

知識ベースマシン

Knowledge Base Machine

酒 井 浩 武 脇 敏 晃
Hiroshi Sakai Toshiaki Takewaki

知識情報処理は今後ネットワーク化が進み、複数のサイトが推論に使う知識や途中結果を共有しながら互いに協調して処理することが多くなると考えられる。知識ベースマシン Mu-X は、そのような知識情報処理ネットワークにおいて、複数のサイトで共有すべき知識（データ）を格納するキーコンポーネントを目指す実験機である。Mu-X は、データベース機能と Prolog の知識処理向きの演算機能を兼ね備えており、並列処理による高速化を実現している。なお、本マシンは通産省第 5 世代コンピュータプロジェクトの一環として、（財）新世代コンピュータ技術開発機構と（株）東芝が共同で研究開発した。

Knowledge information processing tends to be integrated in network systems and distributed processing will become popular where several sites shares common knowledge and temporal results during inference.

An experimental knowledge base machine Mu-X aims at a key-component, an efficient storage for knowledge in such a network system. It provides both data base facilities and knowledge processing capability as in prolog. It exhibits high performace employing parallel proccessing. It has been developed by the Institute for New Generation Computer Technology (ICOT) and Toshiba in the Fifth Generation Computer Project.

〔 1 〕 ま え が き

近年、人工知能への関心が高まり、人間の持つ種々の知識を計算機に格納し、その知識をうまく使って知的な処理をさせようとする知識情報処理システムが実用になりつつある。たとえばエキスパートシステムでは特定領域の専門家の知識を *if-then* 型のルールで表現する。そして、*if* 部の条件が現在の状況に適合するルールを見つけ、*then* 部を現在の状況に適用することにより専門家のような高度な診断や設計をさせること目指している。このような知識の集合のことを一般に知識ベースと呼び、それらを用いて結論を得る過程を推論と呼ぶ。

特定の専門領域について実用化が始まった知識情報処理は、研究としては、より広範囲の領域への適用が試みられている。そのような試みのひとつに、お互いに関連のあるいくつかのエキスパートシステムを組合せて、協調して問題解決させる方式がある。たとえば、機械設計を行なうエキスパートシステムと機械加工の容易さを評価するエキスパートシステムを協調させて加工の容易さを考慮した機械設計システムを構築することが考えられる。このような方式は、一般に分散協調問題解決と呼ばれるが、その実用化のためにはエキスパートシステムの間で推論の途中結果を共有する機構が必要となる。

一方、知識情報処理システムは今後ますますネットワーク化が進むものと思われ、上に述べた「共有」の機構はネットワークで結合された計算機群の間でも実現する必要を生ずる。

知識ベースマシン *Mu-X* は、そのような知識情報処理ネットワークにおいて、複数のサイトで共有すべき知識（データ）を格納するキーコンポーネントを目指した実験マシンである。本マシンはデータベースが有する知識（データ）を共有するための機能と *Prolog* の知識処理向きの演算機能を兼ね備えており、複数の処理エレメントで並列処理することにより高速化を実現している。なお、*Mu-X* は、通産省第5世代コンピュータプロジェクトの一環として（財）新世代コンピュータ技術開発機構（*ICOT*）と当社が共同で研究開発した。

〔 2 〕 *Mu-X* の 機 能

複数のサイトの間で知識を共有するには「知識管理機能」とそれに対する問合せの「同時実行制御」が不可欠である。知識管理機能では、種々の知識をある一定の秩序に従って整理し、必要な知識がどこに格納されているか容易に判るようにする必要がある。

問合せの同時実行制御とは、知識の参照や更新の問合せの実行順序を知識の一貫性が保てるよう制御する機能である。たとえば、図1に示すようにサイトAとBがほぼ同時刻にデータNを参照し、Aはそれに1を加え、Bはそれに2を加えようする場合を考える。この場合AとBはそれぞれN+1とN+2を書込もうとするため、どちらかの更新データが

失われるおそれがある。このような事態が発生しないようにするには、あるサイトによるデータの参照から更新までを不可分な処理とし、その間はそのデータに対する他のサイトからの問合せを実行しないようにすべきであり、それを実現するのが問合せの同時実行制御である。なお、管理機能と問合せの同時実行制御はビジネス分野のデータベースでも必要とされる機能である。

一方、知識情報処理の分野ではビジネス分野と比べてより複雑な構造の知識を扱う必要があり、たとえばProlog言語が備えている知識処理向きの演算機能が必要となる。

このような考え方に基づいて、MURXでは知識の格納に非正規型関係モデルを採用している。このモデルはビジネス分野で普及しつつある正規型関係モデル（単に関係モデル、あるいはリレーショナルモデルと呼ばれることが多い）を拡張したものであり、正規型モデルでは数値や文字列のような単純なデータしか扱えないのに対し、「項」というPrologの基本的な構造データが扱える。ここではその違いを従業員に関する従業員番号、氏名、資格の管理を例として説明する。

まず、従業員番号と氏名は、それぞれ数値と文字列という構造のないデータであり、正規型モデルと非正規型モデルで違いはない。しかし、運転免許や税理士などの資格は、全くない人から多くの資格を持つ人まで様々である。また、運転免許には普通や大型第二種などの区別、取得年月日、眼鏡着用といった限定条件など運転免許に特有の情報もある。正規型モデルで、これらの情報を「資格」というひとつの属性で表現しようとするすると図2aに示すように文字列で表現することになる。そのため、「1986年以前に大型第二種免許を取得した人」といった「資格」属性を条件とする検索は困難である。正規型モデルで資格情報を検索などの操作の対象とするには、図2bに示すように「普通運転免許の有無」、「普通運転免許の取得年月日」のように多数の属性にわけて表現することが考えられる。しかし、この方法では資格情報を格納するのに非常に多くの属性が必要となり、また新しい資格ができるごとに対応する属性を追加しなければならないという問題がある。

それに対してMURXの非正規型モデルでは、属性の値として「項」を格納できるため、「資格」属性の表現方法をたとえば次のように決めることにより、うまく処理できる。

まず運転免許など個々の資格の表現方法を約束する。たとえば運転免許は図4aに示すように「運転免許」という3引数関数とその引数で表現することにする、そして第1引数は運転免許の種類、第2引数は取得年月日、第3引数は限定条件を表わすことにする。これを項グラフと呼ばれる図形表現を使うと図4bのように表わすことができる。このように関数とその引数で構成される構造データを一般に「項」と呼ぶ。項の引数はさらに別の項でもよく、たとえば取得年月日は図4cのように「日付」という3引数関数と西暦年、月、日という3つの引数で表わすこともできる。また、限定条件は人により条件の数が異なるので、「項」の一種である「リスト」を使って限定条件を並べあげることにする。リストは項グラフでは図4dのように表わされる「項」のことであり、Prologなどでいくつかの値の並びを表わすのに使用される構造データである。このようにして個々の資格の表現方法を

決めると、あとは「資格」属性に個々の資格を表わす項をリストにして格納すればよい。

Mu-Xは、このようにして格納した項に対し、その要素を取出して条件に適合するかどうか調べるといったPrologの演算機能を備えている。

〔 3 〕 推論マシンとの接続

Mu-XはICOTで開発した推論マシンPSIとLAN (Local Area Network) やDDXなどのネットワークを経由して接続し、使用されるようになっている(図5)。PSIはPrologを拡張した論理型言語ESP (Extended Self-contained Prolog)を直接実行する計算機であり、論理型言語をもとにオブジェクト指向と呼ばれるプログラミング環境を提供している。

PSIからMu-Xに格納した知識(データ)を操作するには次に述べる2つの方法のいずれかを用いる。ひとつは、Mu-Xに格納されている知識をテーブルの形式で表示し、それに対して条件を設定することにより知識ベースの操作を行なう方法である。この方法は、プログラミング言語を意識する必要がないため問合せを素早く作成することができる。もうひとつは、プログラムの中からMu-Xへの問合せを発行する方法である。そのために、オブジェクト指向でクラスと呼ばれる処理モジュールをMu-Xとのインタフェースとして用意した。Mu-Xへの問合せでは、プログラマはこのクラスの使い方について知るだけで良く、Mu-Xやネットワークを意識する必要はない。そのためプログラマは、たとえば端末からキー入力を受けつけるのと同じような感覚で、Mu-Xへの問合せを発行できる。

〔 4 〕 処理方式

4. 1 項データの処理方式

データベースとPrologの機能を兼ね備えているMu-Xは処理方式の面でも両者の方式をベースとしている。項の扱いに関しては、Prolog処理系と同じように各要素にタグと呼ばれる識別情報を付加している。そして項に対する演算の処理方式もProlog処理系と合わせることで、Prologと同じ処理結果を高速に得られるようになっている。

項が扱えるというMu-Xの特徴は、普通のデータベース処理でも下記の2つの点でメリットがある。ひとつは可変長データが扱えるという点である。たとえば、従業員の住所を格納する場合を考えてみる。住所の文字数は従業員によって大きく変わるため、住所を固定長のフィールドに格納するのでは記憶領域の無駄が多い。しかし、Mu-Xでは文字列を可変長データとして扱えるため、格納に必要な記憶領域は実データの文字数に見合った大きさでよい。なお、Mu-Xに格納できるレコード1件の大きさは、最大で64Kバイトである。もうひとつのメリットは「どのような値かわからない」とか「そもそも存在

しない」といった異常値を扱えることである。これらは、たとえば運転免許で取得年月日が不明の場合や従業員データで未婚であっても配偶者の名前を記入されている場合に有効である。Mu-Xではそれらの異常値を‘Not Available’と‘Non Existent’という値で表わし、上記の例では図6のように表現する。個々の演算では異常値の意味を反映した結果が得られるようになっている。

4. 2 計算機アーキテクチャ

Mu-Xは、非正規型関係モデルを採用することにより、従来のデータベースを知識処理向きに拡張したものとなっているが、実験機としての計算機アーキテクチャは可能な限り汎用の構成をとっている。Mu-Xのハードウェアは、図7に示すようにローカルメモリとディスクを備えた8台のPE(Processing Element)と2種類の共有メモリを主要な構成要素としている。ひとつの共有メモリは通常の意味での共有メモリであり、各PEからワード(4バイト)という小さな単位で情報にアクセスできるが、アクセス競合を生じやすくそれが並列処理の隘路となりやすいという短所がある。もうひとつの共有メモリはマルチポート・ページメモリ⁽²⁾であり、ページ(数Kバイト)という比較的大きな単位でしか情報にアクセスできないものの、方式的にアクセス競合は生じないという長所がある。Mu-Xでは各PEで共有すべきデータのうち管理制御用の情報は通常の共有メモリに格納し、演算途中のページデータはマルチポート・ページメモリに格納するという具合にデータの特性に応じて2種類の共有メモリを使い分けている。これにより単一の共有メモリを使った場合と比べてより高い並列処理効果が得られるようになっている。

4. 3 並列処理方式

Mu-Xの並列処理の基本的な考え方は、複数のPEでの並列処理により演算時間を短縮すること、複数のディスクとマルチポートページメモリによりI/O処理を高速化すること、オーバーヘッドの小さな並列制御方式を採用することである。主な特徴は次のとおりである。

- (1) 正規型関係モデルの各種演算の並列処理方式(たとえば⁽³⁾)を非正規型関係モデル向けに改良したものをを用い、演算時間の短縮をはかる。
- (2) リレーション中のレコードを複数のディスクへ分散して格納し、メモリに並列に読出すことによりI/O時間の短縮をはかる。
- (3) PE間の負荷分散は仕事のないPEが共有メモリ上に置かれた制御情報を探すことで実現する。
- (4) 問合せの同時実行制御のための排他制御において、排他制御の対象をリレーション全体と個々のレコードの2階層とすることにより同時実行可能な問合せを多くする。
- (5) オペレーティングシステムが提供する機能は一般に処理オーバーヘッドが大きいのであるべく使わず、処理プログラムの中で同等の処理を行なう。

〔 5 〕 性能評価

Mu-Xのような実験機の開発ではベンチマークテストを行ない、性能評価することが重要である。Mu-Xの場合、独自の非正規型関係モデルを採用しているため、本マシンが得意とするこの種の演算を他のマシンでも行なわせ、結果を直接比較することは難しい。そこで、性能評価は正規型関係データベースを扱う場合について行なうことにし、ウィスコンシン大学で開発されたウィスコンシンベンチマーク⁽⁴⁾を用いることにした。測定には10,000件のレコードを含むリレーションを使用した。各レコードの大きさは208 バイトで、13個の4バイト整数型属性と3個の52バイトの文字列型属性で構成される。なお、このリレーションはキー属性でハッシュした結果に応じて8個のディスクに分散格納されている。測定に使用した演算は次のとおりである。

制約演算：10,000件のリレーションについてある属性の値を条件に100 件の結果を得るような検索を行なう。ただし、条件の属性はキー属性ではないとする。

結合演算：10,000件のリレーションAと1,000 件のリレーションBについて「双方のレコードのある属性の値が等しい」という条件でレコードの組を求める演算（結合演算）を結果が1,000組だけ得られる場合について行なう。ただし、条件の属性はキー属性ではないとする。

キー検索：10,000件のリレーションに対してキー属性によるレコードの検索を行なう。

表1にMu-Xのベンチマーク結果とDeWittらが開発した関係データベースマシンGammaのベンチマーク結果を示す。Gammaは17台のVAX11/750で構成される並列マシンであり、うち8台のVAX11/750にディスクがついている。リレーションはそれらのディスクに分散格納されている。

制約演算と結合演算に関しては、Mu-Xの方がやや遅いことがわかる。この原因はディスクとメモリ間のデータ転送速度が最大で240Kバイト/秒と遅いためである。従って、これを高速化すればMu-Xの性能は非常に良くなるものと期待される。またキー検索に関してはMu-Xの方が高速である。これは主としてMu-Xの方が制御のオーバーヘッドが少ないためと考えられる。

〔 6 〕 あとがき

通産省の第5世代コンピュータプロジェクトの一環としてICOTと当社が共同開発した知識ベースマシンMu-Xについて報告した。Mu-Xは知識を格納するため非正規型関係モデルを採用し、並列処理により高速化をはかっている。

謝 辞

知識ベースマシンM u - Xの研究開発を指導していただいたI C O Tの方々に深謝する。

文 献

- (1) S. Shibayama : Overview of Knowledge Base Mechanism Proc. of the Int'l. Conf. on FGCS 1988 (1988)
- (2) Y. Tanaka : A Mitiport Page-Memory Architecture and A Multiprot Disk Cache System, New Generation Computing, 2 (1984)
- (3) M. Kitsuregawa : Architecture and Performance of the Relational Algebra Machine GRACE, Proc. of the Int'l Conf. on Parallel Processing (1984)
- (4) D. Bitton : Benchmarking Database Systems - A Systematic Approach, Proc. of the 9th Int'l Conf. on Very Large Data Bases (1983)
- (5) D. DeWitt : A Single User Evaluation of the Gamma Database Machine, Proc. of the 5th Intn'l Workshop on Database Machines (1987)

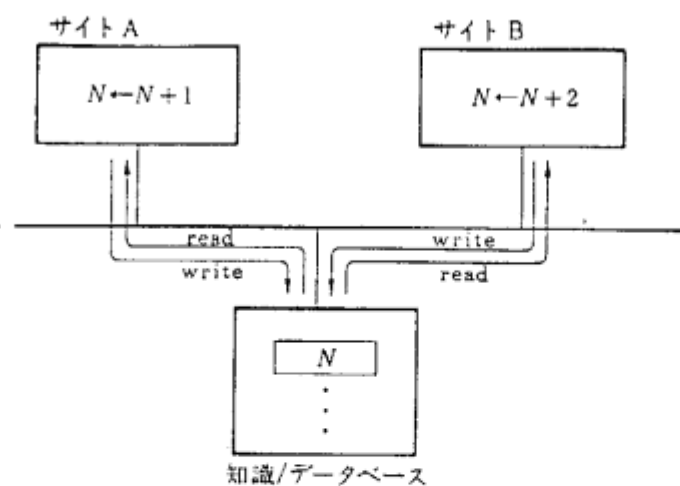


図 1. データの一貫性に関する問題 複数サイトがあるデータを同時に更新しようとする場合、いずれかの更新データが失われるおそれがある。
Lost update problem

従業員番号	氏名	資格	...
1 2 3 4 5 6 7	東芝太郎	運転免許／普通／79年 5 月 24 日／眼鏡等、技術士／情報処理部門／85年...	
2 3 4 5 6 7 8	東京花子	運転免許／普通／84年 8 月 13 日、情報処理技術者／1 種／88年 1 月 30 日、...	

(a) 資格を文字列として表現した例

従業員番号	氏名	普通 運転免許	普通運転免許 取得年月日	運転免許 限定条件	技術士	技術士 部門	技術士 取得年月日	...
1 2 3 4 5 6 7	東芝太郎	有	79年 5 月 24 日	眼鏡等	有	情報処理部門	85年...	
2 3 4 5 6 7 8	東京花子	有	84年 8 月 13 日		無			

(b) 資格を複数属性として表現した例

図 2. 正規型モデルによる従業員データの表現
正規型関係モデルでは、従業員の資格のように構造をもつデータを扱うのが難しい。
Representing employee records in first normal form

運転免許（免許の種類、取得年月日、限定条件）

(a) 運転免許の表現方法

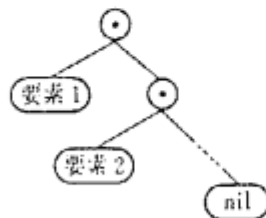


(b) 運転免許の項グラフ

日付（年，月，日）

[限定条件1，限定条件2，…，限定条件n]

(c) 取得年月日，限定条件の表現方法



(d) リストの項グラフ

図3. 項による構造データの表現 従業員の資格は、項でうまく表現できる。

Representing complex data with terms

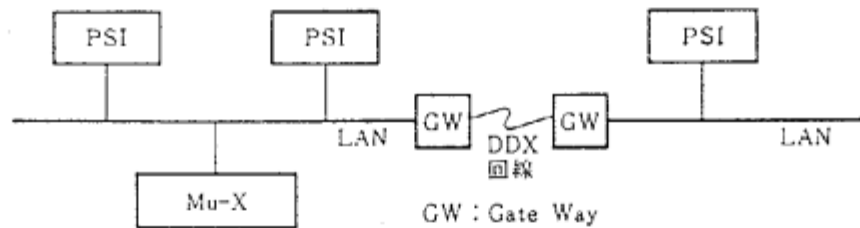


図4. PSIとMu-Xの接続 Mu-XとPSIは、ゲートウェイを経由することによりリモートアクセスも可能である。

Network connection of Mu-X and PSIs

従業員番号	氏名	資格	配偶者	...
9 8 7 6 5 4 3	東芝太郎	[運転免許（普通，'Not Available'，[眼鏡等]）]	'Non Existent'	
2 3 4 5 6 7 8	東京花子	[運転免許（普通，日付（84，8，13），[]）…]	東京三郎	

図5. 異常値を用いた記述例 実世界の情報をデータベース化する場合，異常値がしばしば必要となる。

Representing records with unusual values

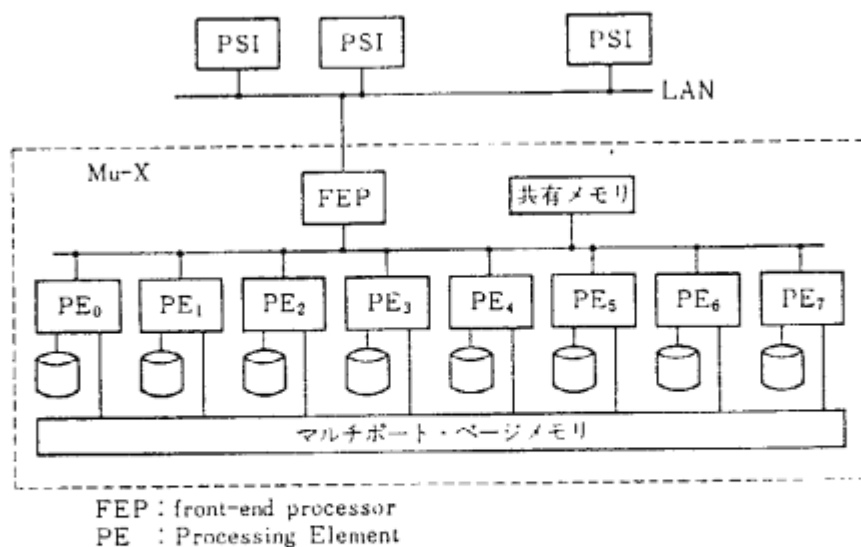


図6. Mu-X のハードウェア構成 Mu-X は、二種類の
共有メモリを備えている。

Hardware configuration of Mu-X

表1. ウィスコンシン ベンチマークの結果
Wisconsin benchmark results

	Mu-X	Gamma
制約演算	1.94 秒	1.56 秒
結合演算	6.4	6.0
キー検索	0.05	0.15