

TM-0155

知識ベースシステム

横田 治夫、伊藤 英則

January, 1986

©1986, ICOT

ICOT

Mita Kokusai Bldg. 21F
4-28 Mita 1-Chome
Minato-ku Tokyo 108 Japan

(03) 456-3191~5
Telex ICOT J32964

Institute for New Generation Computer Technology

1. はじめに

近年, 知識を計算機で取り扱う研究が盛んに行われるようになってきた。知識を使って従来の情報処理をより効率的に行ったり, 新しい情報処理の分野を開拓しようとする試みが各所で行われている。特に, 専門家の知識を計算機内に格納し, その知識を利用して推論を行ない, 十分に複雑な問題を計算機を使って解くシステム (エキスパートシステム) の研究が精力的に進められている。

もともと, エキスパートシステムで格納してある知識の集合を知識ベースと呼んでいた。最近では, エキスパートシステムに限らず, 知識を利用するシステムに使用される知識の集合を知識ベースと総称することが多い。知識を利用して処理を進めるシステムが実際に使われるようになるためには, 大量の知識を整理して格納し, 効率良く検索, 更新できるようにすることが重要となる。さらに, 個々の知識が高価なものになると予想されるために, 多くのユーザあるいはアプリケーションの間で知識を共有することも必要となる。これらの機能は, 今までデータベースの分野で培われてきた機能と重なることが多い。これらの機能を持つ知識の集合は, まさに知識ベースと呼ぶに相応しいといえる。データベースシステムが独立したシステムとして発展してきたように, 知識を体系的に管理する知識ベースシステムを一つの独立したシステムとして開発することが注目されている。

知識ベースシステムと類似した言葉としては, 知識ベースシステム, 知的データベースシステム, 知識データベースシステムなどがある。情報処理における

知識利用の研究が始まったばかりで、各用語の定義が確定しているわけではない。それぞれが重なった意味で使われるこのもあるが、筆者らは以下のようにとらえている。

知識ベースシステムとは、知識を使って処理を進めるシステムのことで、例えばエキスパートシステムも知識ベースシステムの1つである。知識ベースシステムに対するアプリケーションプログラムと見ることもしできる。逆に、知識ベースシステムは、知識ベースシステムのサポートシステムといえる（図1）。

知的データベースシステムは、従来のデータベースシステムをより使い易くするために高機能化したもので、特に知識を格納することを前提としていない。知識を使って高機能化した場合には、知識ベースシステムの1つのアプリケーションと見ることができる。

知識データベースシステムは、従来のデータベースの枠組みの中に知識を格納しようとするもので、知識ベースシステム構成の1つのアプローチである。

本稿では、知識を体系的に格納し、効率的に管理することを目的とする知識ベースシステムを中心に、知識ベースシステムで取り扱う知識の表現方法と、知識ベースシステムが持つべき機能について考察を行ない、知識ベースシステム構築の方法を探る。

2. 知識ベースシステムにおける知識の表現

2.1 知識とデータ

知識とデータはどこが異なるのかということは、知識の表現方法を検討する上で重要な問題である。知識

とデータの違いを厳密に定義することはなかなか難しいが、直感的には、データは必要な情報をそのまま表現しているもので、知識は相互に関連し合った必要な情報を適当な構造に置換えて表現しているものといえる。一番簡単な構造がデータであるとすれば、データを知識の部分集合と見ることもできる。

例えば、ある人物Xの祖先を列記することを考える。この場合、Xの祖先を全て書き並べてしまう方法と、Xの各祖先の親子関係を書き並べて置いて、「祖先とは親もしくは親の祖先のことである」という知識を用いて記述する方法とがある。親か親の祖先が祖先であるということは、何らかの構造を用いて表現する。Xの祖先を検索する場合にはいずれの方法で記述してあってもよいが、それ以外の場合には明らかに知識のほうがデータに比べ柔軟性があることがわかる。Xの親Yの祖先を探す場合には、前者の方法では検索できないのに、後者の方法では問い合わせの内容さえ変えればよい。

上記の例では、データと知識を集合の外延表現と内包表現に対応付けることができる。ただ、知識は必ずしも内包表現のようなものとは限らない。例えば手続きのようなものであってもかまわない。つまり、「年齢とは現在の日付からその人の生年月日を引いたものである」というのも知識である。

2.2 各種の知識表現

知識を表現するための構造とはどのようなものであるべきか。計算機を利用するためには、あらかじめ決めておいたシンタクスで知識を格納する必要がある。このシンタクスは知識の各対象が本来持っている複雑な構造を的確に表現できなければならない。さらにそのセマンティクスは、少なくとも知識ベースを構築す

るものと利用する者との間で統一的に取り決められていなければならない。この知識を格納するために取り決められたシンタクスとセマンティクスは、その知識ベースの表現モデルであり、一般に知識表現と呼ばれるものである。この表現モデルは、実際に取り扱う知識に対して、十分な記述能力を持つ必要がある。

現在、既に各種の知識表現が提案され、エキスパートシステムなどに使われている。特に、述語論理、プロダクションシステム、フレーム、意味ネットワークなどの知識表現が広く利用されている。述語論理は、述語名と引数とで記述するリテラルと論理記号を用いて知識を表現し、事実に関する推論が得意である。論理型プログラミング言語PROLOGは、述語論理の部分集合であるホーン節の演繹を基にしている。プロダクションシステムは、IF (条件) → THEN (行動) という規則で知識を表現するため人間に理解しやすく、エキスパートシステムの知識表現として最もよく使われている。フレームは、型にはまった状況を表現するのに適した知識表現で、枠組みと対象の性質を記述するスロットからなる。意味ネットワークは、ネットワークの節点で概念や対象を表わし、弧でそれらの間の関係を表わす。概念的な知識を表現するのに向いている。

最近では、意味ネットワーク的な知識を述語論理の確定節を用いて表現しようとするDCKR (Definite Clause Knowledge Representation)、述語論理の記述とフレームのスロットを組合せたCIL (Complex Indeterminate Language) などの知識表現も提案されている。

2.3 知識表現の機械語

上述の各知識表現には、それぞれ適した適応分野があり、一つ一つの知識表現がいかなる知識を格納する

のにも万能であるとはいえない。そこで、一つのシステムで幾つかの知識表現をサポートする方法が検討されている。そのような知識ベースシステムの構築方法としては、それぞれの知識表現用の機構を一つのシステムの中に取り込む方法と、一つの基になる計算機向きの表現を用意しておいて、各知識表現で記述された知識をその表現に変換する方法が考えられる。

前者のようなシステムでは、各知識表現に合った機構を作ることができるので、高い処理性能が期待できる。反面、それぞれ異なった知識表現を管理する処理系を用意する必要があるためコスト高となる。また、ある知識表現で格納してしまった知識をそのまま他の知識表現を用いているアプリケーションで利用することが困難になる。

計算機用の一つの特別な表現を用意しておく、処理系はその表現を管理するものの一つでよい。また、どのような知識表現からも変換できるような表現を選定して変換機構さえ備えておけば、異なる知識表現を使う多くのアプリケーションから同一の知識ベースを利用することもできる。この特別な表現は、データベースにおける三層スキーマの概念スキーマに当るものと思ってよい。あるいは、各種知識表現を計算機で取り扱い易くするためのある種の機械語とみなし、変換機構は知識のコンパイラと呼ぶこともできる(図2)。専用ハードウェアを持つような知識ベースマシンには、このような機械語表現を持つ知識ベースシステムが向いていると思われる。

筆者らは、明確に定義された項という構造体を関係(テーブル)の形で格納し、単一化を用いてその項関係を検索する知識ベースモデルを提案しており、各知識表現をそのモデルの上に落とす知識コンパイラを開発しようとしている。

3. 知識ベースシステムに要求される機能

知識ベースシステムと呼ばれるためには、上述したような知識表現が単に格納されているだけでは不十分である。大量の知識を容易に参照できるように整理して蓄積し、外部からの内容の変更要求に対し動的に対応できなければならない。さらに変更に際しては、論理的に矛盾する知識を検出したり、冗長な知識を取り除く機能を持つ必要がある。また、知識ベースの有用性を高めるために、多くのアプリケーションから利用したり、同時に複数のユーザに開放したりするための機能も必要となる。これらは、データベースシステムの分野で研究されてきた機能から類推できるものが多い。そこで、データベースシステムと知識ベースシステムの比較を行ないながら、知識ベースが持つべき機能について検討を進める。

3.1 データベースシステムと知識ベースシステム

当初のデータベースシステムは、現在のように洗練された体系的なシステムではなく、拡張されたファイルシステムのようなもので、検索や更新の機能は決して十分なものとはいえなかった。その後、データベースのモデルが幾つか提案され、データベースがシステムマティックに扱われるようになった。特にCoddの関係モデルの提案によって、データベースが数学的に取り扱われるようになり、データベースの意味論の研究が進んだ。これらモデルの確立と並行して、実際にデータベースを利用するための各種機能の研究も進んだ。例えば、データベースを同時に複数のユーザに開放するための機能、データベースに異常が発生した場合に内容を回復するための機能などは、現在のデータベースシステムにはなくてはならない機能である。これら

の機能およびデータベースの検索機能，更新機能を総称してデータベース管理機能と呼ぶ。

知識ベースシステムにおいてもそれらデータベース管理機能と同様の知識ベース管理機能が必要となる。概念的には，データベース管理の考え方をそのまま知識ベース管理に拡張できるものが多い。しかし，そもそも格納される対象が異なるうえ，データベースシステムは事務管理情報を蓄積することを中心に発展してきたため，事務管理特有の性質を持っており，そのまま変更なしで使うことができる機能はそれほど多くはない。意味を持つ複雑な構造体である知識の取り扱いについての研究が必要となる。

一方，データベースの分野にも研究し残された問題は多い。例えば，null値の取り扱い，インテグリティ制約のチェック，一貫性の制御，問い合わせの最適化などが挙げられる。これらの問題の中には，知識ベースの研究の過程の中で解決され，データベースにフィードバックされるものも出てくると思われる。

3.2 知識ベースの検索

データベースには，外部からデータベースに対する検索を指示するために，各種の問い合わせ言語が用意されている。特に，関係データベースでは，関係代数や関係論理を基にした簡潔な問い合わせ言語が開発されている。知識ベースにも高機能で簡潔な問い合わせ言語が必要である。データベースを知識ベースに発展させる一つのアプローチとして，関係データベースの問い合わせ言語を述語論理に拡張する研究が各所で行われている。

内部の機構としては，大量の知識を効率良く検索するための工夫が必要となる。特に，現在の技術では大

容量の記憶媒体としてはディスクが主流であり、今後メモリ素子がさらに安価になってシステム構成が変化することも考えられるが、当面は二次記憶装置の利用を前提としたシステムの検討をする必要がある。データベースでは、格納データへの高速アクセスのために索引やちらし編成（ハッシュ）などの機構を用いたり、データを体系化（モジュール化）して格納したりしている。また、問い合わせの最適化なども行って高速化を図っている。これらの機能は、知識ベースの場合にも重要となるが、データと知識との違いを十分考慮しなければならない。ことにモジュール化の場合には、知識の意味まで考慮したモジュール化が必要となる。このため、知識ベースに対象指向を導入して、集合としての対象のクラスを多階層に取り扱うことにより、知識の効率的な取り扱いを目指そうという研究もある。

実際の知識ベース検索機能としては、知識の複雑な構造を取り扱うためには、強力なパターンマッチ機能が必要となる。このパターンマッチ機能は人間の知的処理に対応しているともいえる。

3.3 知識ベースの更新

知識ベースを常に最新の状態に保つためには、動的な知識ベースの更新が重要となる。データベースの場合でも、インテグリティ制約のチェックや一貫性の制御、あるいは索引の保守など更新に関する問題は多い。更新を容易にするために、関係データベースではデータ間の従属性に着目した各種正規形が提案されている。知識ベースでは、各対象が構造を持ち互いに関連し合いそれぞれに意味を持っているため、さらに更新が複雑となる。

もとの知識ベースに新しい知識を追加したり、もとの知識ベースを変更しようとした場合、知識ベース全

体の一貫性や無矛盾性が論理的に保たれているか、冗長性がないかを調べることは容易なことではない。これらの問題は、知識獲得の問題として人工知能の分野で熱心に研究されている。述語論理による知識ベースの上では、部分問題に限定しているが、知識獲得の研究で良い成果が得られている。知識ベース全体をある公理体系とみなし、加えられた知識がその公理体系の上で証明できるかどうかで冗長性を、その知識の否定が証明できるかどうかで無矛盾性をチェックするというのが基本的な方針である。

3.4 知識ベースの共有性

一つの知識ベースを複数の異なったアプリケーションから利用することや、同時に複数のユーザに開放することは、知識ベースの有用性を高める上で非常に重要である。

データベースの分野には、データベースを共用するアプリケーション群を安定させるために、データ独立という概念がある。即ち、データベース環境の変化とアプリケーション群を独立させ、データベースの内部スキーマが変化しても外部スキーマには影響を与えないようにすることである。これは、外部スキーマと内部スキーマの間に概念スキーマを入れることにより実現される。このような考え方は知識ベースにも必要であり、データ独立に対応付けて知識独立と呼ぶことにする。ただ、知識独立の場合には、より広義に、各アプリケーションの知識表現と知識ベースシステム内の知識の格納形態とを独立させるために、知識ベース用の知識表現を用いるものとすることもできる。

同時に複数のユーザをサポートするためには、トランザクションという概念を導入する必要がある。トランザクションとは、システムに対するユーザの作業単

位である。同一の対象に対して、同時に複数のユーザがアクセスする場合、書き込んだ内容に上書きされてしまったり、読み出す内容の一貫性が失われてしまったりしないように、制御する必要がある。このため、トランザクション単位で対象にロックを掛けたり、タイムスタンプを用意したりする方法が用いられている。また、ロック方式の場合には、デッドロックが発生する可能性があるので、デッドロックを回避する機構も必要である。知識は、互いに関連し合っているため、細かい単位でのロックは困難であろうと予想される。

この他、複数のユーザに知識ベースを開放するためには、知識ベースの構成を各ユーザに知らせるために、ディクショナリを公開する必要がある。ディクショナリも一種の知識であるから、知識ベースの内容と同様に扱えるのが望ましい。

また、ユーザによっては、知識ベースの内容を全部公開できない場合もあるので、知識ベースの機密管理も重要である。データベースの場合と異なり、各対象が複雑に関連し合っているため、どの部分を見せ、どの部分を隠すかというのは、簡単に決めることができない。

3.5 知識ベースのリカバリ

アプリケーションのプログラムが途中で異常終了した場合には、仕掛かり中の知識ベースの変更は取り消されなければならない。また、知識ベースの内容が、ハードウェアあるいはソフトウェアなどの異常で破壊された場合にも、破壊される直前の正常な状態まで知識ベースの内容を戻す必要がある。

リカバリの方法としては、ログ、シャドウ、差分ファイルなどがある。システム異常に対しては、定期的

に知識ベースのバックアップを取っておき、更新情報のログをそのバックアップに適用することによりリカバリを行うことが多い。アプリケーションの異常終了に対しては、シャドウファイルを作成しておき、異常時に更新の行われていないファイルに変えることによりリカバリを行うことが多い。いずれの場合も、上述したトランザクション単位で知識ベースの内容をコミットすることが重要である。

3.6 メタ知識

知識に対する知識をメタ知識と呼ぶ。知識ベースの検索における最適化の知識、更新における一貫性、無矛盾性、冗長性などをチェックするための知識などが知識ベースのメタ知識に当たる。あるいは、知識ベースのディクショナリなどもメタ知識と呼べるであろう。これらメタ知識は、知識ベースシステムの中核をなすものであり、今後中心的に研究が行われていかなければならない。最終的には、メタ知識も知識の一部として知識ベースの中に格納され、一括して管理されることが望ましい。

4. 知識ベースシステムの構築

知識ベースシステムを構築する場合には、フロントエンドの知識ベースシステムに対して、バックエンドシステムとして構成するのが望ましい（図1）。これは、知識の検索を中心にした処理と、検索された知識に対する推論を中心にした処理とを機能分割することに対応する。実際の知的情報処理分野では、必ずしも大量の知識を必要とせず手続き的処理の繰り返しですむ場合もある。そのような場合には、フロントエンドシステムだけで処理を進め、知識を必要とする場合にバックエンドの知識ベースシステムに問い合わせを

発生すればよい。

現在、そのようなバックエンドシステムとしての実用的な知識ベースシステムはまだ発表されていない。研究段階として、幾つかのアプローチが提案されている。代表的なアプローチとしては、一階述語論理の推論システムとデータベースシステムを結合させる研究および知識ベースマシンの研究がある。

関係データベースに推論システムを結合させたものは、演繹データベースと呼ばれ、各種の実現方法が提案されている。これは、知識ベースシステム構築への第一歩であると同時に、高機能データベース検索言語の提供でもある。ここで注意したいことは、演繹データベースの場合、バックエンドシステムの中で管理されるのは、知識ベースの部分集合としてのデータベースのみであるという点である（図3 a）。つまり、演繹データベースは、知識ベースの中で量が増えるのはデータベース部分であるという前提のもとで、システムを構成している。知識ベースシステムとしては、しばらくの間は各種機能の検討のために演繹データベースを使うことは有益であるが、そのままでは統一的な知識ベース管理は不可能であるため、知識ベース全体を管理するようなシステム構成に移行する必要がある。

知識ベースマシンは、知識ベース全体を統合的に処理するための専用計算機で、データベース部分と知識ベース部分を区別することなく全体を一括して管理する（図3 b）。現在は、各アプリケーションのバックエンドとして1つの知識ベースマシンを置くことを前提としているが、将来的には、アプリケーションの実行も知識ベースマシン上で行うという構成も考えられる。

筆者らは、前述したような知識表現の機械語と知識

コンパイラを具備し、高機能のパターンマッチ（単一化）のための専用ハードウェアを持つ知識ベースマシンを提案している。そのマシンでは、知識ベースは二次記憶中に体系的に格納され、I/Oアクセスのボトルネックを解消するためのアーキテクチャ上の工夫がなされている。現在のところ、筆者の知るかぎりでは、このマシンほかに知識ベースマシンとして発表されたマシンはない。今後、研究が期待される分野である。

以上集中管理を前提とした知識ベースシステムの構成方法について検討してきたが、複数のサイトに分散された知識ベースから構成する分散知識ベースシステムも重要な研究課題であろう。

5. まとめ

知識ベースシステムの特徴について、データベースとの関連を中心に述べてきた。知識ベースの研究は、まだ始まったばかりで未解決な問題も多いが、今後の発展が期待できる分野である。本稿では取り扱わなかったが、不確実な知識を扱うことのできる知識ベースシステムや、類推することのできる知識ベースシステム、さらには学習機能を備えた知識ベースシステムなどが将来実現され、人間の知的活動を強力にサポートするようになるのではないだろうか。

参考文献

- 1) Borkin, S. A.: Data Models: A Semantic Approach for Database Systems, The MIT Press, Cambridge (1980).
- 2) Codd, E. F.: A Relational Model of Data for Large Shared Data Banks, C. ACM Vol.13, No.6, (1970).

- 3) Date, C.J.: An Introduction to Database Systems, Volume II, Addison Weseley, Reading (1984).
- 4) Davis, R., King, J: An Overview of Production Systems, in Machine Intelligence, Vol.8, pp.300-332, Wiley, New York (1977).
- 5) Deering, M. F.: Architecture for AI, BYTE Vol.10, No.4, pp.193-206, (1985).
- 6) Gallaire, H., Minker, J., and Nicolas J-M.: Logic and Databases: A Deductive Approach, Computer Surveys, Vol.16, No.2, (1984).
- 7) Henschen, L. J., Naqvi, S. A.: On Compiling Queries Recursive First-Order Databases, Journal of the ACM, Vol.31, No.1, pp.47-85, (1984).
- 8) 北上, 國藤, 宮地, 古川: 論理型プログラミング言語Prologによる知識ベース管理システム, 情報処理, Vol.26, No.11(1985).
- 9) 國藤, 北上, 宮地, 古川: 知識工学の基礎と応用 [第4回], 計測と制御, Vol.24, No.6, (1985).
- 10) 小山, 田中: Definite Clause Knowledge Representation, Proc. of the Logic Programming Conference '85, pp95-106 (1985).
- 11) Minsky, M.: A Framework for Representing Knowledge, in the Psychology of Computer Vision, edited by P.H. Winston, McGraw_hill, New York (1975).
- 12) Miyachi, T., et al.: A Knowledge Assimilation Method for Logic Databases, New Generation Computing, Vol.2, Ohmsha/Springer, pp.385-404 (1984).
- 13) Mukai, K., Yasukawa, H.: Complex Indeterminates in Prolog and its Application to Discourse Models, New Generation Computing, Vol.3, Ohmsha/Springer,

pp.441-466 (1985).

- 14) Schweppe, H.: Some Comments on Sequential Disk Cache Management for Knowledge Base System, ICOT TR-041 (1984).
- 15) Ullman, J. D.: Implementation of Logical Query Language for Databases, ACM TODS, Vol.10, No.3, PP.289-321 (1985).
- 16) Wiederhold, G.: Knowledge and Database Management, IEEE Software, pp.315-330 (1985)
- 17) Yokota, H., Kunifuji, S., et al.: An Enhanced Inference Mechanism for Generating Relational Algebra Queries., Proc. of the 3rd ACM PODS, pp.229-238 (1984).
- 18) Yokota, H. Sakai, K., Itoh, H.: Deductive Database System based on Unit Resolution, Proc. of the 2nd Data Engineering (1986).
- 19) Yokota, H. Itoh, H., A Model and an Architecture for a Relational Knowledge Base, to appear in Proc. of the 13th Int. Conf. on Computer Architecture (1986).
- 20) 特集：知識工学，情報処理，Vol.26, No.12 (1985).

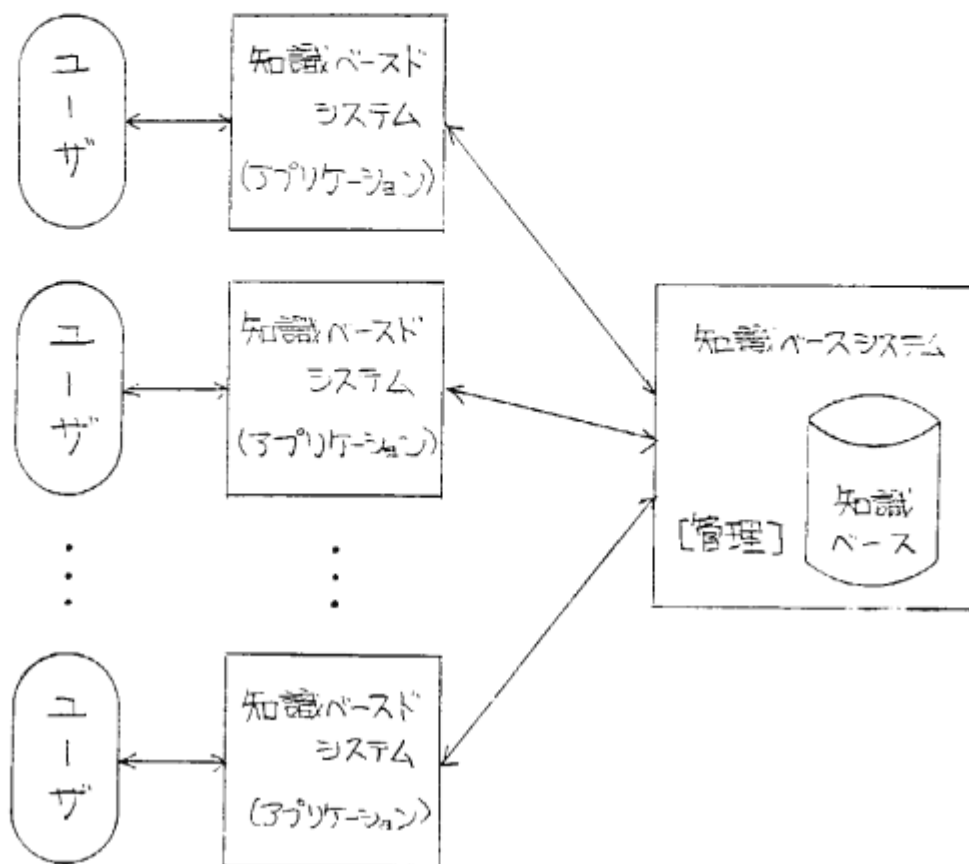


図1. 知識ベースシステムと知識ベースシステム

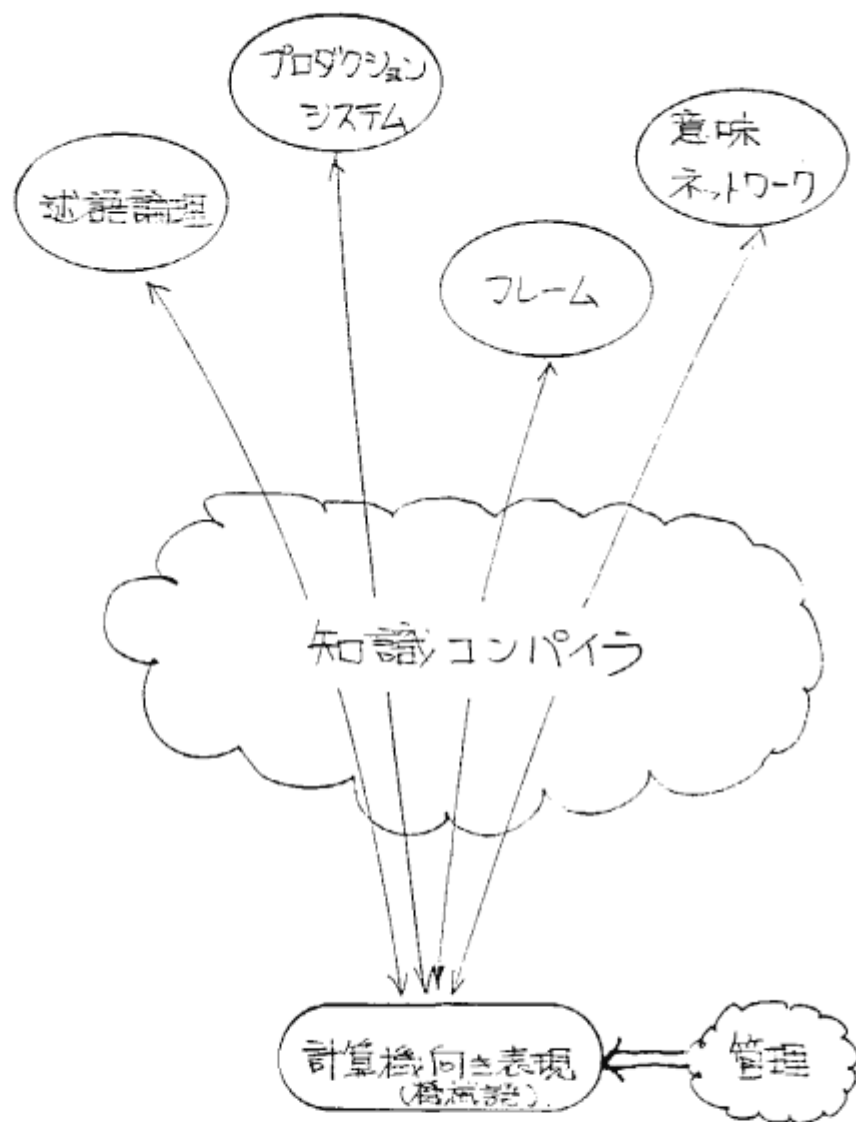


図2. 知識表現の機械語の位置付け

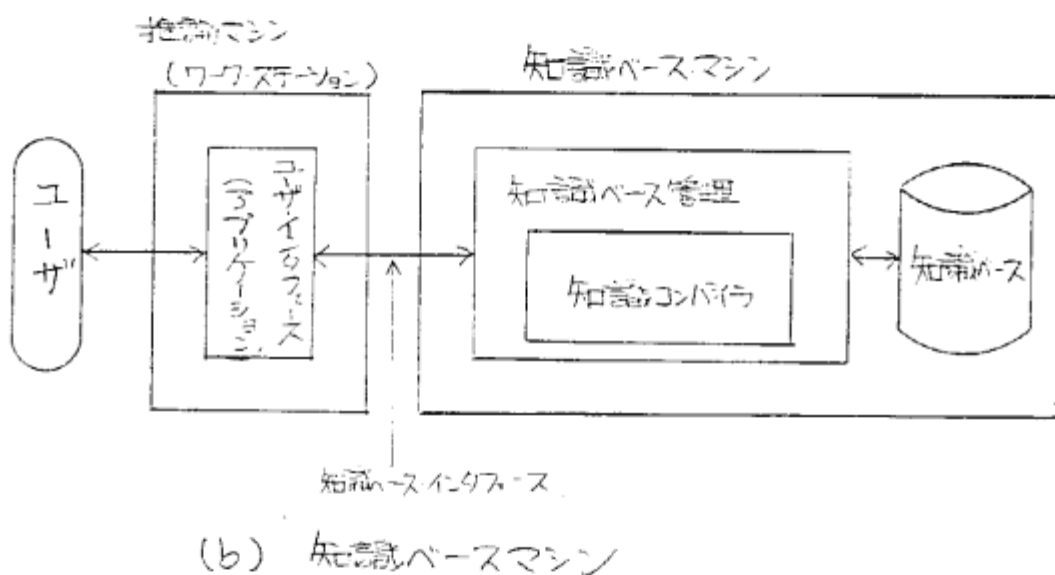
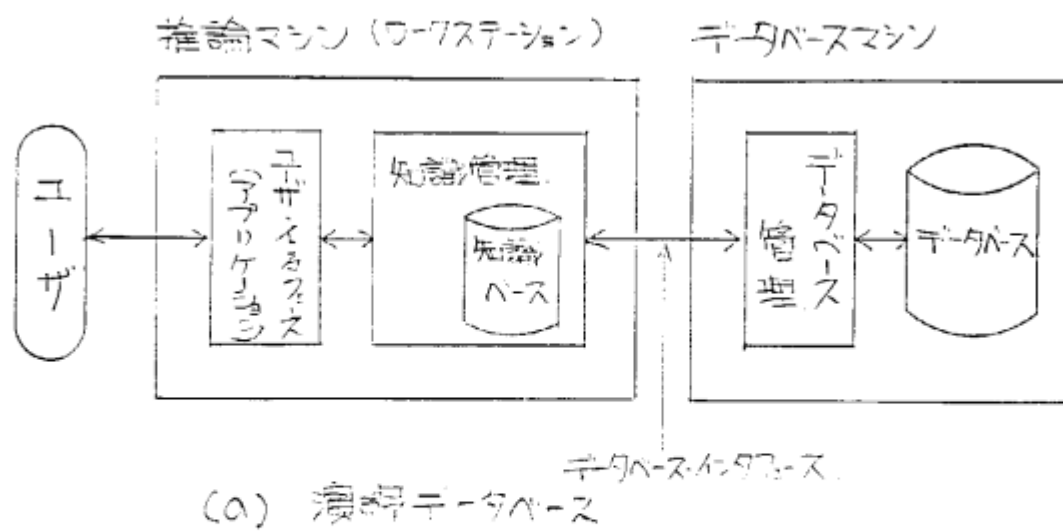


図3. 知識ベースシステム構築モデル-4.