

# マルチポート・ページメモリを用いた 大規模知識ベースマシンの制御方式

物井 秀俊†、村上 昌己†、伊藤 英則†、横田 治夫‡

†(財)新世代コンピュータ技術開発機構

‡(株)富士通

現在検討している知識ベースマシンの、アーキテクチャと制御方式について述べる。

本稿で述べる知識ベースマシンは、大量の知識を扱うために、二次記憶へ知識を格納することを前提としている。本稿では、知識ベースとして関係モデルの上で決定性一階述語論理式で表現した知識の操作を可能とする関係型知識ベースを提案し、この関係型知識ベースを専用に検索する知識ベースマシンの実験機とその実現方式を提案する。関係型知識ベースを対象とすることにより、複雑な構造を持った知識の操作を、単純な項関係(term relation)演算の組合わせで実現でき、知識の検索を従来のデータベースマシンと同じ枠組みで実現できることを述べる。

マシンとしては、二次記憶上の大量知識を対象とするため、検索処理の効率化と二次記憶と一次記憶間のデータ転送上のボトルネック解消を目的とする。このため、ストリーム処理を行う検索専用プロセッサとこのプロセッサを複数個用いた並列処理により、検索処理の効率化を図る。さらに、複数の検索専用プロセッサと二次記憶装置をマルチポートページメモリで結合することにより、マルチポートページメモリを二次記憶に対するバッファメモリ及び複数プロセッサ間の共有メモリとして使用でき、大量知識に対する効率的な並列検索処理を実現できる。

## A Large-Scale Knowledge Base Machine Control Technique Using Multi-Port Page-Memory

Hidetoshi Mono†, Masaki Murakami†, Hidenori Ito†, Haruo Yokota‡

† Institute for New Generation Computer Technology (ICOT)

‡ Fujitsu Ltd.

The architecture and control of a knowledge base machine are discussed.

The knowledge base machine introduced in this paper aims to retrieve a large-scale knowledge base stored in secondary memory. We have introduced a relational knowledge base model which enables us to treat knowledge as a set and perform retrieval from the knowledge base in the same way as from a relational database. Using a relational knowledge base we can investigate knowledge base machines in the manner of database machines.

The machine consists of a number of dedicated processors called unification engines, several disk systems, and a multiport page-memory. The unification engines perform retrieval from the relational knowledge base. They are executed in the SIMD manner enabling high-speed retrieval. The multiport page-memory is located between dedicated processors and disk systems. It is used as a buffer memory for the disk systems and shared memory for the dedicated processors, enabling efficient parallel retrieval from a knowledge base stored in a secondary memory.

## 1.はじめに

第5世代コンピュータシステムプロジェクトでは、知識情報処理のコンピュータシステム開発のため、推論マシンと知識ベースマシンの実現を図っている。我々は、知識の表現をロジックプログラミングの観点からとらえ、知識の大量格納及び高速検索が可能なる知識ベースマシンを研究開発中である。

これまでの知識ベースシステムの研究は、一次記憶上の知識の表現形式や格納形式及び知識ベースの管理方式等に関するものが中心であり、一次記憶上に格納しきれない程度大量な知識の扱いについては重要視されていなかった。

我々は、大量の知識を二次記憶へ格納することを目指している。このため、知識の集約を目的として扱い、集合上に定義された演算を組み合わせることにより、二次記憶上大量の知識に対する効率的な検索機能を実現することを考える。集合上の演算の組み合わせによる検索の実行は、従来の関係型データベースと同じ枠組みである。我々は、知識の集合を単一化を用いて操作する関係型データベースモデルと、そのモデル上で一階述の論理のサブセットであるホーン節の操作方法を提案した[6][7]。本稿で述べる知識ベースマシンは、この関係型知識ベースを

検索の対象とする。一般に、二次記憶上に置かれた大量データを処理するシステムでは、二次記憶装置と主記憶装置間のデータ転送のボトルネック[1]やデータ処理能力の不足が、性能上の問題となる。この問題を解消するため、データベースマシンでは、ストリーム処理を実行する検索専用プロセッサ、その専用プロセッサを複数用いた並列検索処理、共有メモリを用いた二次記憶装置と専用プロセッサの結合等のアーキテクチャの研究が行われている[4][5]。

複数のプロセッサと共有メモリの結合方式は、バス結合方式、クロスバー結合方式、ネットワーク結合方式等があり、ハた種々の検討がなされている。ここでは、二次記憶装置と専用プロセッサ間のデータ転送能力を最優先し、ネットワーク方式を用いたマルチポートページメモリ[2][3]を採用する。マルチポートページメモリは、二次記憶に対するバッファメモリ及び検索処理の中間結果の一時的格納に使用される。

本稿では、このマルチポートページメモリと検索専用プロセッサを用いた知識ベースマシンの構成方法を中心に述べる。先述せず、2章で関係型知識ベースについて述べ、次に3章でマルチポートページメモリ

と検索専用プロセッサを用いた知識ベースマシンの構成方法について述べる。最後に、4章で現在開発している知識ベースマシン実験機の構成について述べる。

## 2.関係型知識ベース

大量の知識を処理の対象とする場合、知識表現モデルに対する要求は、効率的な共有化や高速な検索を可能とすることである。この要求を充たすためには、知識表現に汎用性を持たせること、及び知識エレメントの一つ一つを意識することなく、知識を集合として取り扱えるようにすることが一つの手法として考えられる。

我々は、関係モデルの上で知識の操作を可能とする関係型知識ベースを提案した[6][7]。このモデルを使用することにより、知識の操作に、関係データベースでこれまでに得られたデータの集約の扱いに関する様々の技術を、取り込むことが可能となる。

関係型知識ベースは、関係データベースにおける関係操作の対象を項(term)に拡張したもので、項関係(term relation)を構成要素とする。ここで、項とは変数を含む一種の論理構造体である。図2.1に項関係の例を示す。この例では、Prologのプログラムを、二つの属性から成る項関係を格納している。即ち、各ホーン節をhead部とbody部に分け、それぞれを別の属性に格納する。尚、一個のホーン節は一個のタプルとして格納する。

項関係に対する操作としては、項間の単一化(Unification)を基本とする以下の演算が定義されている[6][7]。

### ●単一化結合(Unification-Join)

$$C \leftarrow A \text{ att}_1 \bowtie \text{att}_2 B$$

二つ項関係(A,B)の指定された属性(att<sub>1</sub>, att<sub>2</sub>)間で単一化操作( $\bowtie$ )を行い、二つの項関係を結合する。

### ●単一化制約(Unification-Restriction)

$$B \leftarrow \sigma \text{ att} \diamond \text{condition} (A)$$

項関係(A)の指定された属性(att)と条件(condition)の間で、単一化できたタブルのみを抜き出す。

### ●射影(Projection)

$$B \leftarrow \Pi \text{ att}_1, \text{att}_2, \dots (A)$$

関係データベースにおける射影と同じである。

これらの演算は、従来の等号条件による結合や制約演算等の関係操作を単一化操作に拡張した演算である。項関係に対する操作は、見方を変えると単一化による検索であると見ることができ、単一化結合や単一

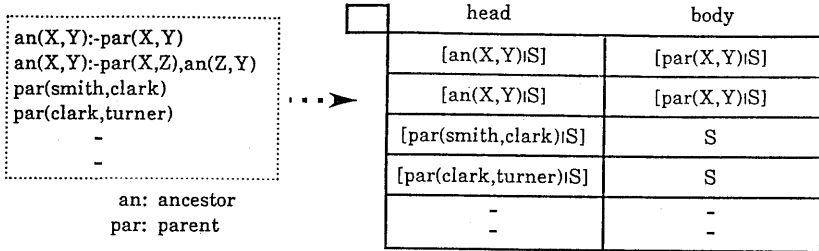


図2.1項関係の具体例

化制約による検索を、RBU (Retrieval By Unification)と呼び、また単一化結合や単一化制約等の演算を総称してRBU演算と呼ぶ。

RBU演算を用いた知識ベースの検索例[7]を、図2.2に示す。

ここで、Tはこの検索で対象となる項関係、goalはクエリ、Rは結果を格納する項関係、及びT', T'', T<sub>i</sub>(i=0,1,2,...)は中間結果を格納する一時的な項関係である。

例えば、Tとして図2.1に示す項関係を与え、goalとして

ancestor(smith,A)

を与えると、smithの祖先Aを項関係Rに求めることができる。

関係知識ベースの検索は、項関係に含まれる変数を、クエリとして与えられたgoalを出発点として単一化により具体化(instanciation)を繰り返し、解を求める操作と見ることもできる。

### 3.アーキテクチャ

#### 3.1処理モデル

2章に述べた基本操作は、項の集合間の関係演算と見なせる。知識ベースマシンでは、大量の項の集合間の関係演算を、効率良く実行することを目的とし、このためのメカニズムを実現することを、アーキテク

チャに対する要求条件とする。

大量知識処理の効率化のために、ページ単位で並列処理を行うことを考える。

充分大きい項の集合A,B間の項関係演算R(A,B)を行う場合、集合A,Bは充分大きいため、以下のように複数ページに分けて格納されていると仮定できる。ここで、複数ページに跨ったタプルは存在しないものとする。

$$A = (a_1, a_2, \dots, a_n)$$

$$B = (b_1, b_2, \dots, b_m)$$

各々のページについての組み合わせによる、項関係演算を実行する。

$$R(a_1, b_1), R(a_1, b_2), \dots, R(a_1, b_m)$$

$$R(a_2, b_1), R(a_2, b_2), \dots, R(a_2, b_m)$$

$$\dots$$

$$R(a_n, b_1), R(a_n, b_2), \dots, R(a_n, b_m)$$

関係型のデータは、処理上の相互作用がないタプル同志の集まりである。このため、関係をタプル方向に分割して複数のプロセスに分配することができる。このページ単位の項関係演算は、互いに実行時にプロセス間のインタラクションを必要としないため、効率的な並列処理が可能である。

いま、k個のプロセスP<sub>1</sub>, P<sub>2</sub>, ..., P<sub>k</sub>を用いると、図3.1に示す方式で並列に処理することができる。

```

R ← Φ; /*結果を格納する項関係の初期化。 */
T0 ← σhead ◊ goal(T); /*ゴールによる単一化制約。 */
i ← 0; /*制御カウンタの初期化。 */
while Ti ≠ Φ do /*中間結果が無くなるまで繰り返す。 */
  begin /*
    T' ← σbody = [](Ti); /*body属性が空のタプルを抜き出す。 */
    R ← R ∪ T'; /*結果の格納。 */
    T'' ← Ti ◊body ◊head T; /*単一化結合。 */
    Ti+1 ← πT''-head ◊ T ◊ body(T2); /*射影により2属性の項関係にする。 */
    i ← i + 1; /*制御カウンタの増加。 */
  end /*

```

図2.2 関係型知識ベースのRBU演算による検索

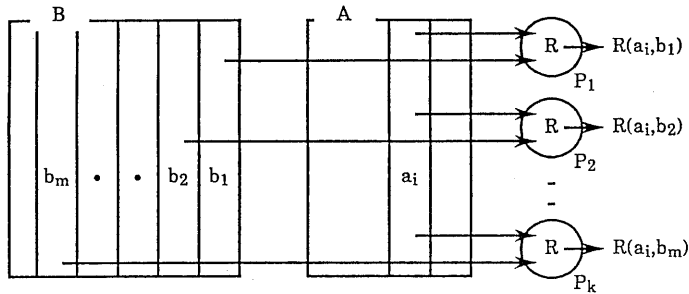


図3.1関係演算の並列処理モデル

図3.1に示す並列処理を効率的に実行するためには、

- ページ単位の項関係演算を実行するプロセッサによる並列処理、
- Aを各プロセッサに分配するメカズム、
- Bを各プロセッサから同時にアクセス可能とするメカズム

が必要となる。

また、データの関係演算と同様に、項関係演算の処理においても、入力となる項関係を整列処理(sorting)後関係演算Rを実行させることが、処理効率の点から必要となる。このためには、項関係A,B双方の全体をまず整列させるか、ページ分割した単位 $a_i, b_j$ で整列させるかについての、考察が必要となる。ここでは、前期に開発したDeltaで行った、大量データに対する整列処理の並列処理効率に関する解析結果[11]から、後者を選択する。

さらに、プロセスへの入力データを削減するため、項関係A,Bをクラスタリングにより絞り込んで、効率化を図ることもできる。

### 3.2アーキテクチャに対する要求条件

3.1節で述べた処理モデルを実現するためアーキテクチャとしては、SIMD (Single Instruction Stream-Multiple Data Stream) [12]を前提とする。

二次記憶上の大量データを、並列に処理するための、アーキテクチャに対する要求条件として、以下のものが考えられる。

- 二次記憶と一次記憶間のボトルネック解消のため、充分なバス幅を確保する。
- 二次記憶に対するアクセスの効率化のため、バッファメモリを置く。
- 複数の検索用プロセッサによる並列処理を効率化するため、多重アクセス可能なバッファメモリを置く。
- バッファメモリは検索処理の中間結果を、

一時的に格納するためにも使用する。

ここに示した(a)~(d)の要求条件は、幾つかのデータベースマシンにおいても提案されている。

関係型知識ベースの検索では、項関係に繰り返し単一化操作を施す処理を基本としているので、中間結果の量とその扱いの考慮が特に必要となる。このため、(d)の要求条件は、従来のデータベースマシンよりさらに重要となることが予想される。

さらに、検索専用プロセッサではストリーム処理を実現する。二次記憶上の検索処理の効率化は、二次記憶装置との入出力を含めて考える必要がある。入出力は検索専用プロセッサの速度に比べて非常に遅く、検索専用プロセッサから入出力を何度も起動していたのでは、処理効率を悪くしてしまう。このため、データをストリームとして流し、専用プロセッサがこのストリームを直接処理できるようにする方が効率的である。

### 3.3知識ベースマシンの全体構成

先に述べた要求条件を満たす知識ベースマシンとして、マルチポートページメモリ(MPPM: Multiport Page-Memory)[2][3]により、複数の検索専用プロセッサと二次記憶装置を結合する構成方式を考える。

現在検討中の知識ベースマシンの全体構成を図3.2[7][9]に示す。この構成は、知識ベースマシンをバックエンド型のマシンとすることを前提としている。ホストマシンとのインタフェース及び検索の並列処理や資源割り当て等の制御は、制御プロセッサ(CP: Control Processor)によって行う。CPは、入出力プロセッサ(IOP: I/O Processor)により、MPPM内のデータをアクセスする。関係型知識ベースは、ディスクシステム(DKS: Disk System)内の二次記憶装置に格納する。また、知識ベースの検索は、検索専用プロセッサである単一化エ

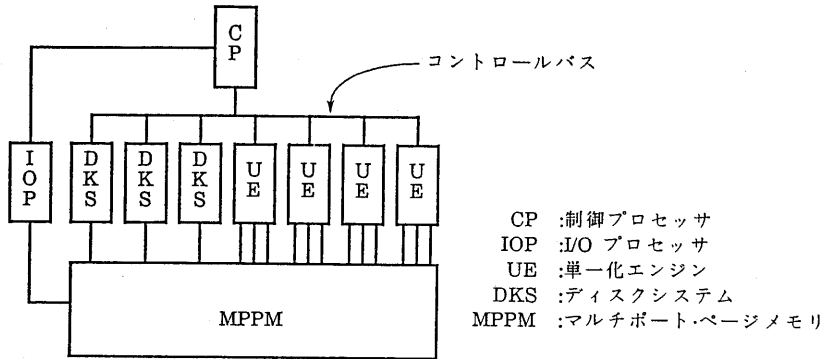


図3.2 知識ベースマシンの全体構成

エンジン(UE: Unification Engine)により行う。

ここで、MPPMは図3.3に示すように、複数のI/Oポートと複数のメモリバンクと、I/Oポートとメモリバンクを接続するスイッチングネットワークから構成される。I/Oポートからのアクセスはページを単位とし、複数のI/Oポートから同一のページを同時にアクセス可能とするため、論理的なページを全てのメモリバンクにまたがるように配置し(図3.3)、スイッチングネットワークにより競合が生じないようにアクセスできる。アクセスがページ単位であるため、スイッチングネットワークの接続関係は、物理的な相対位置を崩さずに一定周期で切り替わる。

このMPPMにより、データ転送及びデータアクセス時の競合がなく複数のUEと二次記憶装置そしてバッファメモリを結合でき、高速な二次記憶に対する検索が可能となる。さらに、MPPMを複数UE間の共有

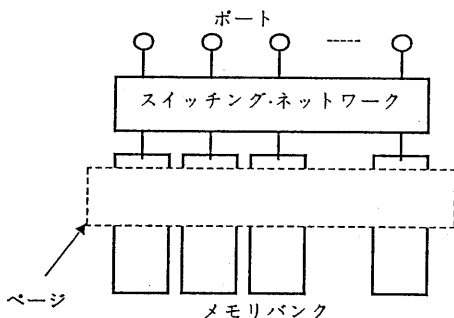


図3.3 マルチポートページメモリの概念図

メモリとして使用することにより、入力となる項関係や中間結果をMPPM上に展開して、効率的な並列処理を実現できる。

次に、UEとして要求される機能は、項関係を入力としてRBU演算を直接実行することである。RBU演算では、項間の単一化結合が最も負荷の重い処理となる。単一化結合では、入力した二つの項関係の全てのタプルの組合わせ(直積)に対して、単一化の可否チェックをし単一化を実施する必要があるので、単一化結合の処理量を削減するため、図3.4に示す構成[8][9][10]で、項の間の順序付けと、この順序付けによる単一化可能な項の組み合わせの絞り込みを行い、単一化処理への入力データの削減を図る。さらに、UEの各構成要素では、後戻りの無いストリーム処理アルゴリズムとその処理機能を実現する。

知識ベースマシン全体の制御とホストマシンとのインタフェース制御はCPが行う。CPは、UE、DKS、IOPというプロセッサ毎に処理コマンドを送出して処理を起動し、各プロセッサから送られる制御情報により処理状態を把握する。ここで、CPと各プロセッサ間の情報のやりとりはコントロールバスを通して行われる。

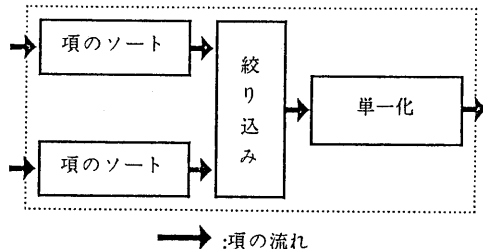


図3.4 単一化エンジンの構成

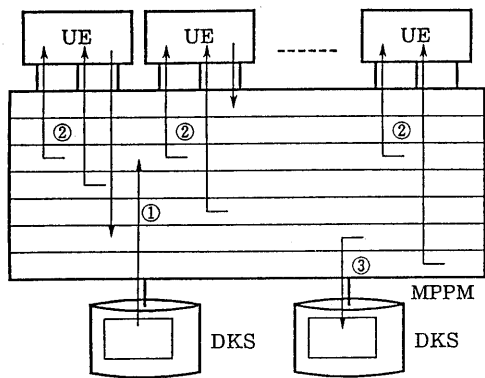


図3.5データの流れ

### 3.4制御方式

図3.2に示す構成において、データの流は以下のようになる。このデータの流を図3.5に示す。

- ①二次記憶装置内のデータを、DKSによってMPPM上にステージングする。
- ②UEは、MPPMからデータをストリームとして入力し、処理結果をストリームとしてMPPM上へ出力する。
- ③UEの処理結果は再びUEへの入力となるか、DKSを用いて二次記憶に書込まれるか、IOPを通してホストまたはCPへ出力される。

ここで、MPPMはDKSに対してバッファメモリとして動作したり、複数のUE間の共有メモリとして動作する。

システム全体の制御としては、並列処理

を実現するためのクエリの分割、及び分割した処理を効率良く実行するためのUEやMPPM等の資源管理が必要となる。制御方式の例を、図3.6に示す。図3.6の各制御モジュールは以下のような機能を持つ。

#### ●コマンド解析部

コマンドとして、関係型知識ベースの検索要求を受け取る。このコマンドの解析により、RBU演算の実行順序と各々の処理フェーズにおける並列度、処理粒度、および処理量の解析を行う。

#### ●スケジュール解析部

並列処理を実行するときの、RBU演算の実行順序に関する制御を行う。

例えば、関係演算の実行において、一方の関係が専用プロセッサのバッファメモリ内に収まりきる程小さければ、もう一方の関係が大きくても効率的な処理ができることが実証されている[11]。この結果より、演算の実行順序としては、関係の大きさを縮小する制約演算(restriction)を先に実行することが考えられる。また、ユーザー特性により決定する必要があるが、使用頻度の高い関係を、プロセッサ内に滞留させる等の考慮も必要である。

#### ●資源管理部

システムの物理資源としては、プロセッサ数、UE内のレジスタ容量、及びMPPMのページ数がある。並列処理においてこれらの物理資源の最適制御を行うためには、これらの負荷状況を監視し、演算特性にあった、動的資源割り当て制御に反映する必要がある。

### 4.知識ベースマシン実験機の開発

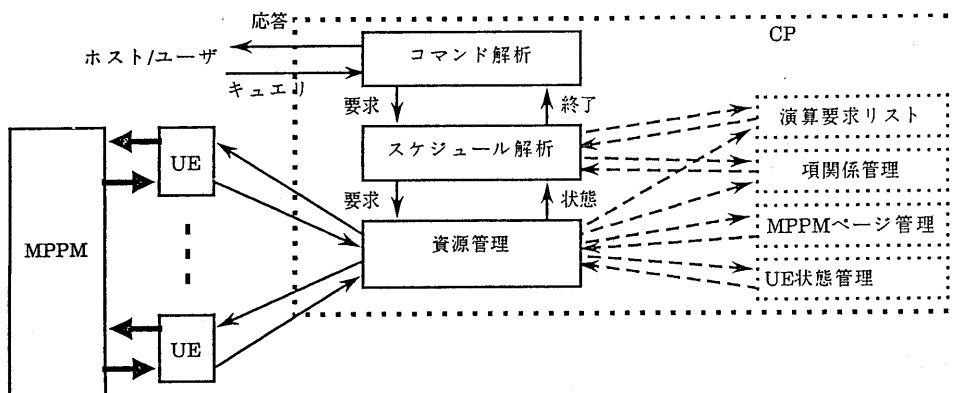


図3.6 CPに於ける知識ベースマシン制御

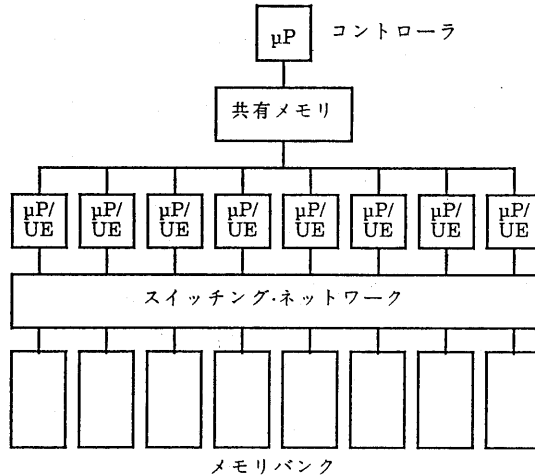


図4.1 知識ベースマシン実験機の構成

知識ベースマシン実験機は、並列処理の実行戦略や並列処理への資源割り当て戦略に関する種々の実験を行うための、実験環境の提供を目的として開発する。

実験機では、幾つかのコンポーネントを開発し各々について実験と改良を重ねて、最終的なシステムへと組み上げる。以下で、このコンポーネントと各々の開発方針及び開発の現状について述べる。

#### (1)知識ベースマシン実験機

実験機の全体構成については図4.1に示す。ここでは、マルチポートページメモリ自体の開発上のノウハウを得ると同時に、関係型知識ベース検索の並列処理に関する種々の実験を行う。

この実験機は、知識ベースシステムにおける並列処理実験環境を提供することを第一目標とし、早期開発を目指している。このため、マルチポートページメモリとしては、ポート数を8×8程度、各ポートの転送速度を4MB/S程度と考えている。また、各ポートにはマイクロプロセッサを結合する。

ハードウェアシミュレータ上で、図3.5に示した制御方式を実現し、項関係に対する並列処理実行戦略や並列資源割り当て戦略に関する実験解析を行う。

#### (2)単一化エンジン

UEについては、項間の順序付けとスタックを用いたページジェネレーション・アルゴリズムによりデータストリームに対して後戻りのない処理方式を提案し、提案した

アルゴリズムをハードで実現する時の問題点や性能上の限界を明らかにするため、ソフトウェアシミュレータの開発を行っている。

このシミュレータによって、項関係処理の定量的な評価を行い、マシン構成に反映する。さらに、項関係に対するRBU演算の実験を行い、RBU演算を効率良く実行するための、多重アクセスメモリ管理や並列処理に関する基礎データを採取する。

ソフトウェアシミュレータと同時に、UE実験機の設計を進め、最終的にはマイクロプロセッサに組み込むかまたはマイクロプロセッサをリプレースする。

#### (3)制御プロセッサ

実験機では、CPとして高い能力を持ったプロセッサは使用せず、マイクロプロセッサにより実現する。将来は、推論マシンにリプレースする。また、この推論マシンを複数結合し、MIMD (Multiple instruction stream-multiple data stream) アーキテクチャへ拡張する。

#### (4)制御プログラム

並列処理のための資源管理や並列実行制御についての、シミュレータによる実験結果を活し、設計を進める。

#### 5.まとめ

関係型知識ベースモデルに対する、知識ベースマシンの構成方式と現在開発中の実験機について述べた。

関係型知識ベースによる知識ベースマシ

ンは、従来のデータベースマシンの発展形態としてとらえることができることを示し、これまでデータベースマシンで採られてきた種々のアプローチが、無理なく知識ベースマシンに適用できることを示した。

本稿で述べた知識ベースマシンでは、二次記憶への知識ベースの格納を前提としている。このため、ストリーム処理を目的としたUEと、二次記憶と一次記憶間のボトルネックの解消を目的としたMPPM結合方式により、二次記憶上の知識の検索を効率的に行うことを目指している。

今後は、ソフトウェアシミュレータやハードウェアシミュレータによる種々の定量的な実験を行う。これにより、制御に必要な基礎データを採取し、制御方式の解析を行い実験機の仕様の詳細化を行う。

#### 謝辞

本検討を進めるにあたり、有益な御示さんを頂いた、知識ベースワーキンググループのメンバの方々及びICOT第三研究室の方々々に感謝いたします。また、熱心な討論を頂いた、東芝のVLKB会議メンバの方々々に感謝いたします。

#### [参考文献]

- [1] Boral, H., and DeWitt, D.J., "Database Machines: An Whose Time has Past? A Critique of the Future of Database Machines", *Database Machines. H.O.Leilich and M.Missikoff (eds.) (Springer-Verlag. Berlin,1983,pp. 166-187)*
- [2] Tanaka, Y., "A Multiport Page-Memory Architecture and A Multiport Disk-Cache System", *New Generation Computing 2*, pp. 241-260, 1984
- [3] 横田、物井、森田、伊藤、"マルチポート・ページメモリの構成方法"、昭和60年度電子通信学会情報システム部門全国大会予稿集、602
- [4] Sabbatel, G.B., Dang, W., Ianeselli, J.C., and Nguyen, G.T. "Unification for a Prolog Data Base Machine", *Proceeding of the Second International Logic Programming Conference*, pp. 207-217, July 1984
- [5] Boral, H., DeWitt, D.J., Friedland, D., Jarrell, N.F., and Wilkinson, W.K., "Implementation of the Database Machine DIRECT", *IEEE Transactions on Software Engineering*, VOL SE-8, NO.6, November 1982, pp. 533-543
- [6] 横田、安部、森田、伊藤、"単一化による知識ベース検索"、昭和60年度電子通信学会情報システム部門全国大会予稿集、S12-2
- [7] Yokota, H., and Itoh, H., "A Model and Architecture for a Relational Knowledge Base", *Proceeding of the 13th International Symposium on Computer Architecture*, November 1985, pp. 2-9
- [8] 森田、横田、西田、伊藤、"単一化結合の処理方式"、情報処理学会第32回全国大会予稿集、1M-7
- [9] 横田、村上、森田、伊藤、"単一化エンジンを用いた知識ベースマシンの構成方法"、情報処理学会第32回全国大会予稿集、1M-8
- [10] Morita, Y, Yokota, H., Nishida, K., and Itoh, H., "Retrieval-by-Unification on a Relational Knowledge Base Model", *to appear in proceedings of the 12th International Conference of VLDB*, August 1986
- [11] Itoh, H., Sakama, C. and Mitomo, Y., "Parallel Control Techniques for dedicated Relational Database Engines", *ICOT Technical Report*, TR-182
- [12] Flynn, M.J., "Some Computer Organizations and Their Effectiveness", *IEEE Transactions on Computer*, Vol.C-21, No.9, pp. 284-294