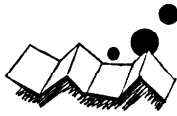


解説



知識ベースとその応用†

田中 幸吉††

1. まえがき

データベースは、ユーザの立場からみれば、現実世界の実体に関するデータの集合であるが、これらのデータがその実体を正しく表現していることを保証するために、データ意味論が重要となってくる。データの意味はある適当な表現形式によって記述される。逆に言えばある表現形式に基づく記述に対する意味づけをする仕組がデータモデルである。現在までに提案されたデータモデルの大部分は、データの構文的構成に関するものであり、現実世界におけるデータの意味を直接的に述べるができるものは今までのところ存在しないと断言してもよからう。従来のように単にスキーマを提供するようなことでは不十分であり、考えている問題全体を直接表現できるようなデータモデルが今後必要になってくる。一方、人間の知的活動によって獲得された意味を担ったデータの集積が知識である。知識の獲得・表現・処理・利用に関する研究が人工知能における重要な研究課題となってきた。知識表現の問題はその根底にある情報意味論(セマンティックス)に大きな関心がある。データベース・システムに限らず、パターン理解、自然言語処理、マン・マシン・システム等の知的情報処理システム、さらにソフトウェア工学の諸問題、意味処理のためのアーキテクチャなどにおいて、情報意味論を正面から考慮せずには扱えない問題に直面しつつある。本稿ではデータベース・システムをも含む知的情報処理システムの将来像として、知識ベース・システムとその応用例について述べる。ここで少し付言しておくが、いわゆる従来のデータベースはデータのアクセス手段が与えられて初めて利用可能であり、いわば static である。知識ベースはデータへのアクセス能力をも兼ね備え、いわば dynamic である。

2. 計算機システム上での知識のあり方

人間が考えるように計算機に考えてもらいたいという発想から、計算機上での知的活動の具体化が望まれる。これは形式化 (formalization) への挑戦ともいえるものである。人間の思考と理解といった知的活動を、どこまで、どのように形式化できるか、さらにそこに得られた形式をいかにして計算機上に具現するかという問題に直面する。しからば、この問題に対処する場合、一体どんな条件が要請されるであろうか。(i) まず知識とか問題解決法などの知的活動を抽象化したモデルの世界に写像しなければならない。このモデルの世界が semantics の世界である。抽象化を通して夾雑物を除去したからこそ物事の本質が透明な形に見えるし、またその上の論理的結果は一般性・汎用性をもってくる。(ii) つぎに semantics の世界は well-defined な形式的表現形式 (syntax) を用いて記述されねばならない。(iii) さらに抽象化と矛盾するが、具体的・個別的・特殊的でなければ現実の特定問題に対処できない。つまり semantics は pragmatics と関連づけられて現実的な意味をもってくる。

3. 知識の表現形式

プログラミング言語の意味論に対する数学的アプローチとして、帰納的関数によって表示する方法 (McCarthy, J.), プログラム束の理論に基づく方法 (Scott, D.), あるいは述語論理や第一階論理体系を用いて記述する方法 (Floyd, R. M.) などがプログラム理論において最も基本的なものとなりつつある。

プログラミング言語の意味論やコンパイラにおける意味処理の場合のように対象が明確なときでも、意味処理は十分に行えない。いわんや自然言語理解や画像・情景・音声などのパターン理解の場合にはさらに問題は困難になる。数学的枠組で表現したからと言っても、なんら問題の解決にならない場合が多々あるからである。これらの事情は情報意味論の深さを暗示し

† Knowledge Bases and Their Applications by Kokichi TANAKA (Department of Information and Computer Sciences, Faculty of Engineering Science, Osaka University).

†† 大阪大学基礎工学部情報工学科

ている。人工知能における意味情報のための知的表現の研究は、このような状況への一つの解決策を与えるものと考えられる。

3.1 知識表現の諸形式

知識を宣言的に表現するか、あるいは手続的に表現するかという事が、まず知識表現における問題点である。結局モジュラティの点で優れている宣言的表現と、インタラクションの点で優れている手続的表現とを旨く組み合わせる方法論を確立すべきであるという考え方に落ち着くようである¹⁾。つまり知識をデータとしてどのように表現し、それにどのような基本的手続を組み合わせると一つの組織にすれば、認知を実現することのできる知的構造が出来上がるのかが人工知能における知識表現の中心的課題である。この課題に対して現在までに種々の試みがなされてきた。それらを宣言的・手続的表現という尺度の上に並べて表示したものが、次の図-1である。表-1は宣言的表現と手続的表現の長所(ひいては相手方の短所)の比較表示である。宣言的表現と手続的表現がモジュール性と相互作用性を両極端とするスペクトルを表わしている。この両極端の中間的な方向を見い出そうとする試みが図-1

表-1 宣言的表現と手続的表現の比較

宣言的(記述的)知識表現形式	モジュール性に優れるが static である。 1. 事実の宣言的記述に便利 (assertion 関数)。 2. 知識の追加・修正が容易。 3. 形式的推論が容易であり、柔軟性、ひいては記憶容量の経済性にも優れる。 4. 表現構造に単純・一様性があり、理解しやすい。
手続的知識表現形式	相互作用性に優れ、dynamic であるが、柔軟性に劣る。 1. 動作とか事柄の関係を手続として記述するのに便利 (theorem 関数)。 2. 二階 (second order) の知識とか発見的知識の埋め込み容易。 3. 領域依存型の問題解決に適す。

の知識表現形式の系譜である。

PLANNER とか CONNIVER などの人工知能用言語は (i) pattern-directed invocation, (ii) automatic backtracking, (iii) pseudoparallel process, (iv) procedural specification などの能力を備えており、いわば大局的モジュール性を達成しようとする試みでもある。ACTOR モデルは知識の個々の断片的な局所的モジュール性を指向した手続的知識の表現・格納・使用のための一般的形式である。他方、プロダクション・システムは複雑な相互関連をもつ知識を、各

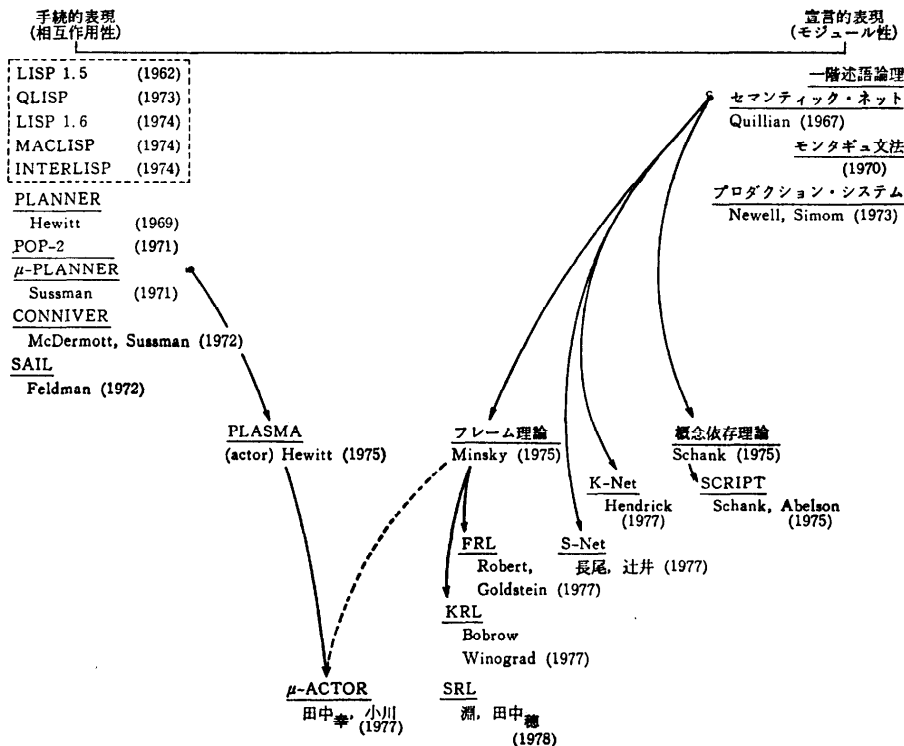


図-1 知識表現形式の系譜

- ADAM は人間である. Human (ADAM) (1)
- EVE は人間である. Human (EVE) (2)
- EVE は罪人である. Sinner (EVE) (3)
- すべての人間は誤りやすい. $\forall x(\text{Human}(x) \rightarrow \text{Fallible}(x))$ (4)

誤りやすくして罪を犯した人がいるか?
 $\exists y(\text{Fallible}(y) \wedge \text{Sinner}(y)) ?$

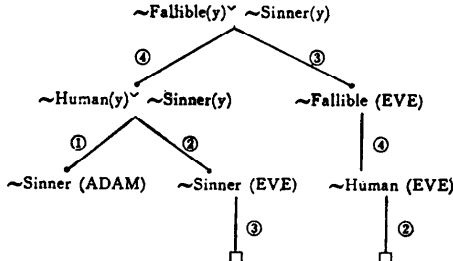


図-2 導出原理による推論の例

知識単位のモジュール性を保持しながら、計算機内部に表現する形式として用いられる。またフレーム理論は (i) 各フレームはある範例的な状況 (stereotyped situation) に関する知識を表現しているが、(ii) フレームの使い方についての情報も含み、(iii) また階層的に構成されているので、ある程度の連想・推論も可能である、等の特徴をもっている。これはいわばモジュール性を保持しながら相互作用性を指向した理論である。図-1 に示したすべての表現形式について解説する紙面がないので、本節では後述する知識ベースの応用例に関連する表現形式に限って簡単な説明を付加するととどめる。

(a) 一階述語論理を用いた表現

図-2 は導出原理 (Robinson, J. A.; 1965) による推論法の簡単な例題である。一階述語論理では、同図中の No. (1)~(4) はすべて同一レベルの知識の表現であ

1. (THASERT (ADAM HUMAN))
2. (THASERT (EVE HUMAN))
3. (THASERT (EVE SINNER))
4. (DEFPROP THEOREM 1
 (THCONS (X) (?X FALLIBLE)
 (THGOAL (?X HUMAN)))
5. (THPROG(Y)
 (THGOAL (?Y FALLIBLE) (THTFB THTRUE))
 (THGOAL (?Y SINNER))

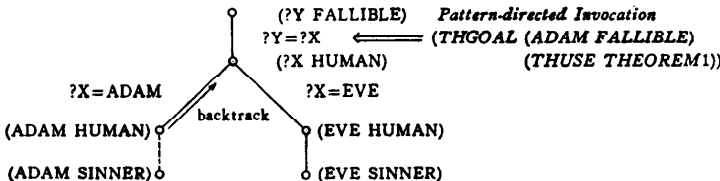


図-3 PLANNER を用いた推論の例

- P1. IF the animal has hair, THEN it is a mammal.
- P2. IF the animal gives milk, THEN it is a mammal.
- P3. IF the animal has feathers, THEN it is a bird.
- P4. IF the animal flies, and it lays eggs, THEN it is a bird.
- P5. IF the animal is a mammal, and it eats meat, THEN it is a carnivore.
- P6. IF the animal is a mammal, it has pointed teeth, it has claws, and its eyes point forward, THEN it is a carnivore.
- P7. IF the animal is a mammal, and it has hoofs, THEN it is an ungulate.
- P8. IF the animal is a mammal, and it chews cud, THEN it is an ungulate, and it is even toed.
- P9. IF the animal is a carnivore, it has a tawny color, and it has dark spots, THEN it is a cheetah.
- P10. IF the animal is a carnivore, it has a tawny color, and it has dark stripes, THEN it is a tiger.

The animal

1. has hair.
2. has pointed teeth.
3. has claws.
4. its eyes point forward.
5. has a tawny color.
6. has black stripes.

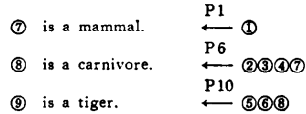


図-4 プロダクション・システムの例題 (Winston, P.H.: Artificial Intelligence pp. 145-146, Addison-Wesley (1977))

ると考えている。つまり完全なモジュール性と表現構造の一樣性に特徴があるが、非効率的である。

(b) PLANNER を用いた表現²⁾

PLANNER はパターンによる呼び出し、定理の直接呼び出し、自動バック・トラッキングが可能であるので、図-2 と同一例題を PLANNER を用いて表現すると図-3 のようになる。PLANNER のインタプリタが定理の呼び出しとバック・トラッキングの制御機構を分担している。

(c) プロダクション・システムによる表現³⁾

プロダクション・システムは Premise→Action の形式で書かれた知識 (ルール) の集合と、その上に定義された制御構造からなっている。図-4 に簡単な例題を示そう。

知識相互の関係が、プログラムを書く時点でわかっているならば、プログラムの中の必要な個所でその

知識を呼び出せばよい。しかしながら、すべての知識の相互関係を事前に決めておくことができない。これがプロダクション・システムの欠点である。MYCINでは、個々の知識間の相互関係は、診断が進むに従って診断の確かさに応じて随時決定するというプロセスを繰り返すことによって、総合的に診断を下す方法とられている。モニタによって起動されるルールの順序づけはまだ確立していないようである⁴⁾。さらに蓄積された知識を一応すべて適用するのでは非能率的である。そこでセマンティック・ネットとかフレームの概念に基づくデータ構造を用いた連想機能の導入などが必要となるであろう。

(d) フレームの概念を用いた表現⁵⁾

Minsky, M. は人間の認知過程においてフレームと呼ぶデータ構造が重要な役割を果していることを指摘した。人間が何かを認識したり行動しようとするとき、直接の対象だけでなく、それに関連した色々な構造化された情報を想起しなければ、そのものの状況を理解することはできない。このような構造化情報をフレームと呼ぶ。フレームの概念には次の3つの重要な考え方が示されている。すなわち(i)フレームはある

範例的な状況 (stereotyped situation) を表わすデータ構造である、(ii)数種類の知識が各フレームに含まれており、その中にはフレームの使い方に関する知識も含まれている、(iii)フレームは階層構造をなす知識を表現することができる。次の図-5は二つの側面と一つのトップからなる簡単なアーチをフレームを用いて表現した例である。

3.2 μ -ACTOR を用いた表現

3.1節の冒頭に述べたように、知識の相互作用の可能性を犠牲にすることなくモジュール性を具備する方法の開発が必要である。KRL¹⁴⁾ や FRL¹⁵⁾ 等が上述の問題に対処する一つのアプローチとして開発された。KRL や FRL はフレームを基礎にしており、宣言的形式の基盤の上に手続的知識を集約しようとする試みである。行動に関する知識は手続きで表わされる。この場合、データがそれを扱うための手続きと結合され、手続きがモジュール性を持つことが望ましい。その一手法として μ -ACTOR* を用いた知識の表現を筆者等が提案した⁶⁾。以下に μ -ACTOR についてもう少し解説を加えよう。

問題解決システムが大きくなると、新しい機能の追加や変更が困難になってくる。これに対処する方法として、一つのシステムを幾つかのスペシャリストと呼ばれるものに分解する方法が知られている。スペシャリストは、それぞれ独立して小さな仕事を達成するための知識と手段をもつものである。すなわち一つの問題を数個の副題に分け、それぞれの副題をスペシャリストに解決させる。このとき、各スペシャリストは完全に独立して個々の副題を解くのではなく、必要な情報を交換しながら処理を行う。このような手法により大きな問題は以下のように解かれる。まず問題を副題に分ける。ある副題についてそれを解くことができるスペシャリストが存在しなければ、その副題をさらに分解して数個の副題に分解し、解決することのできるスペシャリストを捜す。もし副題を解くことのできるスペシャリストが存在すれば、それを解決する。その際のスペシャリストはほかから必要な情報を得たり、結果の無矛盾性をチェックし、ほかのスペシャリストに情報を送ったりする。

このスペシャリストを計算機上で実現したのが μ -ACTOR である。スペシャリストは互いに独立して動作し、かつ互いの情報交換は正確に円滑に行われなければならない。ACTOR 理論 (Hewitt, 1973⁷⁾) はスペシャリスト実現のために強力な概念を与えてくれ

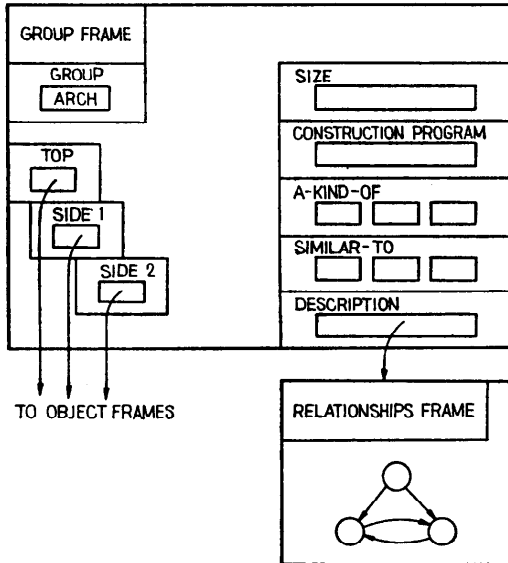


図-5 アーチのフレーム表現の例
(Winston, P.H.: Artificial Intelligence, Fig. 7.2, p. 183, Addison-Wesley (1977))

* Hewitt の了解を得て筆者等が 1977 年に μ -ACTOR という名称を用いて IJCAI-77 に提案した。ACTOR については文献 7) を参照。

た。たとえばスペシャリスト間の情報交換の手段をメッセージ転送のみに限定すると、構成最小単位をACTOR にすることによる構造の統一化、Continuation (後述) の使用等である。

μ -ACTOR を用いて知識ベースを構成した場合、各 μ -ACTOR (スペシャリスト) は知識の小さな断片に対応し、 μ -ACTOR に対応する知識に関する仕事を責任をもって処理を行う。たとえば“家”に対応する μ -ACTOR は家に関する問合せ (家の構造や色等) に答えたり、具体的な家を作ったりする (たとえば図形で示す)。後者の場合、ほかの μ -ACTOR に部分的に仕事を依頼するときもある。 μ -ACTOR により構成された知識ベースは、単に知識を記憶するだけでなく、それらの知識を扱う方法 (これも知識である) も蓄積し、知識を用いた推論も可能である。

(1) μ -ACTOR の記述法

手続的知識を表現する行動情報部と宣言的知識を表現するデータ部および μ -ACTOR に共通した制御部 (LISP を用いて記述) からなっている。

(a) 行動情報部 (SCRIPT): 図-6 に示す構造で記述される。ただし Hewitt, C. は PLASMA (PLanner-like System Modeled on Actor) を使用したが、筆者等は LISP を用いた。(i) メッセージ・パターンは現在のところ二つのタイプを使用している。接頭辞“?” のついたアトムはパターン照合用変数である。ルール中のパターン照合用変数に対応する変数には接頭辞“!” を付ける。たとえば ?CONT に対しては !CONT を用いる。?%N はメッセージの番号 %n とのみ照合する。メッセージ番号はタグとしてメッセージの区別、データの整理に使われる。“MES:” はメッセージ, “RE:” は返答を意味している。“TO: ?CONT” は継続先を表す。!CONT の内容が“NO-ONE” なら返答が不要であり, “ME” ならメッセージを転送した μ -ACTOR に返答することを依頼している。後者はサブルーチン制御と等価である。

```
script=((message-pattern1 rule11...rule1m)
      (message-pattern2 rule21...rule2n)
      .
      .
      .
      (message-patternp rulep1...rulepq)
message-pattern=(MES: ?%N content TO: ?CONT)
                //(RE: ?%N content)
rule=((P-C: programp1...programp2)
     (N-C: programq1...programqt)
     (C-E: programr1...programrn))
```

図-6 行動情報部の構造

(ii) ルールは一つの仕事を遂行するためのプログラムの集合であり、三つの部分からなる“P-C:” は前提条件を意味する。“N-C:” は付随条件を意味し、随意である。発生事象は“C-E:” で表わされ、前提条件 (および付随条件) が満たされたときにほかの μ -ACTOR にどのようなメッセージを送るかを記述する。以上の形式は次のように用いられる。メッセージの内容や μ -ACTOR の内部状態をチェック (前提条件), ほかの μ -ACTOR から必要な情報を獲得したり, 内部状態を変更し (付随条件), ほかの μ -ACTOR にメッセージを送る (発生事象)。なおこの形式は並列および並行処理を行う際に有効である。前提条件と付随条件中のプログラムは出現順に実行されなければならないが、発生事象中のプログラムは並行に遂行可能である。

使用者は、各 μ -ACTOR のデータ部を操作するために3つの基本関数 (FGET, FPUT, FREMOVE) を使うことができる。これらの関数は FRL のそれと機能的に等価であるが、引数にフレーム名 (μ -ACTOR 名) が不要であるという点が異なる。それは μ -ACTOR がそれ自身のデータ部しか扱えないという理由による。

(b) データ部 (Acquaintance): μ -ACTOR が対応する知識および行動中に得た情報に必要なものを記憶する領域であり宣言的知識に対応する。データ部の構成に関しては、ACTOR は明確な規定をしていないが、 μ -ACTOR ではフレームの概念を用いた。これにより知識の格納・使用に便利さが増した。図-7

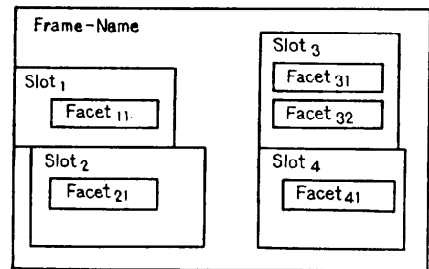


図-7 (a) フレームのスロット・アレイ表現

```
(Frame_Name
 (S1(F11(D111C111)... (D11kC11k)))
 (S3(F31(D311C311)... (D31jC31j)))
 (S4(F41(D411C411)... (D41jC41j)))
 (F32(D321C321)... (D32kC32k)))
 .....
 .....
 (S2(F21(D211C211)... (D21mC21m)))
 S: slot, : F: facet, D: datum, C: comment
```

図-7 (b) フレームの埋込型リスト表現

(a) はフレームを直観的に理解するのに便利なスロット・アレイ表現を示し、図-7(b) はインプリメンテーションする場合の埋込型リスト表現を示す (これは FRL の形式によく似ている)。

(c) μ -ACTOR の行動: μ -ACTOR はメッセージ M_i を受理したとき、 M_i に照合するメッセージ・パターンを見つけようとする。そして M_i に照合するメッセージ・パターン P_i があれば、 P_i と対をなすルールの中から順に前提条件が満足されるルール R^i を探す。 R_i 中に付随条件があるなら、付随条件中のすべてのプログラムが評価される。それらがすべて満足されたとき、発生事象中のプログラムが実行される。もし前提条件を満たすルールが存在しなければ、 M_i に照合する別のメッセージ・パターンを捜し出し、上記と同様の行動をとる。つまり μ -ACTOR は互いにメッセージを交換しながら同時並行に走る⁹⁾ メッセージやエージェント (非常に単純な行動単位) の階層構造とみなしたとき、並列処理を最も基本的な手段の一つとする分担・協力作業による問題解決を行うモデルを記述するデータ構造の一種である。

(2) μ -ACTOR の長所

結論として μ -ACTOR を使用したシステムの利点を挙げておく。

(i) 知識を手続的に埋め込むことができるため、次のことが可能である。

(a) 知識が必要なときのみ使用される、(b) 任意の詳細化レベルで μ -ACTOR を明細化できる。すなわち希望通り詳しくも簡単にでも μ -ACTOR を明細化でき、知識を必要なだけ μ -ACTOR に詰め込むことができる。

(ii) 新しく知識を追加する場合、その知識を扱うために特殊な機能が必要であっても、システム全体を変更することなく、その知識に相当する μ -ACTOR のみに、その機能を組み込んだり、さらに幾つかの μ -ACTOR を付け加えることにより実現することができる。

(iii) システムに変更を加えるときには、書き換えられた μ -ACTOR を受け渡すメッセージが今までのものと矛盾しないか、または今までのものを包含することだけに注目すれば自由に変更することができる。

(iv) 個々の μ -ACTOR の行動はメッセージ転送以外はほかの μ -ACTOR と独立しているためデバッグが容易である。すなわち送られるメッセージに対応

して正しい動作をするようにプログラムすればよい。

4. 知識ベースとその応用

知識ベースを構築し、それを利用するためには (i) 知識表現のほか (ii) 知識ベースを利用して問題解決を行う技法と (iii) 知識ベース構築のための知識獲得方法が不可欠の技術である。問題解決における推論機関 (Inference Engine) は知識表現と密接不離の関係にある。

知識工学は図-8 に示すように、知識ベースと推論機関を基本構成とする知識ベース (利用) システムに関する技術である。ある世界の知識を表現したり、その知識を使って推論することは、複雑な計算機プログラムの構成における、いわば Art—Art: The Principles or methods governing any craft or branch of learning. Art: Skilled workmanship, execution, or agency. (Feigenbaum, E. A.⁹⁾)—である。この Art の真髄は “expertise” とよばれるものである。つまり General Problem-Solver (GPS)¹⁰⁾ は高いパフォーマンスのシステム構成の基礎としては、その能力が弱すぎる。最もよい GPS は Human Problem-Solver であり、その能力はその人が専門領域としていた分野を除いては、弱くて浅いとみなされる。スペシャリストは自分の関係する専門領域における専門的知識と専門的技法とヒューリスティクスをもっている。知識工学におけるキー・プロブレムは次の文章により旨く表現されている。

The fundamental problem of understanding intelligence is not the identification of a few powerful techniques, but rather the question of how to represent large amounts of knowledge in a fashion that permits their effective use and interaction.⁹⁾

次の表-2 は知識ベースシステムの応用例 (知識工学の成果) を示す。同表の中から MYCIN を選び、そ

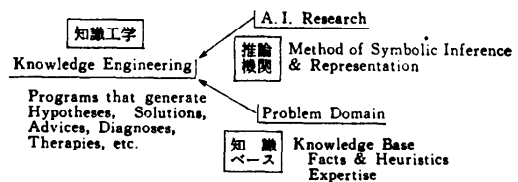


図-8 知識工学の基本構成

(阪大における Feigenbaum, E. A. 教授の講演—1979年10月25日—より)

表-2 知識ベースシステムの応用例

コンサルテーション・システム	有機化合物同定	DENDRAL (Stanford 大) META-DENDRAL (ditto) SU/P (CRYSSALIS) (ditto) MOLGEN (ditto)	化合物構造の推論システム 質量分析ルールの推論システム 蛋白質のX線結晶データ解析システム 分子遺伝学用アドバイス・システム
	技術用アドバイス・システム	SACON (Stanford 大) SU/X (ditto) KNOBS (Mitre 社)	構造体解析システム 信号解析・理解システム 航空機識別シミュレータ
	診断・治療システム	MYCIN (Stanford 大) TEIRESIAS (ditto) PUFF (ditto) MECS-AI (東大病院) CASNET (Rutger 大) EXPERT (Rutger 大) PIP (MIT)	血液伝染病・脳膜炎の診断・治療システム 大規模知識ベースの構成・保守・利用のためのメタレベル知識援用システム 肺機能診断システム 心不全診断・治療システム 線内障の診断・治療システム 甲状腺疾患・リウマチ等の診断・治療システム 腎疾患の診断システム
言語情報処理システム	質問応答システム	SHRDLU (MIT), BUILD (MIT) ELIZA (ditto) GUS (Xerox 社) SCHOLAR (MIT) STUDENT (Bobrow, 当時 MIT) Newton (MIT) Isaac (Texas 大) 京大 (長尾, 辻井) ARIMAS (阪大)	積木の世界におけるロボットの対話 精神分析用 QA システム 旅行計画用 QA システム 地理学習用 CAI 算数の問題解決システム 物理の問題解決システム 物理の問題解決システム 化学の問題解決システム 算数, 幾何の問題解決システム
	文章解析理解システム	SAM (Yale 大), TOPLE (MIT) MARGIE (Stanford 大) LINGOL (MIT) PLATON (京大) EXPLUS (ETL) MILISYJ (ditto) VISUALISER (ditto) MSSS-78 (通研)	談話理解システム 英文解析システム 英文解析システム 日本文解析システム 日本文解析システム 日本語理解システム 同上, 但グラフィックディスプレイ応答 日本語理解システム
	音声理解システム	HEARSAY-II (CMU), LUNAR (Harvard 大)	
	情報検索システム	PLANES (Illinois 大), REQUEST (IBM) RADDER (SRI) やちまた (日本 IBM), RENDEZVOUS (IBM)	関係データモデルを用いた (形式化された) データに対する QA システム
	翻訳システム	METEO (Montreal 大) XONICS MT System (Xonics 社)	天気予報 (英語→仏語) 露語, チェコ語, セルビア語→英語

のほかに筆者等の μ -ACTOR を用いた質問応答システムと物語理解システムの試用例について述べる。

4.1 知識ベースを用いた診断・治療システム (MYCIN)

MYCIN⁹⁾ はスタンフォード大学の Shortliffe, E., Buchanan, B. G., Davis, R. によって作られた血液伝染病と脳膜炎の診断と投薬・治療のシステムであり, 米国で実用の域に達したシステムとして医者の間で高く評価されている。このシステムはプロダクション・システムの考え方による知識表現・問題解決を行うプログラムである。すなわち医学知識からいくつかの規

則を抽出し, それを基にして病名を推定し, 投薬プランを作る。その規則は 3.1 節の (c) 項で述べられたように IF (Conjunctive Clause) THEN (Implication) の形式で, たとえば図-9 のようである。MYCIN はこのような規則を約 200 もっている (規則の内部表現は INTER-LISP の形式を用いている)。MYCIN の推論方法は (i) Rule-Based Inference System* の中の Consequent-Driven System である。(ii) 一種の近似的推論 (Plausible Reasoning) を行っている。すなわち (a) 各種の検査データやそれまでに下された診断結果の実績に基づくルールの評価値をフィードバックする, (b) ルールの評価値は次のようにして決められる, (1) 検査データ e に基づく診断ルール h の Measure of Belief を MB, Measure of Disbelief を MD

* Rule-Based System は Pattern-Directed Inference System の一種である。RBS はさらに Antecedent Driven System と Consequent-Driven System に分類される。META-DENDRAL は前者であり, MYCIN は後者の例である (Hayes-Roth¹¹⁾)。

RULE 092

.....

- IF: 1) THERE IS AN ORGANISM WHICH REQUIRES THRAPHY, AND
 2) CONSIDERATION HAS BEEN GIVEN TO THE POSSIBLE EXISTENCE OF ADDITIONAL ORGANISMS REQUIRING THERAPY, EVEN THOUGH THEY HAVE NOT ACTUALLY BEEN RECOVERED FROM ANY CURRENT CULTURES

THEN: DO THE FOLLOWING:

- 1) COMPILE THE LIST OF POSSIBLE THERAPIES WHICH, BASED UPON SENSITIVITY DATA, MAY BE EFFECTIVE AGAINST THE ORGANISMS REQUIRING TREATMENT, AND
 2) DETERMINE THE BEST THERAPY RECOMMENDATIONS FROM THE COMPILED LIST

OTHERWISE: INDICATE THAT THE PATIENT DOES NOT REQUIRE THERAPY

図-9 MYCIN における規則の例 (文献4) の p.152 Rule 092)

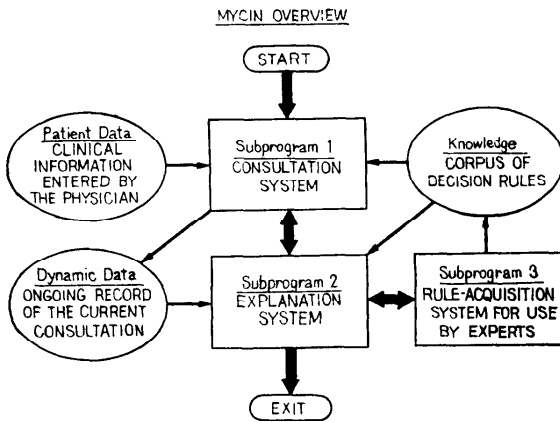


図-10 MYCIN のシステム構成図 (文献4) p. 45 の Fig. 1-1)

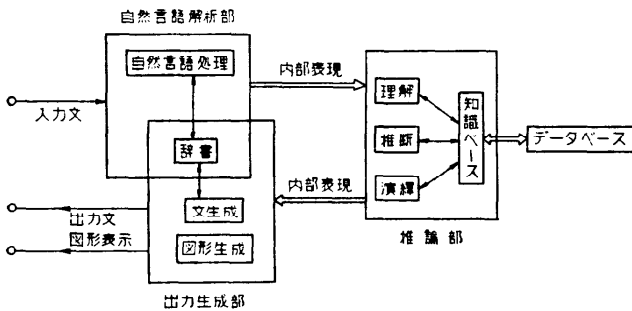


図-11 質問応答システムの構成図

とすれば Certainty Factor CF は $CF(h, e) = MB(h, e) - MD(h, e)$ で与えられる, (2) ルールの合成法則は (1) Conjunction (Disjunction) に対し, MB は $\min(\max)$, MD は $\max(\min)$ の演算を行う, (ロ) 同一のルール h を支持するデータがたとえば e_1 と e_2 の 2 つある場合 (Incrementally Acquired Evidences)*

$$MB[h, e_1 \& e_2] = \begin{cases} 0, & MD[h, e_1 \& e_2] = 1 \\ & \text{のとき} \\ MB[h, e_1] + MB[h, e_2](1 - MB[h, e_1]), & \text{その他の場合} \end{cases}$$

$$MD[h, e_1 \& e_2] = \begin{cases} 0, & MB[h, e_1 \& e_2] = 1 \\ & \text{のとき} \\ MD[h, e_1] + MD[h, e_2](1 - MD[h, e_1]), & \text{その他の場合} \end{cases}$$

で与えられる. かくして Reasoning Network 中のすべての Line-of-Reasoning の CF が決まれば, あるしきい値をもって pruning を行う. いわばこれは Generation-and-Test Procedure による Goal-oriented

なまたは Rule-Based な問題解決である. 以上のような機構をもっている MYCIN システムの構成は図-10 に示すように, 単に CONSULTATION (診断を下す) だけでなく, EXPLANATION や RULE ACQUISITION が重要な役割を受持っている.

4.2 知識ベースを用いた質問応答および物語理解システム

(1) 質問応答システム: このシステムは図-11 に示すように, 自然言語解析部, 推論部および出力生成部の三つの部分に分かれている. 自然言語

解析部は, 辞書中の統語情報, 語い情報を参照しながら主に統語解析を行い, 表層構造から深層構造に変換する. 出力生成部は, 推論部で得られた内部表現を辞書を参照しながら適格文に変換する. また知識ベースから図形に関する情報を得て図形を生成する. 推論部は, 理解, 推論, 演繹の三つの機能に分かれ, それぞれ知識ベースの内容を追加, 変更, 削除したり, 参照したりする. 理解では入力文の内容がどの知識と対応しているか, また知識と

矛盾しないかを調べる. 新しい事実が入力された場合, それを知識に追加する. 推論では入力文の情報と

* この合成法則はファジイ集合論における2つのファジイ集合 A と B の代数和に対するメンバシップ関数の演算と軌を一にしている. ちなみにメンバシップ関数の値 (グレード) が CF に対応した概念である.

- 文
- 1 *家を作りなさい
 - 2 !家はどんなものですか?
 - 3 *長方形の上に三角形を置いたものです
 - 4 !OK
 - 5 *屋根は二等辺三角形の方がよい
 - 6 !屋根は家に関係ありますか?
 - 7 *はい
 - 8 !屋根は家の三角形のことですか?
 - 9 *そうです
 - 10 !OK
 - 11 *その屋根は赤の方がよい
 - 12 !OK
 - 13 *家はタワーのようだ
 - 14 !あなたの言っている家は画面に出ている家のことですか?
 - 15 *はい
 - 16 !OK
 - 17 *その家はどんな家ですか?
 - 18 !その家は必ず二等辺三角形⁵長方形³からなっている
そしておそらく非常に高い高さである家です
 - 19 *家はどんなものですか?
 - 20 !家は必ず三角形と長方形からなっている
そして家1を具体例にしてもつものです
- } μ -ACTOR 家を定義する
- } μ -ACTOR 屋根を推論する
- } Metaphor も可能

図-12 幾何図形に関する質問応答システムの会話例

知識を比較することにより間接的に得られる情報を知識に追加する。演繹では知識から論理的方法 (Goal-oriented な定理証明法, D-グラフ推論¹²⁾など) を用いて結論を導き出す。

次に μ -ACTOR を用いた幾何図形に関する質問応答システムの試作例を紹介する¹³⁾。図-12は μ -ACTOR に基づいた幾何図形に関する質問応答システムの会話例を示している。使用者は日本語を用いてシステムと会話できる。図中, “*” から始まる文は入力であり, “!” から始まる文は出力である。システムは最初文脈として幾何図形の世界を持っており, 各種の三角形や四角形に関する知識 (専門知識ベース) を持っている。すなわち, 各幾何図形に対応する μ -ACTOR が存在している。また, 「作る」「よい」「タワー」に関する μ -ACTOR, および入力文をメッセージの形に変換する μ -ACTOR* がある。 μ -ACTOR タワーには非常に高いという特徴が与えられている。システムは入力文を知識と対応させ処理しようとするが失敗したとき (入力文を処理できる μ -ACTOR がないうち), 使用者に質問する (文 1, 2)。使用者により定義されれば μ -ACTOR (例では μ -ACTOR 家) が作られる (文 3, 4)。文 5 から 10 では, 文脈が家に関することであり, 話題が三角形の一種であることを用いて推論 (Inference) を行い, μ -ACTOR 屋根が作られ

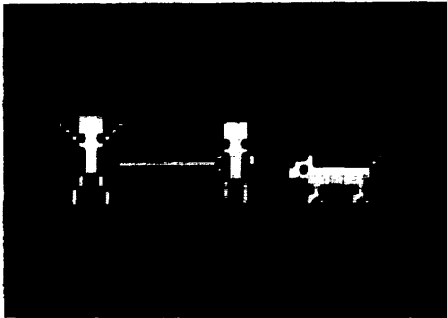
ている。推論結果は使用者により確認される。作られた μ -ACTOR は自由に使用することができる (文 11, 12)。その他比喩 (Metaphor) の処理も可能であり, 一般的な話題と具体例における話題を区別することができる (文 13 から文 20)。

(2) 物語理解システム: 物語が正しく理解されたか否かは, 複数の文によって入力された事象の理解や展開を, 物語に登場する物体 (object) の動きをシミュレーションすることによって確認される。ここでシミュレーションというのは「対象とする世界に関して入力文から得られた知識と計算機内に蓄えられている一般的常識を用いて, 時間変化とともにどのように世界が変化するかを推論すること」を意味する。ここでは推論結果をアニメーションとしてディスプレイ上に表示するシステムを試作した。

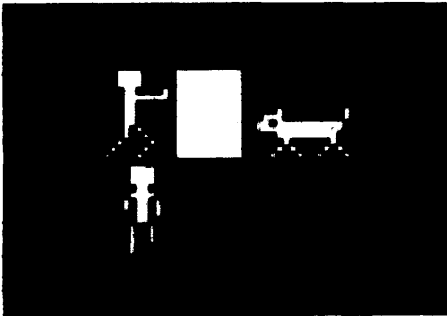
物体の動きを旨くシミュレーションするために, 物体の一つ一つに μ -ACTOR を割り当て, さらに μ -ACTOR に新しくデモンの機能を付け加えた。それは, 個々の物体の動きとほかの物体の動きの関係を扱うために必要なものである。デモンは, データ部の Pre-action スロットと Post-action スロットにルールの形 (詳しくは文献 6) の最後の論文参照) で記述される。 μ -ACTOR にメッセージが送られると, 行動情報部に従って実行する前に, Pre-action スロット中のルールがすべて評価され, 実行後 Post-action スロット中のルールがすべて評価される。たとえば「太郎は町に着けますか」という質問文に答えるためには, 常に太郎の位置を確認しておく必要がある。しかしこれは非常に困難である。というのは, 途中で「お金を忘れた」「怪我をした」などの種々の原因で行くことができなくなるかも知れないからである。この困難を解決してくれるのがデモンの働きである。データ部の Post-action スロットに「もし太郎の位置が町であるならば, μ -ACTOR 答えにメッセージを出せ」というルールを書くことによって質問に答えることができるからである。

さらに複数個の物体が物語に登場し, 個々の物体を同時に動かさなければならない場合, CPU が一つであるため, 実際個々の物体を同時に動かせない。したがって本システムでは内部項目に時刻フラッグを付ける方法を用いた。 μ -ACTOR は擬似並列処理 (詳しくは文献 8) 参照) を行うことができるので, 複数個の物体が動く場合には都合がよい。

図-13 (a) は太郎と次郎が綱引きをしている所に犬



(a)



(b)

図-13 アニメーションの一場面

- (a) 綱引きの場所に犬が近づいてきたところ、犬が嫌いな者が逃げだし、相手が勝った。
 (b) 箱を押しているところに犬が近づいてきたとき、犬が嫌いな者が逃げだしたところ。

が近づき、太郎は犬が嫌いなので犬が近づいてくのを認めて逃げ出し、次郎が万才をして勝ったことを示す(アニメーションのための図形命令発生部は、推論システムの負担を増やさないように配慮し、物語理解システムと同様に μ -ACTOR によって構成されている。なお図形生成部は図形命令を一度に 10 個まで処理でき、使用言語は N-BASIC である)。図-13 (b) は太郎と次郎と一緒に箱を押している所に犬が近づいてきた。太郎は犬が嫌いなので箱を押すのを止めて逃げて行く状況を示す。

5. むすび

人工知能に関する研究の実用的成果の一つの現われとして知識工学が最近とみに注目を浴びようになってきた。知識工学は、人工知能の研究において得られた原理と道具立てを、問題解決のための専門家の知識とか一般的常識を必要とするような困難な応用的諸問題へ利用する技法である。

知識工学における中心課題を列挙すると、1°) 領域

ごとの専門的知識ならびに常識の獲得とその適当な表現形式の研究、2°) 推論機構の研究、3°) 知識ベースの管理システムの研究、さらに、4°) 認知科学の研究、などである。人工知能を実現するための研究を進める上で、単に結果的に見て性能のよいプログラムを作ればよいというのではなく、逆に人間の知能がいかにして発達するかという過程を研究して始めて良い性能のプログラムを作成することができる。つまり“知識”の科学——認知科学(Cognitive Science)——の研究も必要である。

コミュニケーションは送受信者双方が共通の知識をもって始めて成立する。人間は言語の構造的知識、社会常識などを記憶し、それから特殊な場合をすぐ思いついたり、似かよった場合を推察して、冗長でないコミュニケーションを円滑に行っている。質問応答システムのような人間—機械系においても同じようなことが言える。つまり機械の側にも組織化された知識——知識ベース——が必要である。

さらにまた VLSI の実用化とファームウェア技術の進歩によって非ノイマン型のアーキテクチャの機械を設計し、実験することが容易になりつつある。筆者らの μ -ACTOR による知識表現と問題解決法を計算機上にインプリメントした場合、その動作応答速度を実用的速度に向上させることが一つの重要な課題となってきたが、これに対してもマイクロプログラミングとマイクロプロセッサ技術が有力な手立てとなる。筆者らは目下 ACTOR マシンの試作にとりかかっている。

参 考 文 献

- 1) Winograd, T.: Frame Representations and The Declarative/Procedural Controversy, in "Representation & Understanding—Studies in Cognitive Science" (Bobrow, D. G. & Collins, A., eds.), pp. 185-210, Academic Press, New York (1975).
- 2) Hewitt, C.: PLANNER: A Language for Proving Theorems in Robots, Proc. 1st IJCAI, pp. 295-301 (May 1969).
 Hewitt, C.: Procedural Embedding of Knowledge in PLANNER, Proc. 2nd IJCAI, pp. 167-182 (Sep. 1971).
 島田: 定理証明用言語 MICRO-PLANNER について, 情報処理, Vol. 14, No. 5, pp. 341-348 (1973).
- 3) Newell, A.: Production Systems: Models of Control Structures in "Visual Information

- Processing" (Chase, W. G., ed.), pp. 463-526, Academic Press, New York (1973).
辻井: プロダクション・システムとその応用, 情報処理, Vol. 20, No. 8, pp. 735-743 (1979).
- 4) Shortliffe, E. H.: MYCIN: A Rule-Based Computer Program for Advising Physicians Regarding Antimicrobial Therapy Selection, Report for ARPA, ONR & NIH, No. AD/A-001373, pp. 152-160, pp. 196-199, pp. 245-248 (Oct. 1974).
- 5) Minsky, M.: A Framework for Representing Knowledge in "The Psychology of Computer Vision" (Winston, P. H., ed.) pp. 211-280, McGraw-Hill, New York (1975).
- 6) Ogawa, H. and Tanaka, K.: A Structure for the Representation of Knowledge—A Proposal for Micro-ACTOR—, Proc. 5th IJCAI, pp. 248-249 (Aug. 1977).
小川, 田中: μ -actor の実現と知識表現の構造, 情報処理, Vol. 19, No. 9, pp. 846-852 (1978).
Ogawa, H. and Tanaka, K. et al.: An Active Frame for the Knowledge Representation, Proc. 6th IJCAI, pp. 668-675 (Aug. 1979).
- 7) Hewitt, C., Bishop, P. and Steiger R.: A Universal Modular Actor Formalism for Artificial Intelligence, Proc. 3rd IJCAI, pp. 235-245 (Aug. 1973).
- 8) 小川, 田中: 変数を共有する問題の擬似並列処理, 情報処理, Vol. 19, No. 9, pp. 839-845 (1979).
- 9) Feigenbaum, E. A.: The Art of Artificial Intelligence: Themes and Case Studies of Knowledge Engineering, Proc. 5th IJCAI, pp. 1014-1029 (Aug. 1973).
- 10) Ernst, G. W., Newell, A.: GPS: A Case Study in Generality and Problem Solving, ACM Monograph Series, Academic Press, Inc., New York (1969).
- 11) Hayes-Roth, F. and Waterman, D. A.: Principles of Pattern-Directed Inference Systems in "Pattern-Directed Inference Systems" (Waterman, D. A. and Hayes-Roth, F., ed.), p. 585, Academic press, Inc., New York (1978).
- 12) 小川, 北橋, 田中: 有向 AND-OR グラフを用いた形式的推論とその完全性, 信学論 (D), Vol. 60-D, No. 11, pp. 913-920 (1977).
小川, 北橋, 田中: D-グラフ推論と推論過程の最小性, 信学論 (D), Vol. 60-D, No. 11, pp. 905-912 (1977).
Ogawa, H., Tanaka, K. et al.: The Theorem Prover Using a Parallel Processing System, Proc. 6th IJCAI, pp. 665-667 (Aug. 1979).
- 13) Tanaka, K. and Ogawa, H.: Knowledge Representation Scheme for Question-Answering System and Image Interpretation System, in Research on Scientific Information Systems in Japan (Final Report of Research Project entitled "The Formation Process of Information Systems and the Organization of Scientific Information", Inose, H., ed.), pp. 200-206 (Mar. 1980).
- 14) Bobrow, D. G. and Winograd, T.: An Overview of KRL, A Knowledge Representation Language, J. Cognitive Sci., Vol. 1, No. 1, pp. 3-46 (1977).
- 15) Roberts, R. B. and Goldstein, I. P.: The FRL Manual, MIT AI Memo 409 (Sep. 1977).
(昭和55年7月7日受付)