

(資料 2) 第五世代コンピュータ・プロトタイプ・システムの概要

第五世代コンピュータ・プロトタイプ・システムは、主として以下の要素システムからなる。

- (1) 並列推論マシン
並列論理型言語 KL1 の実行のために最適化したアーキテクチャを持つ、並列計算機システム
- (2) 並列論理型言語 KL1 処理系
並列論理型言語 KL1 を効率的に並列に実行するための、プログラム言語システム
- (3) 基本ソフトウェア
並列推論マシン上での KL1 プログラムの実行を司る、もともと基本的なソフトウェア。以下のふたつのシステムからなる。
 - 並列推論マシン・オペレーティング・システム PIMOS
並列推論マシンのハードウェアを効率的に管理し、並列プログラムの開発に適したソフトウェア開発環境を提供するオペレーティング・システム
 - 並列非正規データベース・システム Kappa-P
並列推論マシン上に知識情報処理に必要なデータベース機能を提供するシステム
- (4) 知識プログラミング・ソフトウェア
知識処理応用ソフトウェアで共通に利用できる基礎的な知識情報処理技術を提供するシステム群。代表的なものに以下の3システムがある。
 - 並列定理証明システム MGTP
モデル生成型の並列自動定理証明システム
 - 並列制約処理言語システム GDCC
制約論理型言語の並列処理システム
 - 知識表現言語システム Quixote
大量で複雑な知識を計算機上で効率良く扱うための知識表現言語システム
- (5) 機能実証ソフトウェア
並列推論マシンに構築した応用ソフトウェア・システム群。KL1 言語処理系や知識プログラミング・ソフトウェアの諸機能を利用して構築されており、これらのシステムの機能を実証するものである。代表的なものとしては以下の5システムが挙げられる。
 - 法的推論システム HELIC-II
新しい事件を対して、判例を用いてさまざま法律適用の可能性を提示するシステム

- 遺伝子情報処理システム
タンパク質配列の類似性解析行なうシステム
- 大規模集積回路設計支援システム
LSI-CAD の工程中、論理シミュレーション、LSIセル配置、およびLSI配線の工程を並列に実行するシステム
- 自然言語処理システム
日本語処理システムの研究開発のベースとなる汎用日本語処理系のツール群を提供するシステム
- プログラム自動合成システム MENDELS ZONE
宣言的記述から KL1 プログラムを自動合成するシステム
- 棋士システム 碁世代
囲碁の対局を行なう棋士システム

以下、それぞれについてやや詳しく述べる。

1 並列推論マシン PIM

並列推論マシン PIM は、FGCS プロトタイプ・システムの基本部分を構成する。設計に当たっては KL1 の並列実行に適したバランスの良いハードウェアと、並列化・分散化オーバヘッドの低い KL1 言語処理系の実現に留意した。この結果、応用プログラムレベルでの工夫の相乗効果により、問題によっては 512 プロセサでほぼ 500 倍近くの速度という高い並列処理効果を得、絶対的な性能としても記号処理システムとして世界最高の速度を得ることができた。

高い総合性能を達成するには、単体プロセサ性能の向上、拡張性の高い階層構造を持つ要素プロセサの結合方式という 2 つの技術が鍵を握っている。KL1 実行に適した高い総合性能を持つアーキテクチャを探るために 5 つのモジュール、PIM/p、PIM/m、PIM/c、PIM/i、PIM/k を開発し、様々な要素技術を検証した (別表参照)。

PIM のプロセサの命令体系には RISC、マイクロプログラム、LIW の 3 種類がある。RISC とは、1 命令で実行する処理の内容を小さく揃え、連続するいくつかの命令を多重処理することで高効率な処理を実現するようなプロセサアーキテクチャのことである。マイクロプログラムとは、高機能な機械語を、さらにその一段下のハードウェアを直接制御するソフトウェアで記述するようなプロセサアーキテクチャのことである。LIW とは、1 つの機械語が複数の命令を含むようなプロセサアーキテクチャのことである。また各 PIM モジュールのプロセサは、KL1 を効率良く実行するための専用命令やハードウェア (デレフェレンス命令、タグアーキテクチャ etc.) を装備しており、その効果も確認できた。

PIM/m を除く各モデルのプロセサ間結合方式は、負荷の集中を避け拡張性を高めるために階層構造を成している。8 台程度の要素プロセサ (PE) が 1 本のバスにつながれており、1 個のメインメモリを共有する。この部分はノードまたはクラスタと呼ばれる。クラスタ内の各 PE はアドレス空間を共有し並列にデータの読み書きを行うので、データの整合性を保ちバストラヒックを抑えるために、スヌープキャッシュ

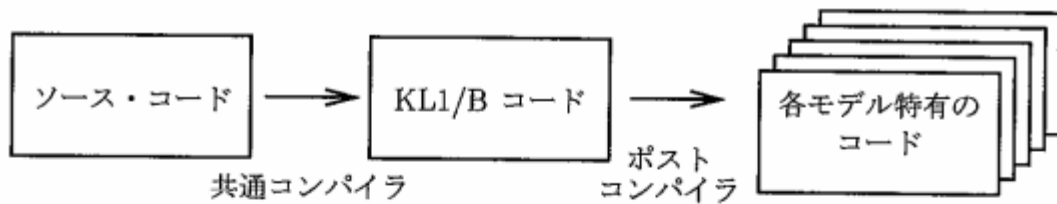
各 PIM モジュールの仕様

	全 PE 数 ¹	ノード構成			疎結合ネットワーク	
		PE 数	メモリ	バス容量 (最大)	トポロジ	バンド幅 ²
PIM/p	512	8	256 MB	130 MB/s	6 次元ハイパーキューブ	33 MB/s × 2
PIM/m	256	1	80 MB	40 MB/s ³	2 次元メッシュ (16 × 16)	8 MB/s
PIM/c	256	8	160 MB	200 MB/s	クロスバ (32 × 32)	40 MB/s
PIM/i	16	8	320 MB	30 MB/s	—	— ⁴
PIM/k	16	4 ⁵	1 GB ⁶	80 MB/s (両階層とも)	2 階層キヤッシュ	—
Multi-PSI	64	1	80 MB	25 MB/s ³	2 次元メッシュ (8 × 8)	10 MB/s

¹PE は要素プロセッサ (Processing Element) の意。 ²ノード・ノード間でリンク双方向合わせた値。 ³メインメモリとキヤッシュ間のスループット。
⁴クラスタ同士は SCSI バスで接続。 ⁵ミニクラスタは 4 PE から成る。 ⁶16 PE で共有されている。

PE アーキテクチャ				
命令セット	プライベートメモリ ⁷	キヤッシュ方式 (状態数 ⁸)	キヤッシュサイズ	
PIM/p	RISC + マクロ呼出	IIM (8 K 語, 1 語 = 6 バイト), LM (2MB, キヤッシュ経由でアクセス)	スヌープ, 書込無効型 (4)	64 KB
PIM/m	マイクロプログラム	WCS (32K 語, 1 語 = 64 ビット)	書戻型 (3)	5 KB (i), 20 KB (d) ⁹
PIM/c	マイクロプログラム	レジスタファイル (256W), WCS (32K 語, 1 語 = 104 ビット)	スヌープ, 書込無効型 (5)	80 KB
PIM/i	長命令形式 (LIW)	LM: 160K バイト (i), 80K バイト (d) ⁹	スヌープ, 書込更新型 (6)	160 KB (i), 160 KB (d) ⁹
PIM/k	RISC .	LM: 128K バイト (i), 128K バイト (d) ⁹	2 階層スヌープ, 書込無効型 (4)	1 階層目: 128 KB (i), 256 KB (d) ⁹ 2 階層目: 1 MB (i), 4 MB (d) ⁹
Multi-PSI	マイクロプログラム	WCS (16K 語, 1 語 = 53 ビット)	書戻型 (3)	20 KB

⁷WCS はマイクロプログラムメモリ, IIM は内部命令メモリ, LM はローカルメモリの意。
⁸ロック状態を除く。 ⁹(i) は命令, (d) はデータの意



KL1 言語処理系のコンパイル方式

という一種のキャッシュが装備されている。さらにこれらクラスタはパケット交換を行うネットワークにより疎結合されている。ネットワークのトポロジにはメッシュ (N 次元格子)、ハイパーキューブ (N 次元超立方体)、クロスバ (完全結合) などがある。ネットワークは、そのデータ転送性能とハードウェア量の兼ね合いを考慮して、KL1 実行に適したものが選択された。PIM/k は疎結合ネットワークを使わずに、キャッシュとバスを階層的に配置して全プロセサを接続するようなアーキテクチャを採用している。

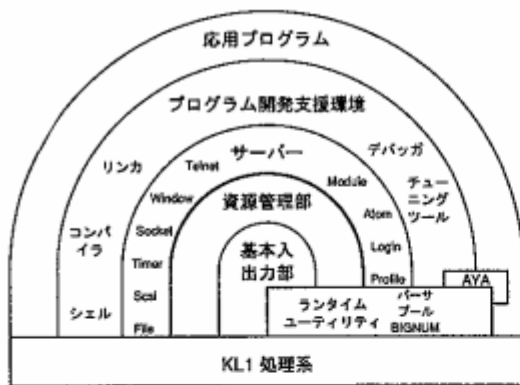
PIM/p と PIM/m の総合性能を概算する。PIM/p、PIM/m の要素プロセサ 1 台は 1 秒当たり 300 ~ 600 K 回推論を行う能力を持っている。PIM/p 最大構成 (512 台) で約 150 MLIPS (1 秒間に 1 億 5000 万回の推論を行なう速度) の性能を持ち、これは従来の汎用機に換算すると約 6 GIPS に相当する。

2 並列論理型言語 KL1 処理系

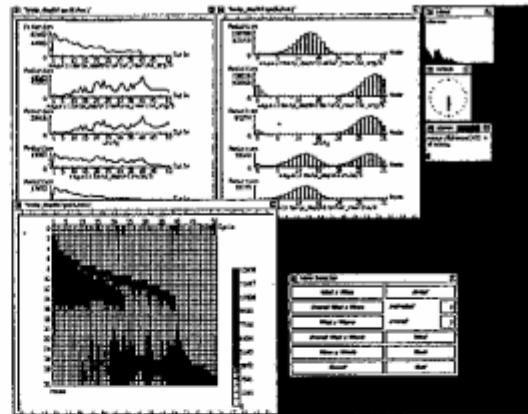
並列論理型言語 KL1 を効率的に並列に実行するための、プログラム言語システムである。同一の言語仕様と共通する処理方式を用いながら、各モデルの PIM に適合した最適化を施している。

特徴的な処理は単一プロセサ内の逐次実行に関する処理と、プロセサ間に渡る処理がある。プロセサ内の処理については、KL1 プログラムをいったん抽象機械語 KL1/B に最適化コンパイルし、これを各モデルのプロセサに適合した方式 (機械語へのコンパイルや、マイクロコードによる解釈実行) で実行する方法を取っている。これによって言語仕様の共通化を容易にすると同時に、処理系構築の手間を削減している。

並列処理は共有メモリ・クラスタ内の処理と、クラスタ間に渡る処理に大別される。クラスタ内処理においては、メモリ・アクセスの競合を調停しつつ、自動的な負荷分散を施し、プログラマが細かいレベルでの負荷分散を指定する必要を除去している。一方クラスタ間処理においては、プロセスの分散はプログラム中の指定に従うが、データや実行コードの分配を自動的に行ない、不要になった領域の再利用も自動的に行なう機構を持っている。これらの並列処理機構についても、アルゴリズムはもちろん、具体的なコードも各モデルに共通するシステムを用意し、モデルごとに最適な方法でコンパイルなどして実装している。



PIMOS の全体構成



PIMOS のプロファイラの表示例

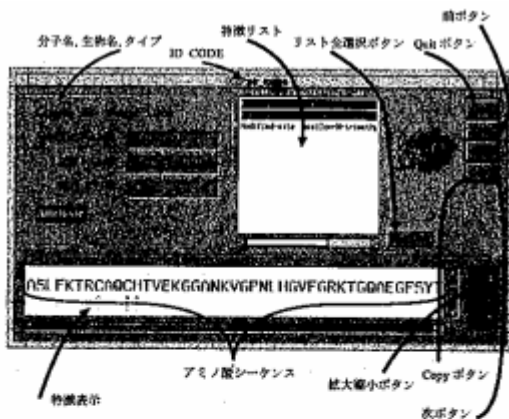
3 並列推論マシン・オペレーティング・システム PIMOS

並列推論マシンのハードウェアを効率的に管理し、並列プログラムの開発に適したソフトウェア開発環境を提供するオペレーティング・システム。どのモデルの PIM でも同一仕様の PIMOS が動作し、同一のプログラム・インタフェース、同一のソフトウェア開発/実行環境を提供する。

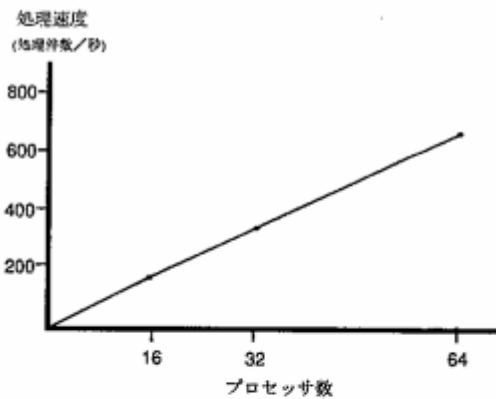
並列処理システム用のオペレーティング・システムは既に数多く開発されているが、大部分のシステムは逐次型計算機用のシステムを若干手直ししたものであり、プロセッサ数十台程度までの小規模システム用か、画一的な処理に向けた SIMD 方式や数値計算などの特殊用途システムでしかなかった。この点 PIMOS は、超高並列システムまでを見通した汎用並列処理システムとして世界で初めて本格的に実装され、実際の研究に役立てられるシステムとなっている。また同時に、世界初の論理型言語による本格的並列オペレーティング・システムである。

高並列システムにおいては、オペレーティング・システムのわずかなオーバーヘッドでも、それが 1 台のプロセッサに集中すれば、システム全体のボトルネックになってしまう。PIMOS は、基本的な資源・実行管理部において徹底した分散管理方針をとることによって、応用プログラムの実行だけでなく、管理自体も分散処理するようにして、このようなボトルネックが生じるのを防いでいる。

PIMOS は、KL1 による並列ソフトウェアの開発に向けたさまざまな機能を提供し、快適なソフトウェア開発環境を提供している。たとえば PIMOS デバッガでは、KL1 言語処理系の機能を利用した自動的なデッドロックの検出や、任意のプログラム部分の実行の一時停止・再開、別ウインドウを用いてのプロセス別のトレースや次第に生成されて行くデータの監視などができる。プロファイラでは、プログラム実行中あるいは実行後に、どのような計算がいつ、どこで(どのプロセッサで)行なわれているのかをグラフィックに表示し、負荷分散状況の容易な把握を可能にしている。これらの機能を利用することによって、知識プログラミング・システム群や機能実証ソフトウェア群の研究開発は大幅に加速された。



統合分子生物学データベース・
インタフェース



Kappa-P によるモチーフ検索の
台数効果

4 並列非正規関係データベース・システム Kappa-P

並列非正規関係データベース Kappa-P は、知識ベース管理システムや知識情報処理システムや遺伝子情報処理システムに対し、知識ベースを格納するための器としてのデータベース管理機能を提供するものである。

知識ベース管理システムや知識情報処理システムや遺伝子情報処理システムが扱うデータは、構造が複雑かつその内容も多種多様で、それに対する操作もさまざまであり、また、データ量も非常に膨大である。Kappa-P は、そのようなデータを効率的に扱うことができるようになっている。

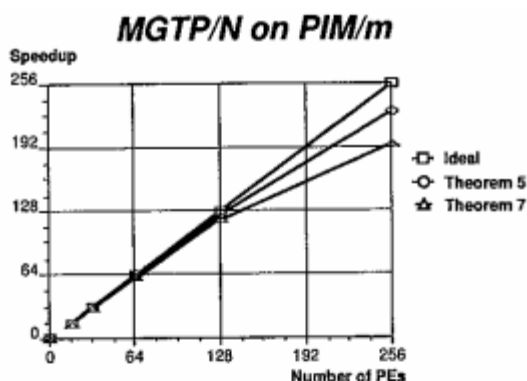
Kappa-P の特徴は、非正規関係モデルと並列処理にある。非正規関係モデルの採用により複雑な構造のデータを効率的に扱うことが可能になっている。また、並列処理により大量のデータを短時間に処理することが可能になる。

これを確かめるために、分子生物学データベースを統合化した蛋白質データベース (配列情報、三次元構造情報等) を設計しその上で動作する応用プログラムを作った。従来の関係データベース管理システムでは、このような複雑な構造のデータベースを効率的に表現、操作することはできなかった。また、複数のプロセッサを使った並列処理により、分子生物学データベースのモチーフ検索などの重い処理を短時間で処理できるようになった。

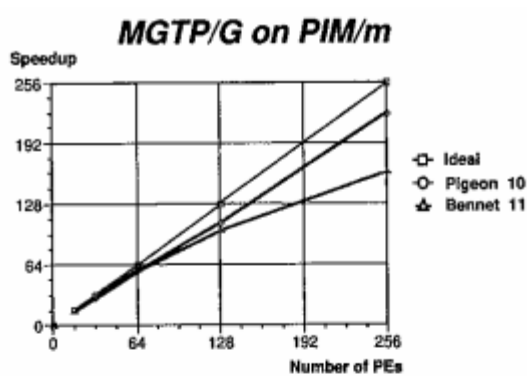
5 並列定理証明システム MGTP

MGTP は高速かつ高機能の自動推論エンジンを提供することを目標に、PIM 上に研究開発されたシステムである。データベース的応用指向及び数学的応用指向の2種がある。前者 (MGTP/G) は有限領域を、後者 (MGTP/N) は無限領域を扱う定理証明系であるが、両者とも性能に関しては世界最高水準にある。

無限領域の定理証明系としては、これまで米国 ANL で開発された OTTER が機能、性能面ともに他を先行していたが、OTTER 方式に比べ、メモリ量、計算量のオーダを数桁下げる新しい方式 (遅延モデル生成法) を開発するとともに、従来困難とされていた分散メモリアーキテクチャ上での並列化方式を実現した。これにより、MGTP/N は PIM/m 単一 PE 上で、SPARC II 上の OTTER の 2~5 倍、PIM/m-



MGTP/N の台数効果



MGTP/G の台数効果

256PE 上では、400 ~ 1000 倍の性能を達成した。

一方、有限領域の定理証明系 MGTP/G については、与えられた問題節を KL1 節に直接コンパイルする手法、並びに場合分けにより生じる組合せの爆発を抑制する手法（ノンホーン・マジックセット）を開発した。MGTP/G でも、MGTP/N と同様に、PIM/m- 256PE 上で 200 倍以上の速度向上を達成している。MGTP/G を用いて、有限代数の問題を試みた結果、カナダの数学者 Benett により提示された準群に関する未解決問題の一部を、自動的に証明することに成功した。

MGTP の機能拡張の一環として、論理プログラミングで開発された「失敗による否定」を MGTP の枠組へ組み込む手法を開発した。本手法は、法的推論システム HELIC II のルールベース・エンジンに採用されている。さらに、本手法の変形として、仮説推論システムを MGTP 上に構築する方式を開発した。まだ実験段階であるが、従来の方式と比較しても、遜色のない結果が得られている。

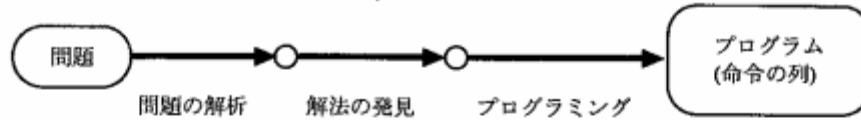
6 並列制約処理言語システム GDCC

従来のプログラミング言語による問題解決では (1) 問題の解析、(2) 解法の発見、(3) 発見された解法のプログラミングの 3 ステップを必要としていた。制約論理型言語は、問題をどう解くかではなく、どのような問題を解くかを記述するプログラミング言語である。このため、問題解決を (1) 問題の解析、(2) 解析された問題のプログラミングという、2 ステップだけで行なうことができる。しかも、従来の言語では解法が想定する場面以外に適用するには新たな解法を発見しなければならなかったのに対して、制約論理型言語では特定の場面を想定せずに、より柔軟なプログラムを記述できる。

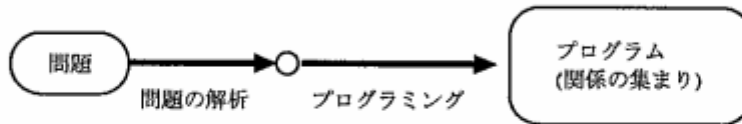
一方、場面に応じた問題解決をシステムが行うため、システムに要求される機能は増大し、結果的にプログラムの実行効率に問題が生じることがある。特に、制約を解く、制約評価系の処理効率はプログラム全体の効率にとって重要な影響を与えるため、この効率化は不可欠である。その点で並列処理を導入することにより、効率化が実現できた。

並列制約論理型言語 GDCC は、このような効率的かつ柔軟な問題解決のためのプログラミング言語を並列推論マシン上にその処理系を構築することにより、より効率的な実行を目指したものである。GDCC の特徴は、世界で初めて非線形代数方程式を扱える制約論理型言語であり、世界で初めて実際の並列計算機上で動作する並列制

従来のプログラミング言語による問題解決



制約論理型言語による問題解決

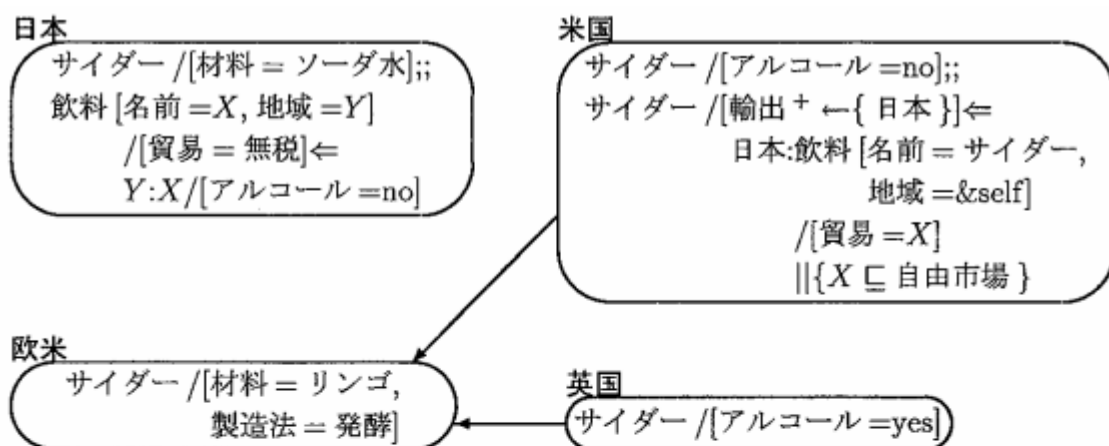


制約論理型言語である点である。

並列推論マシンを利用することにより、非線形代数方程式を扱うシステムは 16PE で約 6 倍の並列効果が得られ、絶対性能についても、実用的なシステムに必要な性能を達成できた。Vidal による処理系や Siegl による処理系といった、他の並列実装と比較してもより効率の良いシステムが得られ、また効率が高いとされている Giovini による処理系、Backelin による処理系のような他システムと比較しても、複数 PE を利用することで、より高速なシステムを実現することができた (次の表を参照)。これらの比較に際しては、比較的処理時間のかかるベンチマークを選んだため、比較対象となる処理系がすべてのベンチマークを解いているわけではないので、それぞれの処理系でデータが得られているものについての比較を行っている。

ベンチマーク	処理系	PE 数				
		1	2	5	12	16
Katsura-4 (単位: 秒)	GDCC の制約評価系	8	8	4	3	4
	Vidal の処理系	17	10	4	4	-
	Giovini の処理系	40	-	-	-	-
Katsura-5 (単位: 秒)	GDCC の制約評価系	84	82	27	17	24
	Vidal の処理系	1103	551	146	79	-
Cyc 4-roots (単位: 秒)	GDCC の制約評価系	1	1	1	1	1
	Siegl の処理系	218	-	-	-	36
Cyc 5-roots (単位: 秒)	GDCC の制約評価系	23	24	12	11	11
	Giovini の処理系	143	-	-	-	-
T-6 (単位: 分)	GDCC の制約評価系	443	453	136	79	66
	Backelin の処理系	90	-	-	-	-

この GDCC を用いて、ハンドリングロボット設計支援システム、階層制約の並列評価システム、ポロノイ図作成支援システムといった応用システムを試作した。それぞれの応用システムにおいて、高い機能と高い効率を実現することができた。たとえば、ハンドリングロボット設計支援システムでは制約論理型言語の持つ柔軟性を生かした、高機能で高効率な支援系を実現し、また、ポロノイ図作成支援システムでは、



その定義を直接プログラムとして記述することで、高機能で、ほぼ PE 台数に比例する台数効果が出るという、高効率のシステムが得られた。

7 知識表現言語システム Quixote

より大規模で実用的な知識情報処理システムを実現するためには、大量で複雑な知識を計算機上で効率良く扱う機能が必要がある。これが知識ベース管理機能である。これを実現するためには以下のことを考慮する必要がある。

- (1) 複雑で雑多な知識をいかに表現するか
- (2) 大量の知識をいかに効率良く蓄積するか
- (3) 蓄積した知識から必要な知識をいかに導出するか

このために、知識表現言語 Quixote を設計し、それによって表現された知識を格納した知識ベースを管理し、必要な知識を導出するためのシステムを開発した (このシステムを Quixote システムと呼ぶ)。

Quixote 言語は、実世界の知識を自然に表現するために、データベース、プログラミング言語、自然言語処理などの複数の分野から検討を加えて設計されたマルチパラダイム言語である。このような知識表現には、オブジェクトの情報は完全であるか不完全であるかによって従来複数の流れがあったが、Quixote はこれらを自然に統合した言語としては世界で初めての言語となっている。

Quixote システムは、上記の知識ベースに対する高度な利用を可能にするために、推論機能を強化し、大量の知識をディスクに格納する機能をサポートしている。前者の機能としては、通常の演繹の他に、知識ベースに仮説を追加して推論する機能や、知識ベースに不足している知識を推論する機能などもサポートしている。このような包括的な機能を持った知識ベース・システムは他に例を見ない。

この Quixote 言語と Quixote システムの有効性は、新しい事件の判決を予測する法的推論、複数の自然言語表現から曖昧性を除去する状況推論、複雑な構造体間の関連を推論する遺伝子情報処理などの応用によって確かめられている。

Quixote で知識がいかに表現されるかを例で示す。たとえば

- (1) サイダーは、日本ではソーダ水の種類だが、欧米ではリンゴから発酵によって作られたものである。
- (2) 米国のサイダーにはアルコールは含まれていないが、英国のサイダーにはアルコールが含まれている。
- (3) 日本では、アルコールの含まれていない飲料については無税である。
- (4) 日本のサイダー市場が開放されているならば、米国は日本にサイダーを輸出する。

のような知識は図に示すように書くことができる (図で示しているが、Quixote による記述量もほぼ同じになる)。

8 法的推論システム HELIC-II

HELIC-II は、新しい事件を与えると、それに対してさまざまな法律適用の可能性を提示するシステムである。一般に、法律に書かれているルールは抽象的なので、与えられた事件にルールをダイレクトに適用することはできない。そこで、HELIC-II はまず与えられた事件と類似する判例を探し、そこで用いられている法適用の論理を利用して抽象的な概念仮説を作りだし、その後でルールを適用する。このシステムは、判例の並列検索と条文の並列適用によって高速な推論を可能にしている。この HELIC-II は、刑法の条文と刑法の判例を格納して、刑法の実問題をシステムに解かせ、その問題解決能力を分析している。このシステムの技術的評価を以下に述べる。

(1) 条文による推論と判例による推論を統合した新しい推論メカニズム

2つの推論モジュールを組み合わせることにより、ルールによる推論の厳密さと事例による推論の柔軟さという利点を活かした推論システムを実現している。なお、ルールによる推論のモジュールは、並列定理証明プログラム MGTP を拡張したものである。また HELIC-II は、事例による推論を初めとする強力な推論方式を実現している。従来、強力な推論手法は時間がかかることが欠点であるとされてきたが、これらを並列化することにより、実用的な時間で結論が求まることを示した。例えば、類似の判例を検出するモジュールでは、PIM の1台のプロセサでは2時間以上かかる計算が、64台では3分程度で実行できている。

(2) 詳細な知識表現と高度な推論技術

刑事事件をコンピュータで扱うには、当事者の認識と事実の食い違い、当事者の意図の時間的変化、1つの行為の立場による解釈の相違、などの詳細な情報を記号として表現することが必要とされる。また、それらの情報を利用するには、類推や仮説の生成や因果関係や時間関係などの推論技術が必要とされる。このシステムでは、刑事事件の実問題の解決に必要な知識表現や推論の手法が開発された。知識表現については、さらに状況理論を用いて形式化を行い、推論技術については、さらに計算モデルとして形式化を行った。その結果、法的推論がどのような性質を持つ推論であるかを理論的に解析することが容易になった。知識表現や推論の要素技術は未だに発展段

階にあるため、汎用で強力な知識表現 / 推論のシステムを開発できる状況にはない。本研究では、法的推論という枠組の中で、個々の要素技術を有機的に統合して強力な推論機構を実現している。しかも、一般性を持たせるために、法律の特殊性をなるべく排除するように工夫されている。また、入力を容易にするためのインターフェース、自然言語による出力インターフェースなども用意されている。

(3) 実問題を解決することにより推論能力を実証

上記の推論メカニズムの有効性を示すため、司法試験の事例問題を解くための知識を分析し、PIMの上で実際に試験問題のいくつかが解けることを示した。試験問題の解答とシステムの生成した解答を比較し、このシステムの持つメカニズムで現実の多くの問題が解決できることを示した。他の法的推論システムの多くは単純化した事件のみを例題として用いていたが、本システムでは、実際の判例を分析して論理の流れを抽出している。その際、標準的な教科書で議論されている学説レベルの詳細度の問題分析を行っている。

(4) 高速な推論を可能にした並列プログラミング技法

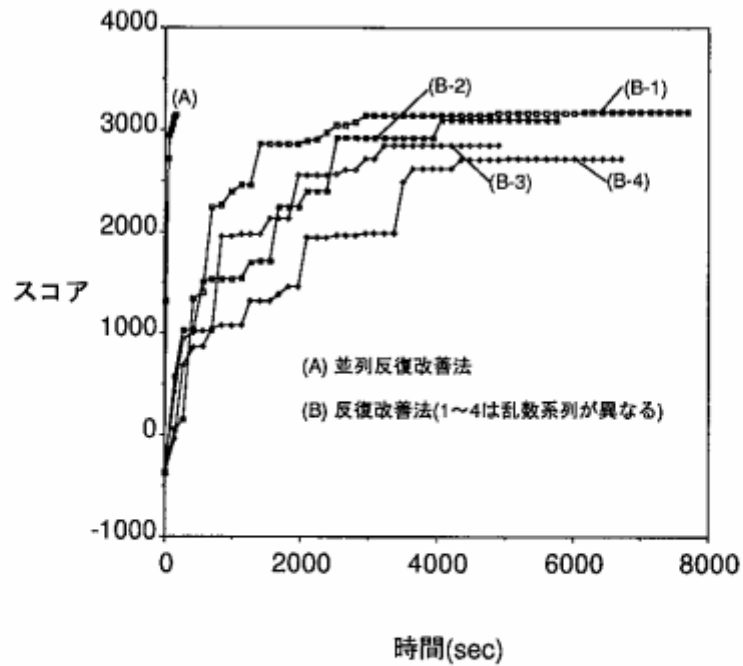
推論システムを効率的に動作させるために、並列プログラミング技法が開発された。例えば、2つの事件の類似性を高速に判定する方法、2つの推論モジュールを並列に走らせる方法、仮説の組合せが不必要に生成されないようにする方法などのプログラミング技法が開発された。

システムの推論モジュール部分（第1版）は並列プログラミング言語 KL1 で約 5,000 行と、従来技術による場合に比してひと桁以上コンパクトな記述、およそ9ヶ月という少ない工数で開発されている。これは、複雑な並列応用システムに対する KL1 の記述力を実証する好例となっている。

9 遺伝子情報処理システム

タンパク質配列の類似性解析技術は、タンパク質の分類、未知の機能の推測、進化の過程の解明に重要な役割を担う技術である。従来は、もっぱら熟練した生物学者が経験と勘を頼りに時間をかけて行っていたが、近年、解析すべきタンパク質配列の数が急激に増加しており、解析技術の計算機による自動化が望まれていた。以前より、いくつかの類似性解析手法が提案され、計算機に実装されてはいるが、生物学的な観点からすると解の品質が不十分であったり、満足すべき解が得られるまで許容できないほどの長時間を要するものが、ほとんどであった。

このシステムは、並列推論マシンを利用し、実用規模の類似性解析を現実的な時間内に実行し、高品質な解析結果を与えるシステムを構築している。本システムは独自に開発した並列反復改善法を用いており、部分的な最適化を並列に、そして反復的に行っている。具体的には、反復の各段階で、可能な部分的最適化を複数、それぞれ各要素プロセサで並列に行い、そのうちで一番評価の良い結果を採用していく方法をとっている。この方法により、乱数で最適化の手順を決め、逐次的に改善を行う反復改善法よりも、256台構成の並列推論マシンで、およそ85倍の高速化を得ている。



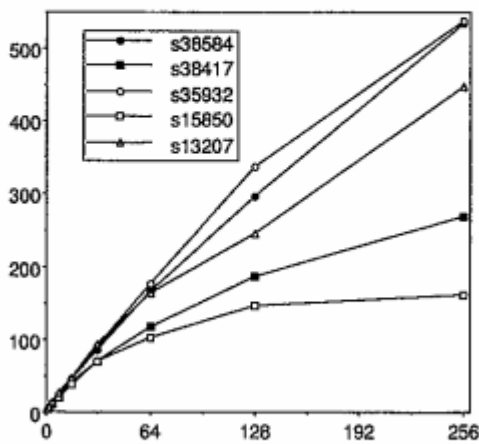
並列反復改善法の効果

10 大規模集積回路設計支援システム

LSI-CAD の工程中、論理シミュレーション、LSI セル配置、および LSI 配線の工程を並列化し、高速化および高品質化を達成した。

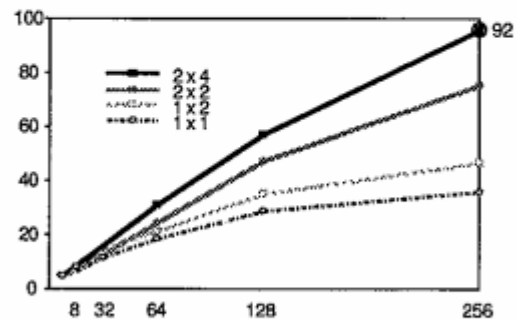
論理シミュレーション、LSI 配線は数日~数か月という多大な処理時間を必要とする工程であり高速化が強く望まれている。また、LSI セル配置では、人手による微妙なパラメータ設定が必要であり、適切な結果を得られない場合が存在する。これらの問題点を解決するために、このシステムでは下記の技術を開発、導入している。

論理シミュレーション 見込み計算 (無駄になるかも知れない計算) を積極的に行



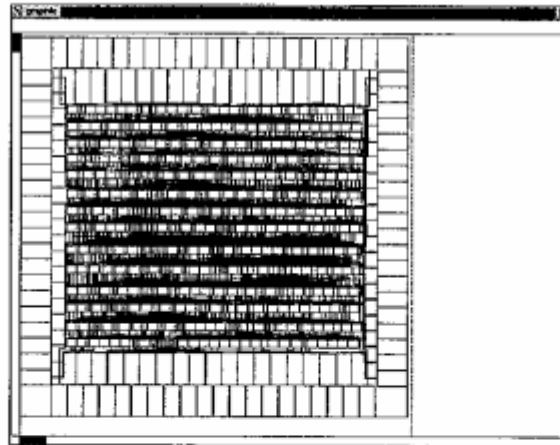
論理シミュレータの台数効果

(縦軸: K イベント/秒; 横軸: プロセッサ数)



LSI 配線の台数効果

(縦軸: 台数効果; 横軸: プロセッサ数)



LSI 配置・配線の結果表示

う機構を導入することで、多大な並列性抽出に成功。また、プロセサ間通信を低減しつつ、負荷の均等化を達成する回路分割アルゴリズムを提案。166 倍の並列処理効果 (256 プロセサ使用時)、数百 MIPS のスーパーコン上のシステム相当の性能を達成。

LSI セル配置 適切なセル配置を求めるためにシミュレーテッド・アニーリング法 (金属の焼き鈍し工程を模擬する方法) を採用し、これにパラメータ設定を自動化する並列アルゴリズムを導入。これによって、極めて最適に近い配置を得る能力を実現。既存のシステムによる結果と比較し、最小のチップ面積を達成。

LSI 配線 オブジェクト指向のモデル化技法を導入することによって、多大な並列性抽出に成功。また、並列に配線経路探索を行いつつ、指定された配線順序を守ることで、配線品質を向上させる機構を提案。92 倍の台数効果 (256 プロセサ使用時)、数百 MIPS スーパーコン上のシステム相当の性能を達成。また、100% の配線率を達成。

11 自然言語処理システム

(1) 概要

第五世代プロジェクトの中期頃までは、適当な例文について人間と機械との対話を行なうシステムを試作した。この結果、人間と機械との対話を円滑に行うためには、従来の文法で扱っている文内の構造のみならず、複数の文にまたがる文脈構造の解析・生成規則が必要なこと、また言語的知識だけでは不十分であり、対象分野やその周辺に関する膨大な背景知識が必要であることが明白になった。その結果、限られた例文の中でトイ・システムの的に全体を通じて動かすことには重点を置かず、個々の要素技術、個別のテーマをさらに深めて研究し、得られた成果を広く世界に公開する方針を取った。

文脈構造の解析・生成については、従来の構文文法の拡充・整備を継続する一方で、いわば文脈文法ともいべき規則の解析収集を行った。背景知識については、

膨大な言語外知識の収集は ICOT における作業では限界があり、むしろ日本電子化辞書研究所 EDR で行われている大規模な辞書、概念辞書の開発、あるいはアメリカの MCC における百科事典的知識の収集、記述プロジェクト CYC などの成果に期待する方針を取っている。

(2) 言語データの蓄積

以上の方針のもとで、文法・辞書を対象に言語データの収集、整備を行った。文法に関しては論理型プログラミングにおける単一化文法の延長上にある句構造文法の枠組みと、従来から特に日本語を対象として適用されてきた係り受け文法の枠組みにしたがって日本語文法を試作した。新しい言語現象に出会うたびに文法を修正していく方針では、文法が大きくなるにつれ体系全体の整合を取ることが困難になるので、日本語の言語現象を分類し、それぞれの現象を押える文法規則を網羅的に記述する方針をとった。この文法に関しては、日本国内のみならず、近年日本語処理に対する興味が高まってきている諸外国の研究機関からの利用希望も数多く寄せられた。

このような文法規則の開発にあたっては、支援ツールの整備が必須なので、作成中の文法を実際に利用して解析を行い、評価、修正を容易にする文法開発支援環境 Linguist や、現実のテキストから指定された言語現象を抽出する KWIC などのツール類を試作した。

また、文脈規則に関しては、新聞社説の文章を対象とし、隣接する文の間関係や文中の語の間の照応関係から文脈構造を抽出する実験を行った。この結果からはいわば文脈文法に相当する文脈解析規則が収集された。

辞書に関しては、形態素情報を中心にした語彙辞書を整備し、約 10 万語、15 万形態素程度の情報を蓄積し、多くのシステムで利用された。

(3) 汎用日本語処理系のツール群の公開

中期に試作したツール群はほとんどが ESP で書かれており、PSI 以外のマシン上で稼働させることはできなかった。そこで、AI 言語研究所 AIR において開発された CESP (Common ESP) に移植することによって、一般の UNIX マシン上で利用可能なものとし、多くのユーザに公開した。CESP に移植したツールとしては、形態素解析システム LAX、文法開発支援環境 Linguist、文生成ツール、KWIC などがある。さらに、いくつかの逐次、並列構文解析ツールに関しても、C、Prolog、KL1 の形態で公開した。

(4) 文生成ツールの拡張

中期までに表層の文にきわめて近い内部データから出発し、それを日本語の文字列に変換する文生成ツールを試作した。しかし、実際にインタフェースとしてこのツールを使い易くするためには、聞き手の内部状態、信念を勘案し、発話すべき内容を組み立てる処理が必要であることがわかった。そこで、発話のためのプランニングを伴う文生成ツールの試作を行った。

このツールは、事実、ルール、判断などを含むシステムの信念に対し、立論ゴールを与えてプランニングを起動する。その結果、聞き手に伝えたい内容を、いわば説得術に相当する言語的・修辭的な戦略に沿って整理して組み上げ、さらに、自分の主

張だけでなく、それを補強するための例示、予想される反駁、反駁に対する反証なども表現する。本ツールは並列マシン上のプログラムとして試作された。

12 プログラム自動合成システム MENDELS ZONE

MENDELS ZONE は形式手法を用いて並列プログラムの開発を支援するシステムである。従来より、ソフトウェアの高信頼性、高生産性を得るために、形式手法が有効であるとの指摘があった。形式手法を用いることで、ソフトウェア仕様の自動検証や調整、仕様からプログラムへの自動変換が可能となるためである。しかし、現実には (1) 検証・調整に要する多大な計算時間、(2) 変換されるプログラムの実行効率の低さが障害となって、実開発への適用は実質的に困難であると考えられていた。この問題に対して MENDELS ZONE では、

- (1) すべての検証・調整手続きを並列化することで計算時間を大幅に短縮、
- (2) KL1 プログラムを生成することでプログラムの実行効率を格段に向上

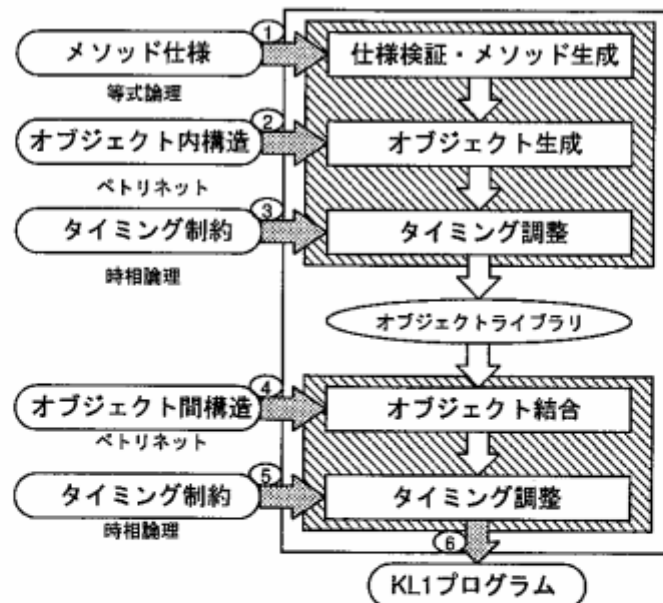
させ、第五世代コンピュータの高い計算能力のもとで、形式手法によるソフトウェア開発を世界で初めて現実のものへと近づけた。これまでの実験では、検証について並列要素数に対してほぼ線形に性能が向上し、生成されるプログラムについては、宣言的な記述から得られるプログラムであっても従来の 10 倍以上の速度で実行されることが確認されている。

また、形式手法によって実システムを記述する際の問題点として (3) システムのさまざまな性質を単一の手法によって記述する難しさ、(4) 大規模システムを形式記述する手間の多さ、(5) 形式記述の理解性の低さなどが挙げられる。これらについては、

- (3) 複数の形式手法を用意し、記述する性質に適した手法を選択、
- (4) 形式手法にオブジェクトの概念を導入し、オブジェクト単位の部品化・再利用を実現、
- (5) 形式記述を図形表示し、記述の理解性を向上

させ、記述性の問題にも対処している。複数の形式手法には、機能を記述するための等式論理と、タイミングを記述するための時相論理、構造を記述・表示するためのペトリネットが用いられる。ペトリネットは、プログラム実行の様子をモニタリングする際にも活用される。

図に MENDELS ZONE におけるソフトウェア開発の流れを示す。(1) では等式論理による機能(メソッド)仕様が入力され、その正しさが並列に自動検証される。(2) ではそれらの構造がペトリネット表示される。(3) では時相論理による機能間のタイミング仕様が入力され、その仕様を満足するように(2)のペトリネットが並列に自動調整される。(4) では複数のオブジェクトがペトリネットエディタを用いて結合され、(5) ではオブジェクト間のタイミング仕様が入力され、(4)のペトリネットが自動調整される。最後に(6)でこれらの仕様が KL1 プログラムへ変換される。



MENDELS ZONEにおけるソフトウェア開発の流れ

MENDELS ZONEの評価を目的としてプラント制御エキスパートシステムを開発した結果、約4,000行の形式記述から6,200行のKL1プログラムが生成され、約5人月を要して目的のシステムが完成した。過去に同一システムを手作業により開発した際のデータと比較すると、プログラムサイズは約34%増加しているが、開発者が記述するコードの量は約15%減少している。開発工数は従来約1/2で、特にデバッグ時間が1/12となっている。これにより、実開発におけるMENDELS ZONEの有効性が確認された。

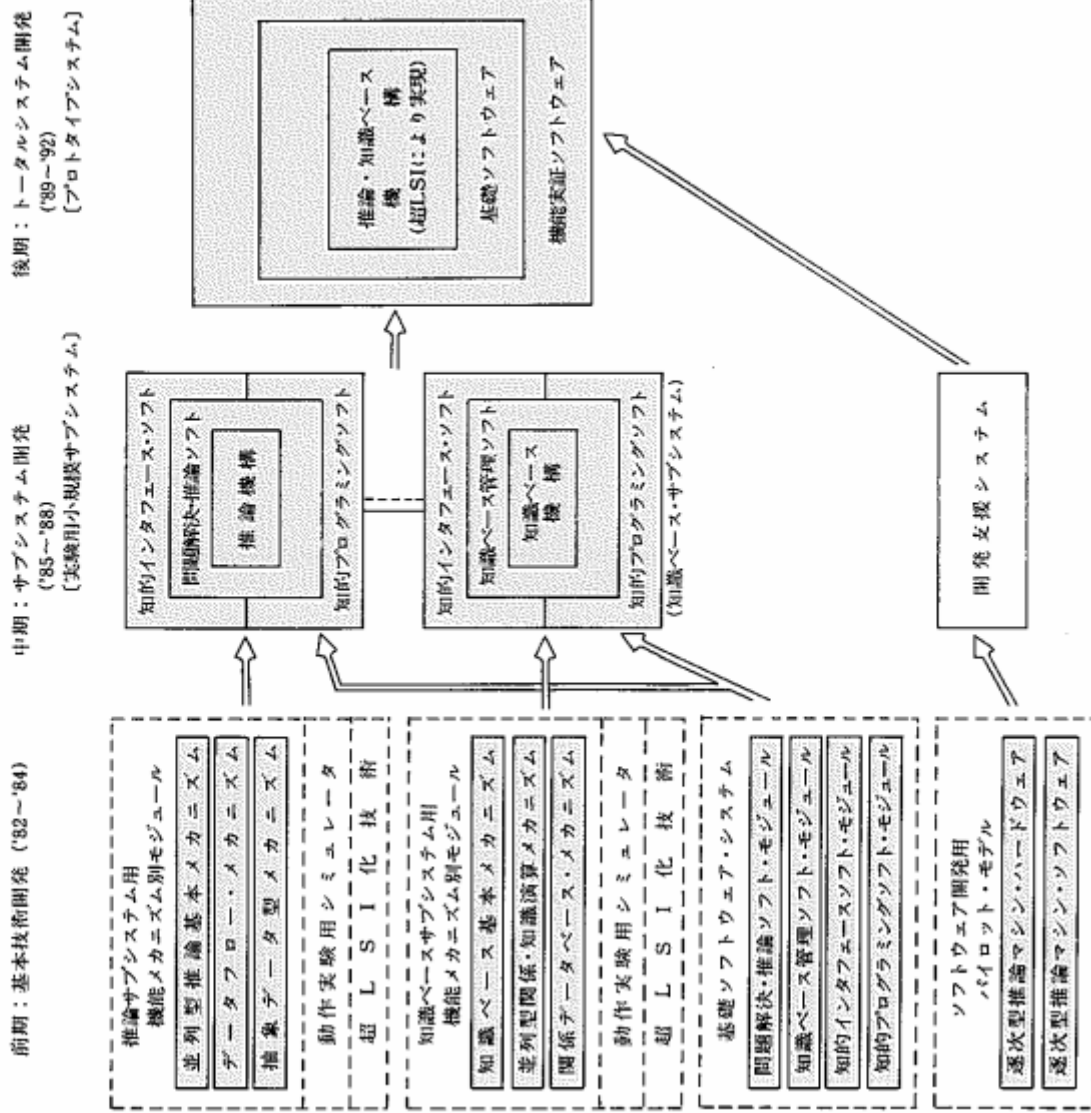
13 棋士システム 碁世代

囲碁の対局を行なう棋士システム。PSI上の逐次版とPIM上の並列版とがある。

コンピュータ囲碁は、チェスなどに代表されるコンピュータゲームプログラムで成功した探索主体の方法では強くなれない。そこで、人間プレイヤーの思考方法のシミュレーションを通じて「碁世代」を開発した。局面認識の方法や、次の着手を考える際に盤面の重要なところに絞り込んで考える点に特徴がある。逐次版「碁世代」はアマチュア中級程度の強さを持ち、1992年の世界コンピュータ囲碁選手権で4位となった。

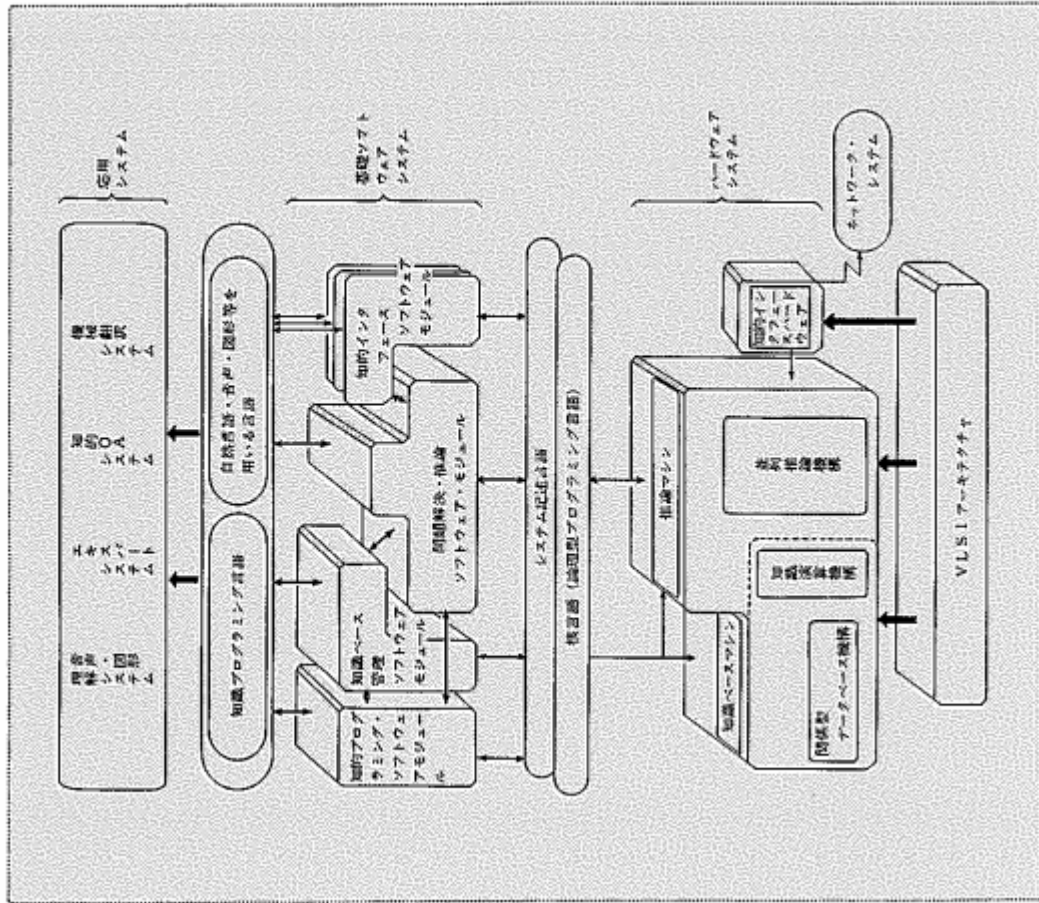
並列版「碁世代」は、世界初の並列処理を取り入れた囲碁システムである。並列版「碁世代」では、次の一着の決定は一定の時間内に収めながら、暇なプロセッサに後で役に立ちそうな別な箇所を検討させる方式を考案・試作し、さらに強いシステムを作れることを実証した。

第五世代コンピュータの開発ステップ

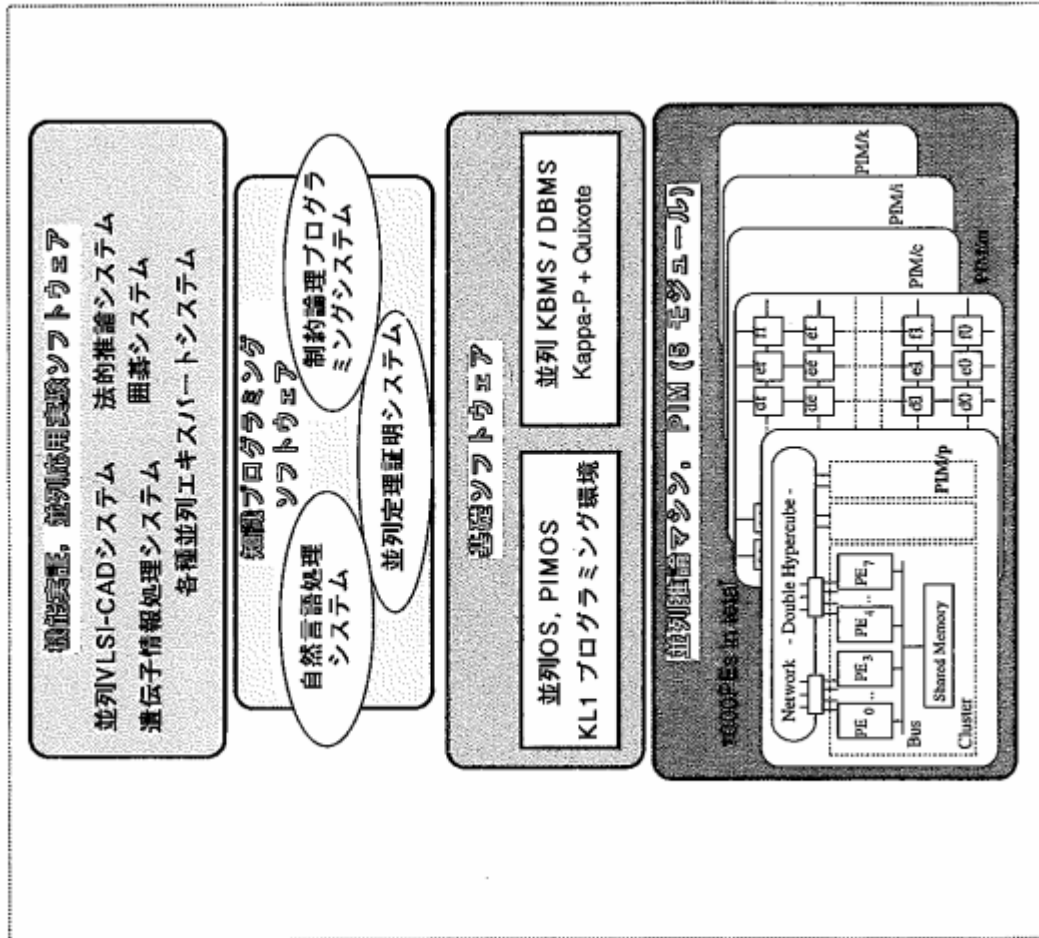


第五世代コンピュータの構成

研究開発構想時の構成概念図



現在の構成

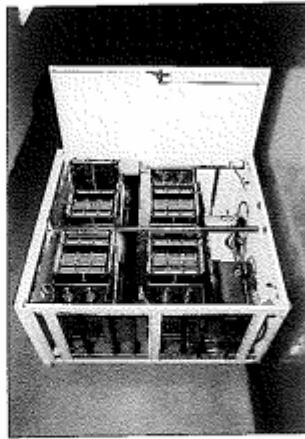


第五世代コンピュータの研究開発

前期 (昭和57年度～昭和59年度) : 基本技術開発

ハードウェア・システム

◇並列推論メカニズムの実現方式の研究のため、8～16並列規模の実験機を試作。



並列処理メカニズム (データプロセッサ) 実験機
→世界初の論理型言語実行用データプロセッサ

◇並列関係演算メカニズムの実現方式の研究のため、関係代数演算の高速実行が可能な関係データベースマシン (Delta) を試作。



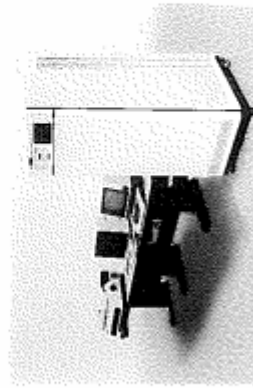
関係データベースマシン (Delta)
→日本で初めて関係データベースエンジンを組み込んだデータベースマシン

ソフトウェア・システム

- ◇論理型言語K L 0 (検査第0版) を試作。この言語仕様に基づき、ソフトウェア開発支援システムとしてP S I を開発。
- ◇K L 0にプログラミング支援機能を入れたE S P (Extended Self-contained Prolog) 言語、E S P処理系第1版を試作
- ◇P S I上のオペレーティングシステム (O S) として、S I M P O S (Sequential Inference Machine Programming and Operating System) 第1版を試作。
→世界初の論理型言語による本格的O S
- ◇逐次型推論コンピュータP S Iおよびその上のソフトウェア・システムであるS I M P O S、E S P言語、E S P処理系から構成される逐次型推論コンピュータシステムを構築し、論理型言語の有効性を確認。
- ◇並列推論メカニズムや並列関係演算メカニズムのハードウェアの試作実験検討を通じて、並列推論を実行する新たな並列論理型言語K L 1 (検査第1版) の言語仕様を設定。

開発支援システム

- ◇論理型言語K L 0新仕様に基づき、パーソナルタイプの逐次型推論コンピュータP S I (Personal Sequential Inference Machine) 及びC H I (Co-operative High Performance Sequential Inference Machine) を開発。
→推論機能をハードウェア化した世界初の逐次型推論コンピュータ



逐次型推論コンピュータM-PSI-I
(パーソナルタイプ)



逐次型推論コンピュータO-PSI-I
(パーソナルタイプ)



逐次型推論コンピュータCHI-I
(大型の高速タイプ)

第五世代コンピュータの研究開発

中期（昭和60年度～昭和63年度）：サブシステムの研究開発

ハードウェア・システム

◇推論サブシステムの研究開発

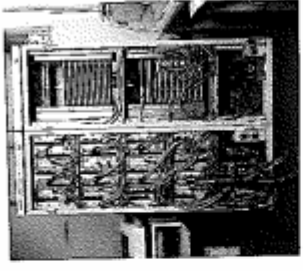
並列論理型言語 K L 1 (後言語第 1 版) を効率的に実行できる並列推論マシン (100 並列規模) のアーキテクチャを確立。64 台の要素プロセッサを接続したマルチ P S I システムを試作して、100 並列規模のアーキテクチャを実証するとともに、十数システムが生産されて、並列ソフトウェアの研究の開発マシンとして、広く利用された。→世界初の並列推論コンピュータ



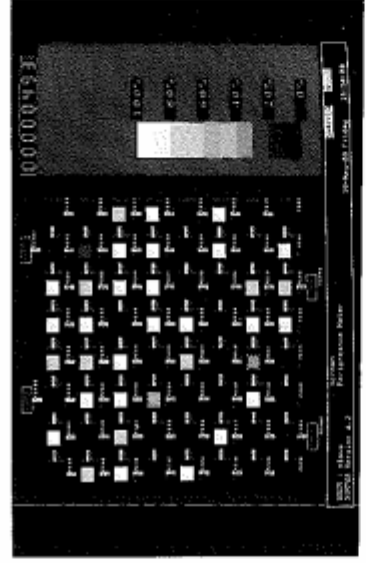
64 台の要素プロセッサを接続したマルチ P S I システム

◇知識ベースサブシステムの研究開発

前期の並列関係演算メカニズムの研究を発展させて、並列知識ベース検索メカニズム実験機を試作。



並列知識ベース検索メカニズム実験機



P I M O S による 64 台の要素プロセッサの稼働状況の表示例

ソフトウェア・システム

◇並列論理型言語 K L 1 (後言語第 1 版) を試作。

◇マルチ P S I システム上に K L 1 処理系を試作。

◇並列推論マシンのオペレーティングシステム (OS) として、P I M O S (Parallel Inference Machine Operating System) 第 1 版を試作。

→世界初の論理型言語による並列 OS

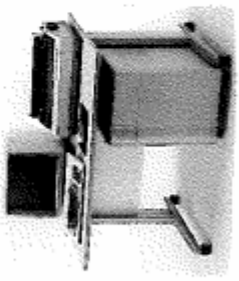
◇マルチ P S I システム上に実装。

◇並列推論コンピュータマルチ P S I システム及びその上の基本ソフトウェア・システムである P I M O S、並列論理型言語 K L 1、K L 1 処理系から構成される並列推論コンピュータシステムを実装させた。このシステムを従って、世界で初めて並列ソフトウェアの本格的な開発に着手することができ、世界に先駆けて並列プログラミング技法や並列知識処理ソフトウェアの作成手法の研究が開始された。

◇また、逐次型推論コンピュータ P S I の小型化・高速化が図られ、その上で自然言語辞書やエキスパートシステム等の大規模な知識ベースの検索・管理を行う並列知識ベース管理ソフトウェア Kappa (Knowledge Application Oriented Advanced Database and Knowledgebase Management System) を試作。

開発支援システム

◇逐次型推論コンピュータ P S I 及び C H I の小型化・高速化が計られ、広く研究開発に利用された。



逐次型推論コンピュータ M-P S I - I



逐次型推論コンピュータ O-P S I - II



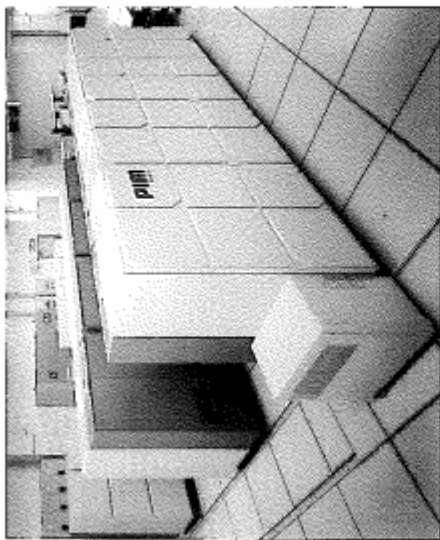
逐次型推論コンピュータ C H I - II

第五世代コンピュータの研究開発

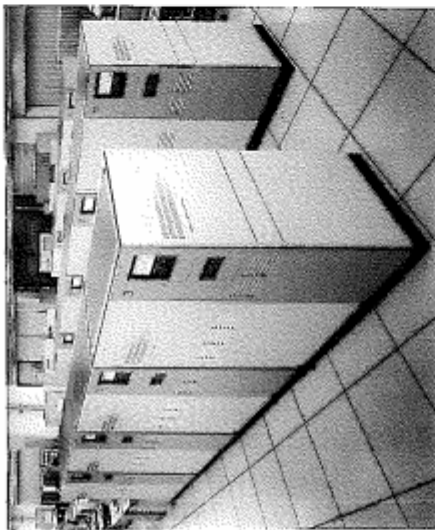
後期 (平成元年度～平成4年度) : プロトタイプ・システムの研究開発

ハードウェア・システム

☆並列推論マシンPIM (Parallel Inference Machine) を世界に先駆けて研究開発。 →第五世代コンピュータは、世界で唯一の、最高速、最大規模の並列推論マシン



PIM/m (256並列)



PIM/p (512並列)



PIM/c (256並列)

ソフトウェア・システム

☆並列推論マシンのオペレーティングシステム (OS) であるPIMOS (Parallel Inference Machine Operating System) をPIM上に実装。
→普遍的な並列OSを目指し、百万並列規模まで見通した基本設計

☆並列知識ベース管理システムKappa-Pの試作。

☆KLL1の上位言語AYA、知識表現言語Quixoteの試作。

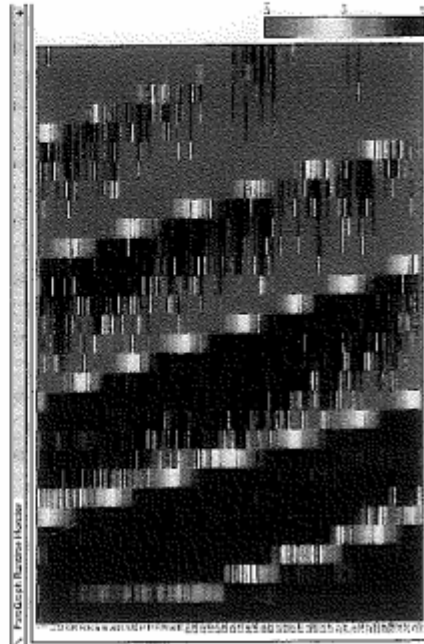
☆並列定理証明システム (高速推論エンジン) MGT P (Model Generation Theorem Prover)、
制約論理プログラミング言語GDCC (Guarded Definite Clauses with Constraints)、
自然言語インタプリタ・ツールLTB (Language Tool Box)の試作。
→MGT Pは現在世界最高速の定理証明システム。数学の分野で世界の研究者が利用を開始。

☆並列応用実験ソフトウェアとして、

- ・遠伝子情報処理システム
- ・法的推論システム
- ・VLSI-CADシステム

等の大規模な知的システムを試作。

→大規模な知的システムの問題解決には、第五世代コンピュータ技術が必須であることを実証



PIMOSの並列プログラム実行可視化機能による表示例

・時間の経過に対応した256の要素プロセッサ (PIM/m) の動作状況

・縦軸は256の要素プロセッサ、横軸は時間。青色は稼働0% → 赤色は稼働100%