

知識工学とは何か*

本多 庸悟**

Introduction to Knowledge Engineering / Tsunenori HONDA

Key words: knowledge engineering, artificial intelligence, expert system, production system, inference, computer science

1. はじめに

知識工学の入門的概説を、という会誌編集委員会からの依頼をいただいたが、すでに当学会誌には関連の秀れた解説がいくつかある²⁷⁾²⁸⁾³¹⁾³²⁾³⁸⁾。そこへ更にもう一つ加えるからには何か特色を出さなくてはいけないであろう。

知識工学は人工知能がその母胎であり、人工知能は計算機が誕生したから生まれたものである。これらの分野から少し離れたところに居られる方々から見れば、これらの分野にはよく訳のわからない概念や言葉が次々とウンカのごとく発生し、まるで異国語を聞くようであろう。

そこでこの解説では限られた紙数の中でのなるべく総花的に多くの概念、用語を挙げることにし、かつ全体的なスコープをイメージとして浮かび上がらせられるよう努力してみることにして、実際に主な概念の間の関連を大きめの図にすることを試みた。これは楽しみのある作業ではあったが、かなり割り切ったりあきらめたりもしなければならぬので、ここに置くのはおかしい、というような疑問を与える恐れもあり、多少の危険もはらんでいる。また不勉強のための誤りもあるかも知れない。

また後半には日ごろ感じていることなどを、知識工学や人工知能、あるいはその先の何かに託して記した。これら全体について御議論、御叱正をいただければ幸いである。

2. 知識工学とその背景

知識工学 (knowledge engineering) は、人工知能 (artificial intelligence, AI) をその親として¹⁶⁾ 1970 年代後半に具体的成果を現し始めた。人工知能という壮大で夢多いテーマのなかで、とくに知識とその使い方に焦点を絞った。かつ、ある特定の専門分野 (分子構造の決定、感染症の診断や薬の処方など) を対象として、その専門家 (expert, エキスパート) が持っている専門的知識、ノウハウ、経験則といったものをどのような形態で計算機上に表現し、利用の段階でユーザたる専門家とどのようなやりとりをすることでコンサルテーションが出来るのか、というようなことを、実際にシステム (エキスパートシステム) を構築しながら論じて来ている。

図 1 は、人工知能や計算機科学、さらにその他の関連分野や項目も含めて、その主な概念を関連づけて描いてみたものである。以下で適宜、参照されたい。

2.1 計算機と人工知能

計算機の第 1 世代は 1940~50 年代で、Wilkes らの EDSAC, Aiken ら Harvard 大学の Mark I, II, III, Manchley, Eckert に von Neumann も加わっての ENIAC, EDVAC, MIT の Whirlwind などがあった。人間の知的活動の一部である計算を機械に置きかえることに意欲を燃やした人々が、人間や生物の神経系や脳に大きな関心を抱き、計算機の知的能力を高めることに興味を持ったことは不思議ではない。プログラム内蔵方式の創案者とされる von Neumann (図 2) も早くから神経系や脳との関連を論じている¹⁾。

1956 年 6 月に McCarthy, Minsky, Shannon,

* 原稿受付 昭和 63 年 7 月 1 日

** 正 会 員 東京農工大学工学部 (小金井市中町 2-24-16)

テムを作り、GPS (General Problem Solver) と呼んだ⁶⁾。

言語理解も主要なテーマであり、記号論理の領域は発展して1965年には導出原理が出され⁵⁾、演繹的推論に路を拓いた。このような体系に基づいた論理型プログラミングシステムとしてPLANNER⁷⁾やPrologが生まれることになる。

2.2 知識工学の抬頭

1960年代後半ごろからStanford大学では、Stanford Heuristic Programming Projectが行われており、肺機能診断システムPUFF、化学構造決定システムDENDRAL¹⁶⁾、細菌感染症のための診断処方システムMYCINなどが開発された³³⁾。Feigenbaumは、知識工学(そしてそのparent scienceであるAI)がartであることを強く意識した。エキスパートのもつ知識と、知識を表現し利用して一連の推論を行ってゆく知識ベースシステムの設計を重視した。そして、GPSは高機能のシステムを作るにはぜい弱であるとして用いず、プロダクションシステム(production system)を用いた。DENDRALを作ったときの様子として、仮説や理論を実験的事実から推論するという研究者のプロセスを学ぶと共に、実験プログラムを動かすことが研究者の知的助力になったという。知識を、「状態→行動」ルール(ある状態ならば、ある行動をとる)、すなわちルールベース(rule-based)にしたことで、知識ベースの修正の柔軟性を得たとしている。

MYCINも専門の医師の相談にのる高度に専門的内容を扱うきめの細かいシステムである。例えば、動作時においてある検査の結果があるかどうか問い、数通りの回答に-1~1の信頼性係数(天気予報の降水確率みたいなもの)を付して、あとは医師の判断にまかせたりする。医師の側も、'What?'とか'Why?'という問いを発して説明を求めることが出来る。

知識ベースを基礎とするこのようなエキスパートシステムのために、その知識ベースをエキスパートが直接、構築するようなプログラムも提案されている¹⁸⁾。

階層的な構造をもつ知識、あるいは状況を作るのに適していると思われるデータ構造、フレーム(frame)が1975年に提案された¹⁴⁾。

また、やはり階層性をもつ対象で、しかも複数の処理プログラムが走るようなシステムに向くような、2次元的な広がりを持つデータ構造、黒板(blackboard)を備えさせ、黒板上の記述は適宜チェックして矛盾したものは消す、というような黒板システムも提案された²²⁾。

3. 知識工学における主な概念と手法

3.1 知識と知識ベース

前述のように、1970年代後半すぎから姿を現して来た知識工学、あるいはエキスパートシステムにおける知識とは、対象について関心のある事実群と、それらを結びつけるルール(rule, 規則)群であり、知識ベースは外からアクセスすることが出来るようにしたそれら知識の集まりである。

このような知識を、計算機向きにどのように書き表すかということは、知識表現の問題としてこれまで多くの関心が払われて来ている。本質的に我々人間の関与することであるから、基本には自然言語に基づく表現があり、それを機械的操作に持ち込む必要性から、記号表現とその処理、記号論理などにかかわる。とくに再帰的処理や記号処理にかかわるLISP、および述語論理とProlog言語については、人工知能全般を通じての大きな成果であるので、まずそれらに触れ、さらに知識表現の手法やその他の概念について述べる。

3.2 再帰処理、記号処理とLISP

McCarthyは1960年刊行の論文²⁾で大きな基礎的成果を示し、のちにこれがLISP⁴⁾として結実した。数値計算はもとより、数式そのものを扱おうとする数式処理や、記号処理などにおいて再帰的な処理(例えば、 $n! = n(n-1)!$)はしばしば現れ、FORTRANなどで手続き的に書くとは効率の悪いことが多い。これを関数的に簡便に表現するために、次の2つの表現を導入し体系化した。

① 条件の表現 $(p_1 \rightarrow e_1, p_2 \rightarrow e_2, \dots, p_n \rightarrow e_n)$,
ここで、 p_i : 述語表現, e_i : 何らかの表現であり、上式は、"if p_1 then e_1 , otherwise if p_2 then e_2, \dots "と読む。

② 関数の λ 表現

これらを基にまず、扱われる記号の取り出しや結合などに関する5つの基本的なシンボリック表現(S表現)を定め、再帰的関数など次々と定義してゆく。また、リスト(list, 文字, 記号の連なり)を単なる連なりの配列として扱うのではなく、データ間のつながりを示すポインタ(pointer)を持つようにしたデータ構造、すなわちリスト構造としたことにより2進木を扱うのに適している。

3.3 述語論理とProlog

我々が扱いたい事実は自然言語の文で書かれるが、これを記号で表現してそれが真(成り立つ)か偽(成り立たない)かを論じるのが記号論理学である。そのなかで、次の形式のものを第1階述語論理(1st

order predicate logic) という²⁶⁾。すなわち、

対象領域 D (個体変数 x, y, z, \dots の定義域) として、

$D = \{a, b, c, \dots\}$

a, b, c, \dots : 個体定数 ;

また、いくつかの個体定数から一つの個体定数への写像として、関数, f, g, \dots ;

いくつかの個体定数間に成り立つ関係について、

[限定作用素] [述語] で表す。

ここで、[限定作用素] : $\exists x, \forall x$, など、

[述語] : $p(x), q(x), \dots$ など。

ついで、次の論理記号により論理式が定義される。

\sim : 否定 (not), \wedge : 積 (連言, and),

\vee : 和 (選言, or), \rightarrow : 含意 (implication),

\leftrightarrow : 同値 (equivalence)。

ここで第1階述語論理とは述語が固定のもので、述語変数を扱うのは第2階述語論理となる。このような論理の解釈とは、変数に定数を対応させることである。Prolog はこうした論理を背景にして作られている。対象とする関係を第1階述語の範囲で宣言的に記述して入力してやると、答を探し回って出力してくれるので、推論のプログラムが手続き的でなく宣言的に書ける。

Prolog のもつ強力な機能としては、単一化 (unification, 引数が一致していない同一形式の述語において、引数を対応づけて一致させる操作) やバックトラック (backtrack, あるルートでマッチングせず失敗した場合に手続きを逆上ってやり直すこと) などがある。

なお、述語論理は (命題論理も) 真か偽の2値である。自然言語がもつようなもっと多様な表現を含むものは様相論理という。

3.4 プロダクションシステム

プロダクションシステムはエキスパートシステムに最もよく用いられる形態である。図3は、プロダクションシステムの基本的構成で、簡単のため初期の pure production system といわれるものを示す¹⁷⁾。まず、専門的知識はルール (rule, 規則) として、一つ一つが if A, then B. の形をして長期記憶に蓄えられる。またその時々状態は短期記憶に書かれる。推論部はルール部の LHS (left hand side, 条件部) を上から一つずつ取り出し、短期記憶の一つ一つとマッチングを試みる。一致した状態があると、その時の条件 A_i に対応する行動 B_i が実行され、短期記憶も必要に応じて書きかえる。

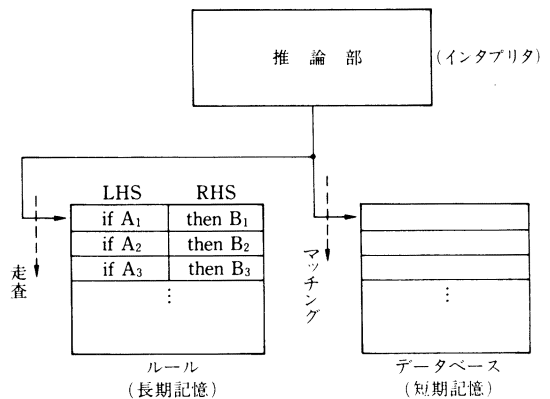


図3 プロダクションシステム (pure production system) の基本的構成

エキスパートシステム³³⁾は、さらに次のようなサブシステムを付加すればよい。

- ① エキスパートもしくは知識エンジニアから知識を取得するサブシステム。
- ② 利用時にユーザと対話するためのユーザインタフェース。
- ③ 必要に応じ、ユーザに説明を与えるための解説サブシステム。

3.5 フレーム (frame)¹⁴⁾

新しい状況に直面したときに、あるいは現在扱っている問題についての考え方を変更したいときに、人間がその記憶の中から取り出して来るであろうモデル的な基本構造である。階層的な枠組で、あるフレームの内容の細部は下位のフレームに連なる。フレームの中には特定のデータの例であるインスタンス (instance) や、データで埋められるべきスロット (slot) を持つ。

3.6 意味ネットワーク (semantic network)³⁶⁾

概念間の意味の関係を示すためのネットワークで、長期記憶の心理学的モデルといわれる。問題はネットワークの作り方で、いろいろなものが提案されている。例えば、1つの概念をクラス、性質、事例に分けて下位のノードにリンクさせるもの、2種の述語: is-a (である) 及び part-of (の一部である, 又は、を持つ) によるリンクを作るもの、などである。後者ではさらに状態 (所有者, owner) をも与えることがある (格フレーム) が、拡張されてくると推論を行ったときに必ずしも正しくない結論が出ることがあり、これを属性の継承の問題という。

3.7 スクリプト (script, 台本)

事物でなく、行為や出来事など行動に関する知識をひとかたまりにして記述したもの。列車時刻表の、い

ろいろな駅の間の運賃表のような三角表として記述することが多い。

3.8 デモン (demon, 守護神)

悪魔の意味もあるが、それでは困る。知識として記述されていないが、必要に応じて常識的に対処しなければならないときに自ら判断して処理をしてくれる割込み的機能をいう。とくにデータベースに付加や削除があったときにそれぞれ監視、察知して対応する。

3.9 メタ知識, メタ規則

知識(規則)を使うための知識(規則)。あの知識よりはこの知識の方を先に用いるべきである、などというように記述される。

3.10 ACTOR モデル¹³⁾

一つの問題をいくつかの副問題に分け、モジュール(actor)とする。各 actor は自分の問題に責任を持ち、かつ相互の間の連絡を円滑に行う。

3.11 オブジェクト指向 (object-oriented)⁴⁴⁾⁴⁵⁾

ACTOR, フレーム, Smalltalk などにおいて中心的役割を果たす概念で、データ群とそれを使用するための基本的な手続き群を一まとめにして一つのモジュールとみなし、システムをその集まりとして表現しようとする考え方である。メッセージを受けると活性化して指定の仕事をする。プロダクションシステムのようなルール指向型に対しても、大規模な知識ベースを作るには構造化、階層化が必要であり、このオブジェクト指向パラダイムとの融合がありうる。

4. 知識工学, 人工知能について 考えることも

4.1 準・エキスパートシステム

これまで述べて来たような、知識工学の産物たるエキスパートシステムを、噛み砕いて人間と対応させると次のようになるうか。

<p>[エキスパートシステム]</p> <p>知識の枠組みを持つ ……多くの知識を柔軟に扱える</p> <p>推論が速い……………頭がよく回転する</p> <p>知識ベースが作りやすい…のみ込みが速い</p> <p>聞きやすく、答えやすい…人当たりが良い</p>	<p>[人 間]</p>
---	--------------

知識工学の発展ぶりを見ていて思うのは筆者らのかつての体験である。切削条件の最適化の試みは1940年代からあったが、1960年代には筆者も所属したグループで切削条件の自動決定システムを⁸⁾、1970年代初めまでに旋削加工について図面の情報を宣言的に入

力すればすべてを自動決定してNCテープを出力する全自動のシステムMELTSを¹⁵⁾¹⁹⁾²⁰⁾、また、マシンニングセンタの工具シーケンスを決定するシステムMAD¹⁰⁾を開発した。当時の計算機環境では対話処理はまだ機能が十分でなく、バッチ処理形式をとった。明確な知識の枠組みには至らず構造的には今日でいう知識処理システムにはなっていなかったが、エキスパートの加工上のノウハウの扱いにはファイルを活用して柔軟性をもたせた。これら成果に引き合いが来たがソフトについては前例がなかったので、工業技術院が慌てたという時代である。目的としては今日のエキスパートシステムと同一で、いわば準・エキスパートシステムとでも言えようか。近年、FAそしてCIMが進展しているが、このような作業設計のプロセスの自動化にも多くの問題が残っている。知識工学の導入と更に新しい概念の発掘が望まれる。

4.2 人間のもつばらつきとの整合

知識工学の導入で、判断を伴うシステムの柔軟性を増す可能性が出て来た。しかし人間は本来、ばらつきを持つものである。フリーハンド手書き図面の読取りに関連して調べると、その精度はたかだか数%である³⁹⁾。それでも我々間のコミュニケーションには十分であるが、計算機で扱うには大変である。体育学の先生によれば、剣道で右足の踵を痛める人が多いが、フォームが悪い気がするという。有段者を対象に実験してみると有意差が出た⁴⁸⁾。画家の絵、音楽家の演奏も、好きな者には違いがわかる。絵では違いを定量的に算出できる可能性がある⁴⁹⁾。図面のばらつきについては、その構成に関するルールを利用することで自動読取りの可能性が高まる⁴⁶⁾。生産技術では、ある仕様を満たす解がしばしば一意でないことも、難しさを大きくしている。

5. お わ り に

知識、知能と言っても、まだ扱っているのは素朴なレベルの内容である。より多様なものが、より「統合」的に扱われねばならない。そもそも「知」だけでは限界がある。知の上に「情」があるとかねてから考えていたら⁴⁷⁾、不勉強にも中京大・戸田教授が綿密に論じておられるのを知り⁴²⁾、共鳴するところが多い。私の好きなHoffnungの漫画¹¹⁾に、例えば‘Listz’と題して、ピアニストが馬のようにあばれるピアノを鞭で扱っているのがある。これなども情と知の融合なくして理解できない。話が飛躍気味かも知れないが、このようなことは生産技術や生産工学にあながち無縁の話ではない。生産とは、そもそも我々人間のためのも

のを作る, 人間的な行為だからである。

参 考 文 献

- 1) J. von Neumann: The Computer and the Brain, Yale Univ. Press, (1958).
- 2) J. McCarthy: Recursive Functions of Symbolic Expressions and Their Computation by Machine, Part I, Comm. ACM, 3, (1960).
- 3) M. Minsky: Steps toward Artificial Intelligence, Proc. the IRE, Jan. 1961, (1961) 8.
- 4) J. McCarthy et al.: LISP 1.5 Programmers' Manual, The MIT Press, (1962).
- 5) J. A. Robinson: A Machine-oriented Logic Based on the Resolution Principle, J. ACM, 12, (1965) 23.
- 6) G. W. Ernst and A. Newell: GPS: A Case Study in Generality and Problem Solving, Academic Press, (1969).
- 7) C. Hewitt: PLANNER: A Language for Proving Theorems in Robotics, Proc. 1st Int. Joint Conf. on Artificial Intelligence (IJCAI-69), (1969) 295.
- 8) 竹山秀彦, 本多庸悟, 関口 博, 高田孝次, 井上久仁子: 数値制御自動プログラミングの研究 (旋削 [長手・荒削り] における切削条件の自動決定), 日本機械学会論文集, 36, 292 (1970) 2155.
- 9) A. Newell and H. A. Simon: Human Problem Solving, Prentice-Hall, (1972).
- 10) 井上久仁子, 本多庸悟, 竹山秀彦, 佐藤 真, 友森健雄, 鈴木 宏, 鈴木 博: マシニングセンタ加工シーケンスの自動決定システム, 日本機械学会誌, 78, 675 (1972) 130.
- 11) Von G. Hoffnung: Hoffnung's Potpourri Cartoons, Deutscher Taschenbuch Verlag, (1973).
- 12) T. ヴィノグラード (淵 一博, 田村浩一郎, 白井良明訳): 言語理解の構造, 産業図書 (1975). (原著は 1973)
- 13) C. Hewitt et al.: A Universal Modular Actor Formalism for Artificial Intelligence, Proc. 3rd IJCAI, (1973) 235.
- 14) M. ミンスキー: 知識を表現するための枠組, P. H. ウィンストン (編), コンピュータービジョンの心理 (白井良明, 杉原厚吉訳), (1979) 238~332.
- 15) 竹山秀彦, 井上久仁子, 本多庸悟, 関口 博, 高田孝次: 旋削加工用ソフトウェア・MELTS~パートプログラミング解説書~, 機械技術研究所資料, 66 (1976).
- 16) E. A. Feigenbaum: The Art of Artificial Intelligence: I. Themes and Case Studies of Knowledge Engineering, Proc. IJCAI-77, (1977) 1014.
- 17) R. Davis and J. King: An Overview of Production System, Machine Intelligence, 8, (1977) 300.
- 18) M. Davis: Interactive Transfer of Expertise: Acquisition of New Inference Rules, Proc. 5th IJCAI, (1977) 321.
- 19) 竹山秀彦, 井上久仁子, 本多庸悟, 関口 博, 高田孝次: 旋削加工用ソフトウェア・MELTS~データ・ファイル解説書~, 機械技術研究所資料, 67 (1977).
- 20) 井上久仁子, 本多庸悟, 関口 博, 高田孝次, 竹山秀彦: 同上~技術処理解説書~, 同上, 68 (1978).
- 21) 長尾 真, 辻井潤一: 知識の記述とデータベース, 計測と制御, 18, 1 (1979) 71.
- 22) L. D. Erman et al.: The Hearsay-II Speech Understanding System: Integrating Knowledge to solve Uncertainty, Computing Surveys, 12, 2, (1980).
- 23) 竹山秀彦, 井上久仁子, 本多庸悟, 関口 博, 高田孝次: 自動加工ソフトウェア, 日刊工業新聞社 (1981).
- 24) 白井良明, 辻井潤一: 人工知能, 岩波書店 (1982).
- 25) E. Rich: Artificial Intelligence, McGraw-Hill, (1983).
- 26) 長尾 真: 言語工学, 昭見堂 (1983).
- 27) 小山昭夫: 知識工学, 精密機械, 49, 1 (1983) 69.
- 28) 岩田一明: 人工知能とCAD/CAM, 精密機械, 49, 10 (1983) 1448.
- 29) 淵 一博 (編著): 認知科学への招待, 日本放送出版協会 (1983).
- 30) 田中幸吉 (編): 知識工学, 朝倉書店 (1984).
- 31) 大須賀節雄: 述語論理のCADへの応用 (その1), 精密機械, 50, 6 (1984) 1011.
- 32) 大須賀節雄: 同上 (その2), 同上, 50, 7 (1984) 1159.
- 33) P. Harmon and D. King: Expert Systems, John Wiley & Sons, (1985).
- 34) 中島秀之: 知識表現と Prolog/KR, 産業図書 (1985).
- 35) 大須賀節雄: 次世代CAD/CAMのための知識処理の応用, マグロウヒルブック (1985).
- 36) 岡本敏雄: セマンティック・ネットワーク・システム, 情報処理, 26, 2 (1985) 1504.
- 37) 戸田正直, 阿部純一, 桃内佳雄, 住往彰文: 認知科学入門——「知」の構造へのアプローチ, サイエンス社 (1986).
- 38) 諏訪 基: 知識工学への期待, 精密工学会誌, 52, 5 (1986) 783.
- 39) 本多庸悟, 金子俊一, 鈴木明子, 五十嵐 智, 北嶋克寛: 手書き機械図面の特性と構造, 情報処理学会第32回 (61年度前期) 全国大会予稿集, 3 N-6 (1986) 1375.
- 40) K. S. Fu et al.: Robotics, McGraw-Hill, (1987). (本多庸悟, 遠山茂樹, 古屋伸幸, 金子俊一訳: ロボティックス, 日刊工業新聞社, 近刊)
- 41) 精密工学会「設計論とCAD」専門委員会: D & C—News Letter, 12, (1987).
- 42) 戸田正直: 人工知能と認知心理学, 電子情報通信学会誌, 70, 9 (1987) 909.
- 43) 上野晴樹: エキスパート・システム概論, 情報処理, 28, 2 (1987) 147.
- 44) 竹内郁雄: オブジェクト指向とほかのプログラミングパラダイムの融合, 情報処理, 29, 4 (1988) 344.
- 45) 米沢明憲: オブジェクト指向計算の現状と展望, 情報処理, 29, 4 (1988) 290.
- 46) 本多庸悟, 金子俊一, 高橋良之, 渡辺 顕: フリーハンド手書き図面の認識システム, 情報処理学会第36回 (63年度前期) 全国大会予稿集, 7 W-3 (1988) 1885.
- 47) 本多庸悟: 新しいシステムと自律統合, 日本工業技術振興協会<パネル討論会>資料 (1988).
- 48) 磯崎芳史, 百鬼史訓, 本多庸悟: 剣道における踏み動作と右足踵部障害との関連について, 武道学研究, 21, 1 (掲載待ち) (1988).
- 49) 宮原昌克: 画像の特徴の抽出と表現, 東京農工大学工学部数理情報工学科62年度卒業論文 (1988).