第4回 シンポジウム

分散KBM

ICOT研究所 第3研究室

大 場 雅 博

第5世代コンピュータプロジェクトにおける知識ベースシステムの研究

前　期(57°～59°)

関係データベースマシン
(Delta1)

中　期(60°～63°)

知識ベースマシン

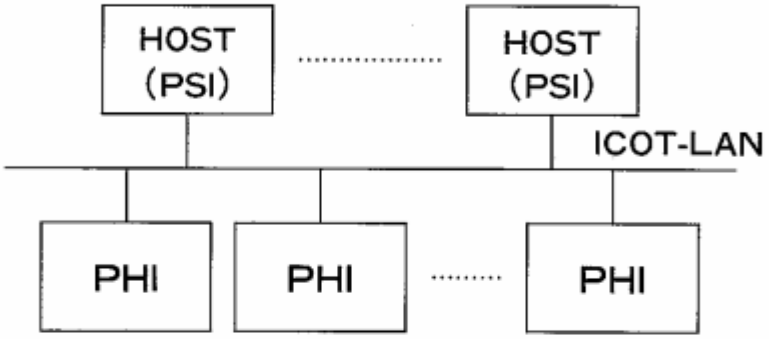
- パイロットモデル(Delta2)
- 分散モデル　(PHI)
- 並列モデル　(実験機)

PHIの開発目標

1．知識ベースの分散処理・管理技術の開発

2．中～大規模知識ベースの管理技術の開発

3．知識ベース高速処理技術の開発



分散知識ベースシステム

PHIの開発方針

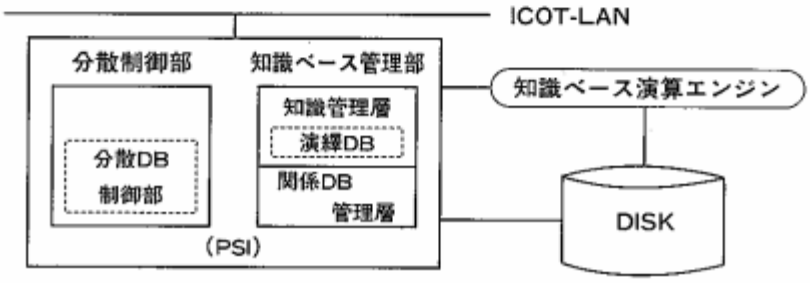
1．分散処理技術による複数の知識ベースの統合

2．論理型プログラミング言語と親和性のある関係DB，演繹DB技術の利用

3．専用ハードウェア化による処理の高速化

4．論理型を基礎にした知識表現

5．ESPによるインプリメント



PHIの構成

分散制御部の基本検討

要求条件

1．ロケーション・トランスペアレンシ

2．分散している知識の一貫性の保持

3．協調問題解決

4．アクセス権，セキュリティ

5．ロバストネス

6．通信オーバヘッドの低減

アプローチ

データ分散から知識分散へ

（分散DB制御からの拡張）

分散DB制御における課題と解決方針

1．ディレクトリ管理……非重複，分散型

2．トランザクション管理

3．分散問合せ処理……問合せのステージ分割

ステージ内での最適化

（転送コスト→低減）

4．コンカレンシィ制御……

事前宣言・2フェーズロック

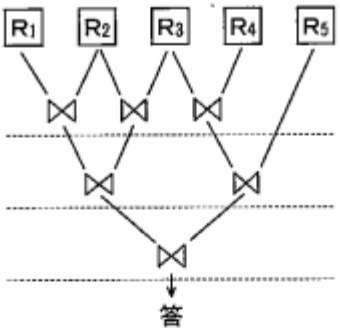
分散型ディテクションによるデッドロック回避

⇑

ブロードカスト型 LAN

分散問合せ処理の最適化方式

転送コストC＝W_1×転送回数＋W_2×転送リレーションサイズ



リレーションの転送時間 ≫
CPUによる計算時間

↓

どのサイトでJOINをとるか

（
・R_2，R_3をブロードカスト　C＝$2W_1+(R_2+R_3)W_2$
・$R_1\rightarrow R_2$，$R_2\rightarrow R_3$，$R_3\rightarrow R_4$　C＝$3W_1+(R_1+R_2+R_3)W_2$
）

知識ベース管理部の基本検討

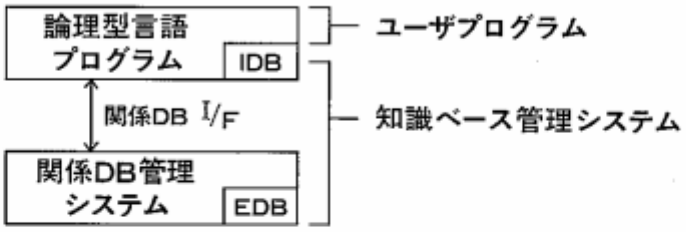
要求機能

1．知識の表現能力
2．知識管理
3．処理効率化

アプローチ

1．論理型言語処理系
　　知識表現能力
2．関係DB管理システム
　　管理機能
　　処理方式（最適化機能／高速アクセス手段）

2階層モデル

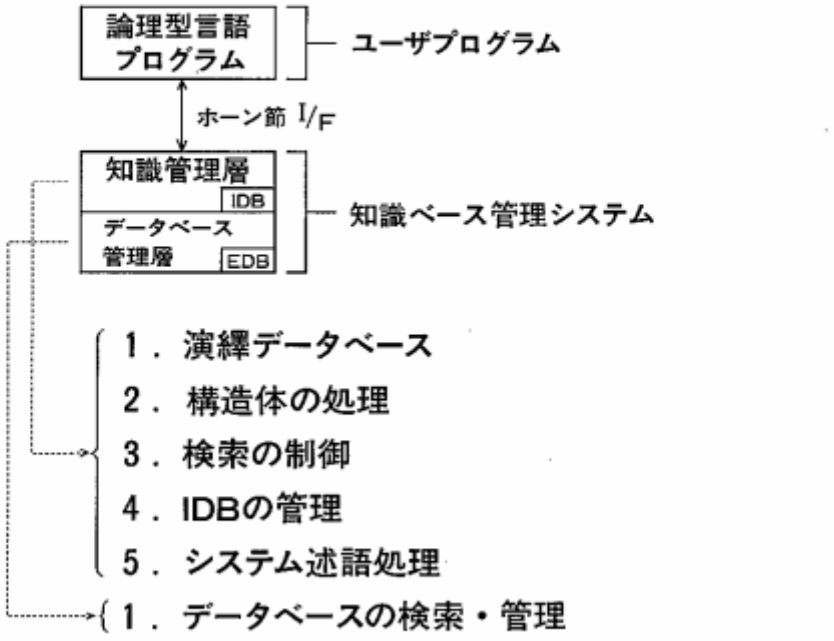


問 題 点

1. ユーザプログラムと知識ベース管理システムの関係が不明確
2. 知識の構造化が不充分，他の知識表現が扱いにくい
3. 知識の格納方法，インテグリティ管理が不充分
4. 実行制御の改良が必要

3階層モデル

(PHIでの知識ベース管理のモデル)



知識ベース管理の階層化による効果

1．論理型知識表現を基礎とし，構造体処理や検索制御機能を実現
2．ホーン節インタフェースにより，ユーザプログラムと知識ベースシステムのインタラクションの低減
3．IDBに対するトップダウン処理とEDBに対するボトムアップ処理の組合せによる処理効率の向上
4．IDBとEDBの論理的分離による分散知識ベース処理の容易化
5．IDBの絞込みを用いた検索空間の削減
6．知識ベースのインテグリティ管理の実現

演繹DBの機能

1．純ホーン節の述語，NOT，比較，算術演算
2．再帰定義
3．任意のゴールに対し，解は“YES/NO”又はファクト中に現われた値の組合せの集合

（例）

```
：－ancestor(X,Y).
ancestor(X,Y)：－parent(X,Y).
ancestor(X,Y)：－parent(X,Z)，ancestor(Z,Y).
parent(X,Y)：－father(X,Y).
parent(X,Y)：－mother(X,Y).
father(X,Y)：－edb(father(X,Y)).
mother(X,Y)：－edb(mother(X,Y)).
```

従来の演繹DBの実現方式と問題点

1．部分コンパイル法

（例）遅延評価法……停止性

edb(father(X,Y)).
edb(mother(X,Y)).
edb(father(X, Z1)), edb(father(Z1,Y)).
edb(father(X, Z1)), edb(mother(Z1,Y)).
edb(mother(X, Z1)), edb(father(Z1,Y)).
edb(mother(X, Z1)), edb(mother(Z1,Y)).
edb(father(X, Z1)), edb(father(Z1, Z2)), edb(father(Z2,Y)).
⋮

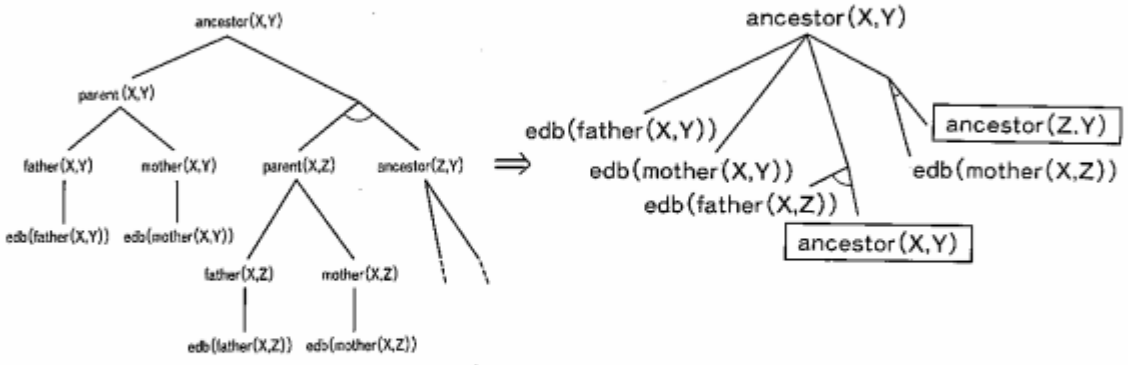
2．完全コンパイル法

（例）セッティング評価法……オーバヘッド

edb(father)
edb(mother)
evaluate in loop
father = edb(father)
mother = edb(mother)
parent = father ∪ mother
ancestor = parent ∪ $\pi_{1,4}$(parent $\bowtie_{2=1}$ ancestor)
end loop

ホーン節変換を用いた完全コンパイル法

（ホーン節変換法）



evaluate in loop
ancestor = edb(father) ∪ edb(mother)
∪ edb(father) $\bowtie$ ancestor
∪ edb(mother) $\bowtie$ ancestor
end loop

ホーン節変換法の特徴と利点

特　徴

1．任意の再帰問合せを単純な繰返し型EDB問合せで実現

2．非再帰型問合せは，非操返し型EDB問合せで処理可能

3．問合せやIDB中の中間的述語の削除による効率的処理

演繹DB拡張上の利点

1．否定を含む再帰問合せの処理が可能

2．構造体処理やカット等の制御機能の組込みが可能

3．述語の順序に依存しない求解が可能

今後の課題

1．分散制御部

(1) 分散DB制御部のインプリメント

(2) 分散知識ベース制御部のイメージ固め

2．知識ベース管理部

(1) 演繹DB＋構造体＋検索制御の実現方式の検討

(2) 演繹DBのインプリメント

3．知識ベース演算エンジン

(1) 基本方式の検討