

# プロトタイプハードウェアシステムと 基本ソフトウェア

## ICOT

第1研究室 後 藤 厚 宏  
第2研究室 近 山 隆  
第3研究室 横 田 一 正

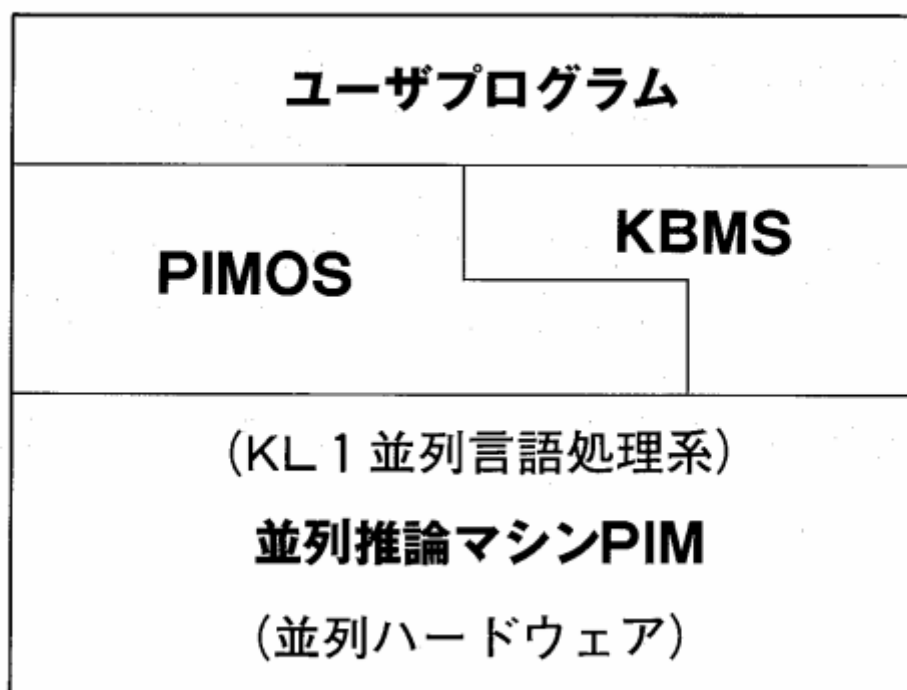
## 講 演 内 容

1. 並列推論マシン(PIM)ハードウェアと  
核言語(KL1)並列処理系 : 後藤
2. オペレーティングシステム(PIMOS)と  
実験的並列応用ソフトウェア : 近山
3. 知識ベース管理モジュール(KBMS) : 横田

## 後期における並列推論システム (5Gプロトタイプ・システム)の目標

- 1000台規模の要素プロセッサを有する  
並列知識処理の本格的実験環境
- 推論性能：数 100 MLIPS
- 高度な並列推論の制御ができるオペレーティング  
システム
- 複雑な知識を表現するための知識表現言語と  
効率の良いデータベース／知識ベース機能

## 並列推論システム



## 平成元年度の研究開発の概要

- 並列推論マシン(PIM)ハードウェア
  - ー 要素プロセッサ開発 ⇔ 組み立て調整を開始
- KL1 並列言語処理系
  - ー 並列ソフトウェアからの要請に沿った改良/拡張
  - ー 新PIM用処理系設計とシミュレータでの試験
- オペレーティングシステム(PIMOS)
  - ー 基本的ソフトウェア開発環境の提供(PIMOS 1.0版)
  - ー より高度な開発環境提供のための拡張改良(→PIMOS 2.0版)
- 実験的並列応用ソフトウェア
  - ー 種々の並列アルゴリズム・マッピング方式の開発と実験
  - ー 中～大規模の実験的並列応用プログラム開発に着手
- 知識ベース管理モジュール(KBMS)
  - ー 並列データベース管理システムの研究開発を開始
  - ー 階層型演繹データベースと遺伝子データベースを試作

## 並列推論マシン(PIM)ハードウェアと

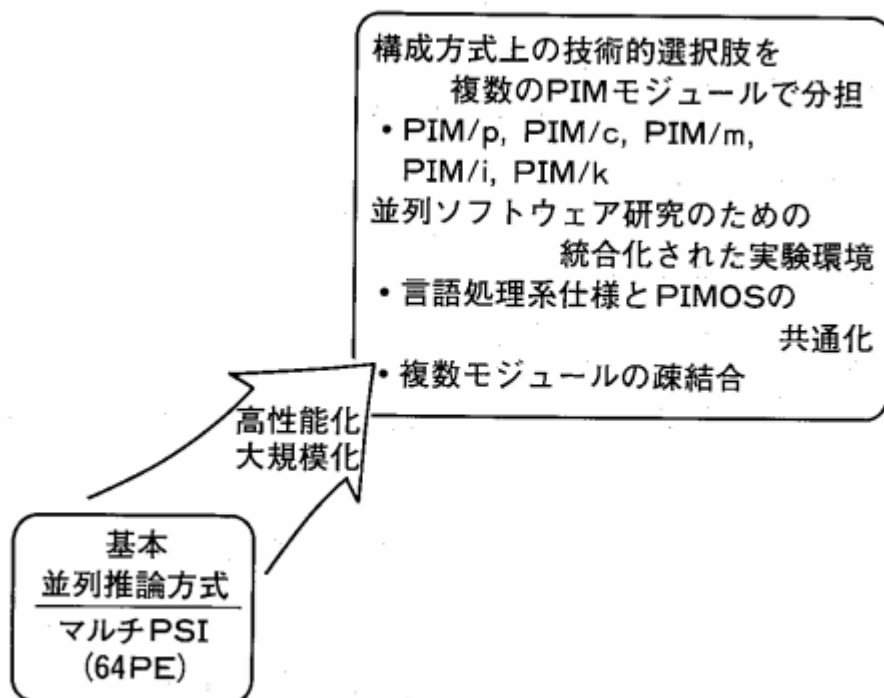
### 核言語(KL1)並列処理系

ICOT

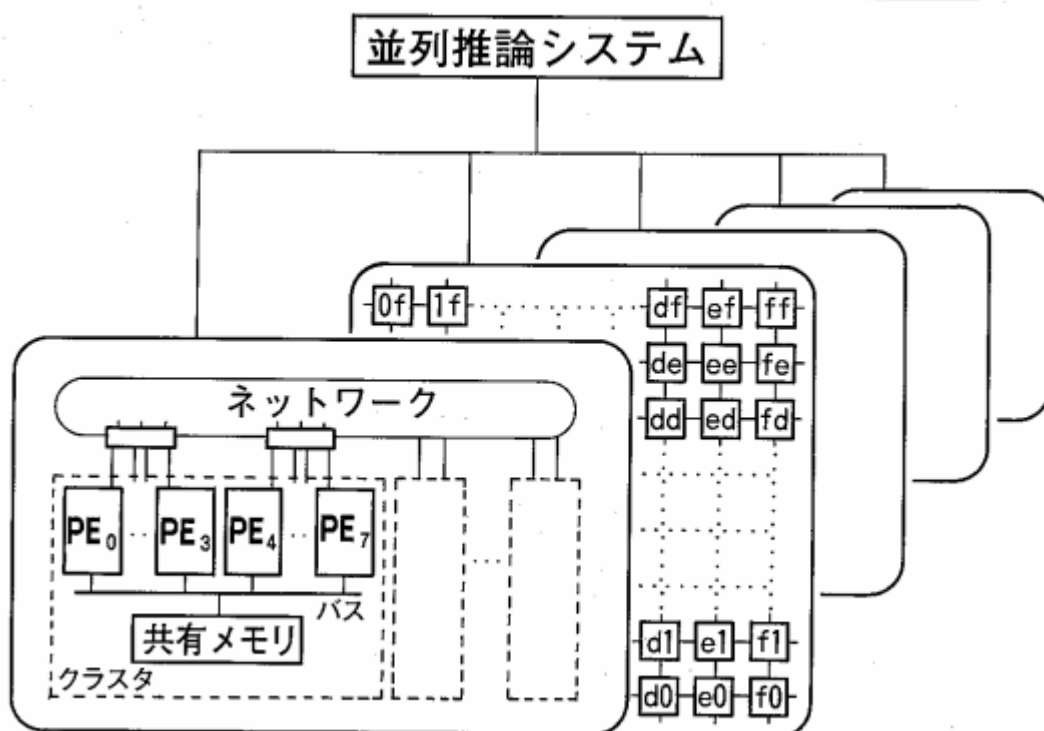
第1研究室

後藤厚宏

## 1000台規模の並列推論マシンへのアプローチ



## 統合化された並列推論の実験環境



## PIMハードウェアモジュールの技術要素

1. 核言語KL1向きマシン命令とプロセッサ構成
  - RISC/CISC (マイクロ命令), パイプライン技術
  - KL1 (記号処理)向きアーキテクチャ
2. 一貫性キャッシュによる共有メモリ結合クラスタ
  - キャッシュ方式 (無効化型/更新型/階層化)
  - 排他制御による共有メモリアクセス方式
3. 大規模結合ネットワークと実装
  - キューブ, クロスバ, メッシュ, etc
  - メインテナンスアーキテクチャ
4. 処理能力に対応した入出力/外部記憶
  - 多数のディスク装置の接続と制御

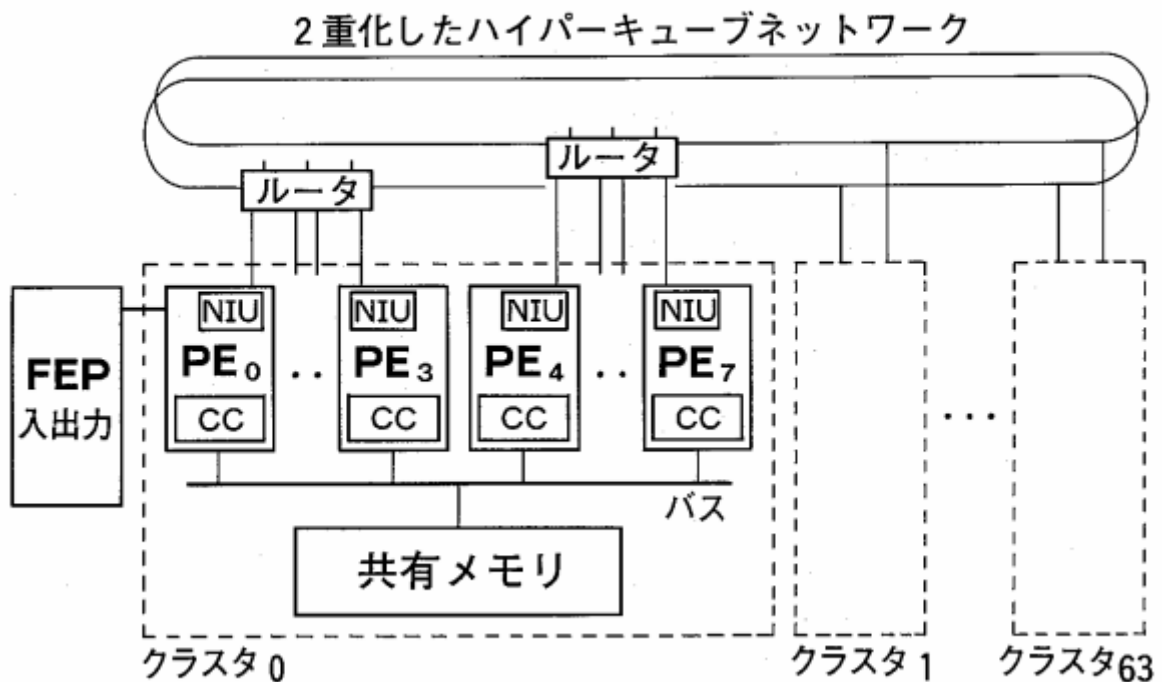
## PIMモジュールの要素プロセッサ

項目	マシン命令 (目標サイクルタイム)	LSIデバイス (線幅)
PIM/p	RISC風+マクロ命令 (60nsec)	スタンダードセル (0.96 $\mu$ m)
PIM/c	水平マイクロ命令 (50nsec)	ゲートアレイ (0.8 $\mu$ m)
PIM/m	水平マイクロ命令 (60nsec)	セルベース (0.8 $\mu$ m)
PIM/i	RISC風 (100 nsec)	スタンダードセル (1.2 $\mu$ m)
PIM/k	RISC風 (100 nsec)	カスタム (1.2 $\mu$ m)

## PIMモジュールのマシン構成の特徴

項目	マシン構成	接続PE数
PIM/p	ハイパーキューブ網による高性能クラスタ(8 PEの共有メモリ結合)の階層型接続	512 PE
PIM/c	クロスバ網によるマルチクラスタ(8 PE+CC+共有メモリ)の階層型接続	256 PE
PIM/m	2次元メッシュ網による大規模ネットワーク結合	256 PE
PIM/i	更新型一貫性キャッシュによる共有メモリ結合	8 PE × 2
PIM/k	2階層一貫性キャッシュによる大容量共有メモリ結合	16 PE × 2

## PIM/pモジュールのハードウェア構成



## KL1 並列言語処理系

PIMOSを含むKL1 プログラマに対して  
高性能なKL1 実行環境を提供

KL1コンパイラ

+

実行時処理系

最適化

- メモリ管理, GC
- スケジューリング
- 負荷分散
- メタ制御機能(荘園)



従来マシンのOS機能の  
一部を含む

## KL1 並列言語処理系の開発

KL1  
並列処理系  
on  
マルチPSI

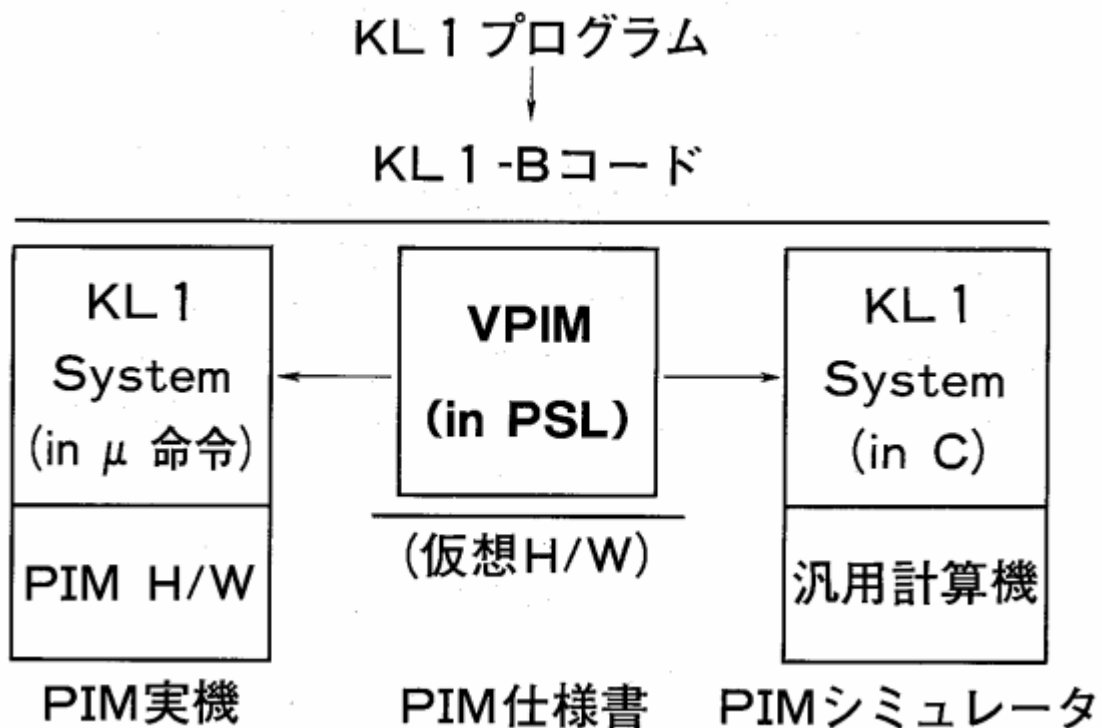
改良  
⇒  
拡張

- 並列プログラミング支援  
デバッグ機能
- 新PIM (PIM/x) への実装  
クラスタ(共有メモリ結合)対応  
異なるネットワーク構造対応
- 並列実行方式の実験  
スケジューリング方式  
負荷分散方式, etc.

## KL1 並列言語処理系の開発システム: PSL/VPIM

- PIMのKL1の実行時処理系仕様(仮想PIM:VPIM)  
⇒PIMモジュール間で仕様を共通化
- VPIM記述言語:PSLと記述支援ツール群  
⇒仕様拡張/変更の容易化
- KL1処理系の試験環境(VPIMシミュレータ)  
⇒インクリメンタルな仕様の確認

## KL1 並列言語処理系の開発システム





## PIMの開発状況

- 並列推論マシン(PIM)ハードウェア
  - ・要素プロセッサ開発 ⇨ 組み立て調整を開始
- KL 1 並列言語処理系
  - ・並列ソフトウェアからの要請に沿った改良／拡張
  - ・新PIM用処理系設計とシミュレータでの試験

## 今後の研究開発

- 全体モジュールの組み立て調整
- KL 1 言語処理系の実装と性能改良



# オペレーティングシステム(PIMOS)と 実験的並列応用ソフトウェア

ICOT  
第2研究室

近 山 隆

## 目 標

- 並列推論処理制御方式の確立
- 並列ソフトウェア研究開発環境の提供
- 並列ソフトウェアの基本技法の開発

## 並列ソフトウェア技術の問題点

	従来の技術	必要な技術
技術開発形態 アルゴリズム 記述言語	逐次技術の延長線上 逐次用を流用 逐次用に機能付加	並列専用技術 新規設計 新規設計
並列動作の指定法	原則：逐次 指定時：並列	原則：並列 指定時：逐次
結果	ぎこちない並列処理	自然な並列処理
単純なシステム	○	◎
複雑なシステム	×	○
逐次技術との連続性	○	×

## 核言語KL1

### 大規模並列知識情報処理システムのための言語

#### 基本部分 = Flat GHC

- 論理型言語 → 「副作用」なし
- データフロー並列 → 暗黙の並列性と同期機構

#### 機能拡張部分

- メタレベル機能 (資源管理, 優先度制御)
- ランダムアクセスできる構造体 (セマンティクスは保存)  
→ 手続き型言語と同じアルゴリズムが使える

# PIMOS : 並列推論マシンのオペレーティングシステム

## 目的

- 並列ソフトウェア研究開発環境の提供
- 並列マシンのオペレーティングのあるべき姿の探求

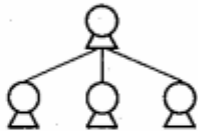
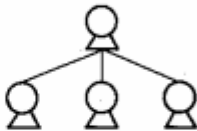
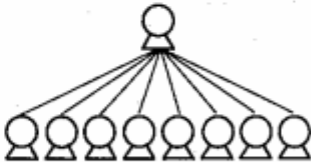

## 方針

- すべてを並列論理型言語KL1で記述
- 疎結合構造を持つ大規模並列計算機を想定
  - 通信コストを意識
  - 集中管理を排除

## 特徴

- 副作用なし・暗黙の同期機構を利用
- システム全体を木構造に分散管理

## 分散管理の必要性

	従来のOS	PIMOS
管理手法	集中管理	木構造分散管理
小規模並列		
大規模並列		
管理の並列性 通信量	小 → ボトルネック アクセス集中により大	大 分散処理により小

## PIMOSの提供形態

**マルチPSI**：最大64プロセサの並列システム  
並列ソフトウェア研究開発環境

**擬似マルチPSI**：1台のPSI-IIでマルチPSIを模擬  
並列ソフトウェアの初期デバッグなど

**リモートアクセス**：  
ネットワーク経由でマルチPSIを利用  
海外からの利用にも対応

→**PIM**：最大512プロセサ程度の並列システム  
第五世代計算機のプロトタイプ

## PIMOSの構成要素と研究開発状況

	管理機能	開発環境	入出力
主な機能	資源管理 入出力管理	プログラム管理 デバッグ支援	外部入出力装置
PIMOS第1版 (1989/7)	基本管理機能	基本機能 (ステップ実行など)	FEP 入出力
PIMOS第2版 (1990/7)	保護機能	デッドロック検出など	遠隔入出力など
PIMOS第3版 (1991～)	マルチユーザ など	性能デバッグ支援 など	ファイルシステム など

## 並列応用ソフトウェアの研究の必要性

**知識情報処理の自動並列化は非常に困難**

( $\leftrightarrow$  数値計算)

→ 並列アルゴリズムの研究開発

**処理の並列性向上と通信，重複／無駄計算コスト低減**

ともに重要だが，両立は難しい問題

→ マッピング(負荷分散とスケジューリング)の研究開発

## 並列応用ソフトウェアの研究開発方針

**基本的並列アルゴリズム／マッピング方式の蓄積**

単純明確な問題に対しアルゴリズム／マッピング方式を開発

→ 性能を解析，有効な方式を蓄積・ライブラリ化

**実験的並列応用プログラムの開発**

実用規模に近い応用を題材に各種並列処理方式を実験・評価

→ 並列処理の有効性を検証

## 平成元年度の研究開発

### **並列アルゴリズム・マッピング方式の開発と実験**

- 多重レベル負荷バランス方式
- 局所通信型プログラムの静的負荷割当て方式
- ヒューリスティック探索

### **中～大規模の実験的並列応用プログラム開発に着手**

- 既存システムの並列化……並列棋士システム
- LSI-CADの問題……配線，論理シミュレーション，最適配置
- 遺伝子情報処理……DNA配列マッチング，蛋白質構造予測

## 研究開発を通して

### **高水準な並列処理用記号処理言語(KL1)で記述**

- 単純な同期のバグは皆無
- 並列システムでもデバッグは容易

### **複雑な解法戦略を必要とする問題を記述**

- 効率的なマッピングはやさしくない(が可能)
- 性能デバッグ支援ツールの必要性

### **社会システムには学ぶ点が多い**

- ← 意識的な並列処理の豊富な経験



# 知識ベース管理モジュール (KBMS)

ICOT

第3研究室

横 田 一 正

## 知識ベース管理モジュールの目的

- プロトタイプ・ハードウェア上での(並列)データベース管理システムの研究開発
- 複数の応用領域の、複雑な知識を表現するための知識表現言語の設計
- それに基づく知識ベース(演繹・オブジェクト指向データベース)システムの研究試作
- (並列)知識情報処理システムへのデータベース/知識ベース管理機能の提供

## 知識ベース管理モジュールのアプローチ

- データベース，知識ベース，応用（インタフェース）の三層に分けて研究開発
- データベースは，処理と表現の効率性を目指し非正規関係モデルに基づき，プロトタイプ・ハードウェアと親和性を持たせる
- 知識ベースは，演繹データベースの拡張としての演繹・オブジェクト指向データベースの枠組で研究試作
- 知識表現言語は，複数応用領域を念頭に置き，論理型言語・拡張項・オブジェクト指向・制約の観点で設計
- 応用とのインタフェースは多くの領域に対応するために拡張可能とし，先端的知識情報処理と結合する

## データベース

目的：プロトタイプ・ハードウェア上でのデータベース管理システム（**DBMS**）の研究開発

成果：－ **PSI**上での（非正規関係）**DBMS (Kappa-II)**の評価とリリース  
－プロトタイプ・ハードウェア上の（非正規関係）**DBMS (Kappa-P)**の機能設計と一部モジュールの試作評価

計画：－大量データに対する効率の良い分散方式と並列処理方式の開発評価  
－並列応用ソフトウェアへのデータベース管理機能の提供

## 知識ベース

目的：複雑な知識を表現するための知識表現言語の設計と、それに基づく知識ベース・システムをデータベース上に試作

成果：—非正規関係に対応する知識表現言語**CRL**の設計試作  
—**CRL**に基づく演繹データベース (**DDB**) と、その階層化**DDB**の試作評価

計画：—演繹・オブジェクト指向データベースを目指した知識表現言語の設計試作  
—複数の応用領域の知識表現言語間の共通要素・共通制約理論の研究  
—上記言語に基づく演繹・オブジェクト指向データベース・システムの設計試作

## 演繹データベースから

### 演繹・オブジェクト 指向データベースへ

- 論理的拡張
  - 否定, 選言, 存在限量子, 空値, 確信度, などの導入
- カプセル的拡張 (データモデル — 受動的オブジェクト)
  - 構造的拡張 — 非正規関係, 複合オブジェクトなどをサポート
  - 手続き的拡張 — データと手続きのカプセル化, 抽象データ型のサポート
  - オブジェクト・アイデンティティの導入
- パラダイムの拡張 (計算モデル)
  - 制約パラダイム — 制約の集合としての, オブジェクトの扱い
  - オブジェクト指向パラダイム — 能動的オブジェクト, 自律性

## 応 用

目的：演繹・オブジェクト指向データベースに基づくデータベース・知識ベースの評価

成果：一遺伝子データベース（DNAの配列データベース**GenBank**）を**Kappa-II**上に格納し，非正規関係の有効性と処理の効率を評価

一配列情報の類似検索を通して，知識情報処理としての可能性を検討

計画：一遺伝子データベースを含む分子生物情報データベースのモデルの検討と、その設計試作

一法的推論などの，高次推論の能力を持つ事例ベース／知識ベースの検討と，そのシステムの設計試作

## なぜ遺伝子情報処理か？

### ● 応用の重要性

一新しいタンパク質(主として医薬品)の設計の可能性

目標タンパク質 ⇒ 該当遺伝子の探索 ⇒ その遺伝子の増殖  
⇒ 目標タンパク質を生成

例：インシュリン(糖の消化酵素)の設計、**AIDS**ワクチンの設計、…

一遺伝病・癌の解明 一例：鎌型赤血球貧血病の解明

タンパク質1ヵ所の生成ミス(ヘモグロビン：グルタミン酸、バリン)

一生命の起源(生物の進化過程)の解明

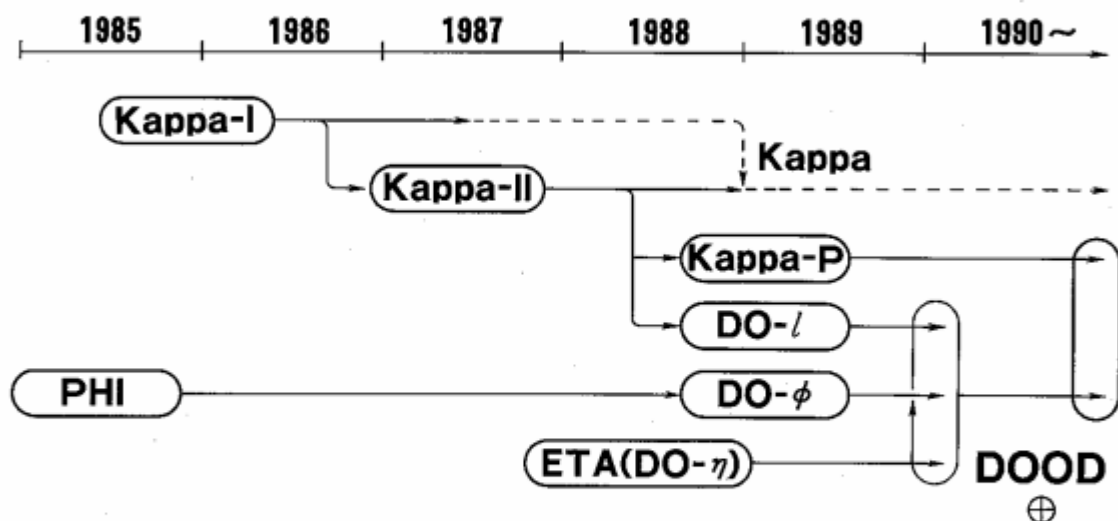
例：チンパンジー (オランウータン科 ⇒ ヒト科)

### ● 情報量の爆発 一例：遺伝子の配列情報 ⇒ 年間倍増のペース

⇒ 知識情報処理による支援の  
必要性

### ● 情報処理としてはまだ未開拓分野 一ICOTの技術との適合性

## 知識ベース管理モジュールの計画



並列知識処理システム  
(自然言語, 証明支援, 遺伝子情報, 法的推論, ...)

## 研究開発の成果・計画のまとめ

### ● データベース

成果: PSI上の(非正規関係)データベース管理システム(Kappa-II)をリリース

計画: Multi-PSI/PIM上の並列データベース管理システム(Kappa-P)を研究開発中

### ● 知識ベース

成果: CRL階層型演繹データベースを試作(DO-l)

計画: 演繹・オブジェクト指向データベース(DOOD)言語を設計中  
複数の応用領域の知識表現言語の共通要素・共通制約理論の研究

### ● 応用

成果: Kappa-II上に遺伝子データベース・システムを試作

計画: 遺伝子情報処理の一環としての分子生物データベースの研究試作  
法的推論などの高次推論機能を持つ知識ベースの研究試作