重視されているが，

CHIは後置マシンとしてそれらの機能を最小限に抑え，論理型言語プログラムの高速実行に最重点をおいた構造になっている。また，PSIが核言語KL0を機械語とする立場であるのに対し，CHIはコンパイラによるプログラム最適化の効果を重視して機械語を低い水準に設定している。

実行速度はPSIの30 KLIPS（Logical Inference Per Second）に対して，CHIは200 KLIPS以上であり，記憶容量は40ビット×16M語：36ビット×64M語，マシンサイクルタイムは200ナノ秒：100ナノ秒などとなっている。素子技術は，PSIがTTLとNMOSを中心としているのに対して，CHIはCMLとCMOSが中心である。

PSIとSIMPOSはすでに実用に供しう3段階に達し，20数台のシステムが各所で第五世代プロジェクトの研究開発に使用されている。

6. 中期計画への展望

第五世代プロジェクトの動機の根底にあるのは，コン

ピュータの歴史において，新しい第二の段階が近づきつつあるという認識である。この認識がひとりよがりのものでなかったことは，その後先進諸国において同種のプロジェクトが次々に発足されているのが何よりの証左である。そこで問題になることは，この第二の段階のために準備すべき新しいコンピュータ技術の枠組としてどんなものが考えられるかということである。

この新しい枠組に対する本プロジェクトの想定は，述論理とよばれる論理系をベースにして，コンピュータのハードウェア，ソフトウェアの体系を再構築しようということである。ここで注意したいことは，再構築ということの内容である。単純な理解でいえば，これはすでに実現されているものを別のやり方でもう一度造り直すということになる。これでは再構築を実行してみても，別のやり方が通用することの証明にはなりこそすれ，ハードウェアやソフトウェアには何の進歩ももたらされない。本当にいいたい再構築の意味は，現在のコンピュータ技術では実現不可能な規模と構造をもったハードウェアと

ソフトウェアの体系をまったく新く造り始めるということである。

このような観点で前期計画の成果をみてみると，それはほぼ上述の単純な理解での再構築を実行したものに相当するといえそうである。したがって，その意義は述語論理をベースにしたアプローチが健全なものであることを実証した点にある。しかし，前期成果の意義はそれだけには止らない。確かにできあがった建築物の規模や構造の点からいえば，それは従来のものを大きく抜け出るものではないが，問題はそれが何に役立ちうるものかということである。それをいまの比喩に即していえば，前期計画の目標は新建築のための足場を組むことであったのであり，それが予定通りに完成した訳である。この足場を使って本当の意味のハードウェア，ソフトウェアの再構築に取組むのが中期計画の内容である。第五世代コンピュータの本当の開発が始まるのは中期からである。

中期計画の研究開発課題の多くは未踏分野への挑戦で

あり，述語論理をベースにした新しい枠組の真価が発揮されるのはこれからである．核言語最終仕様の決定，要素プロセッサ100台程度の並列型推論マシンの実現，本格的な知識ベースマシンモデルの設定，帰納的推論などの高度な推論機能や学習能力などの高度な知識獲得機能の実現，本格的な自然言語意味解釈システムや実験的なプログラム変換・証明・合成システムの開発などといったところが研究開発課題のうちの主なものである．これらの中には，前期の研究開発である程度の見通しが得られているものもあり，また，まったくの白紙に近い新テーマもある．帰納的推論や学習能力などの新テーマに対して，前期で用意された枠組がどこまで有効に作用しうるかは，前期計画の成果を本当に評価する試金石といえるかも知れない．