知識工学と第5世代コンピュータ

古川 康一・渕 一博

1. 知識科学から知識工学へ

いま，コンピュータを利用している多くの分野で，知識工学に多大な関心が寄せられている．それは，電子工学，化学，遺伝子工学，原子力工学，資源工学などの工学の諸分野をはじめとして，医療，教育，オフィス・システムなどの分野にまでおよんでいる．そして，それらの個々の分野で，計算機科学を離れて，それぞれの固有科学への接近を強めてきている．

そのため，知識工学とは何かを論ずることは困難であり，さらに一定の枠組をはめ込むことは，かえって各分野での知識工学の発展を阻害することにもなりかねないが，ここでは，知識工学の誕生の経緯をふまえて，それがいかなるものであるかを概観してみたい．

“知識工学”は，1977年の第5回人工知能国際会議での E. A. Feigenbaum の招待講演に始まったといえよう[1]．その講演で Feigenbaum は，彼の研究グループで行なっていた実用人工知能研究の成果をとりまとめて発表し，彼らのアプローチに対して知識工学という名前を与えた．

彼らの知識工学のねらいは，それまでの人工知能研究を脱皮して，人工知能の研究成果を実際に役に立つようなシステムの実現に利用することであった．すなわち，“知識科学”から“知識工学”への転換といってもよいであろう．あるいは，科学としての人工知能研究から応用人工知能への発展といったほうがより適切かもしれない．

知識工学以前の人工知能研究の主流は，汎用問題解決機の仕組みをさぐる研究と，知識の諸側面をどのようにプログラム化するか，その表現形式をどうするか，すなわち知識表現の研究，の2つであった．特に，問題解決について考えてみるとそこでの主眼は，汎用の推論方式を考え出すことに多くの努力が払われた．問題解決機能は，人工知能の中でも最も重要な機能であることはいうまでもない．そして，“推論能力”は，問題解決にとって不可決である．推論能力は，ちょうど算数の応用問題を解くような能力であり，そのメカニズムは人工知能のエッセンスであるともいえる．この種の問題を対象とした人工知能の研究は，コンピュータの出現とともに始まったといえよう．人工知能研究にとってコンピュータが果たした役割は大変大きい．人工知能研究の成果を目に見える形で示してくれるのは，コンピュータ上に動くプログラムをおいてほかはない．その結果，計算機科学の一分野として人工知能研究が大きく発展をとげてきたわけである．

さて，ここで初期の頃の人工知能研究をふりかってみると，まず，定理証明プログラム，チェス・プログラム，幾何の証明プログラムなどの知的水準の高いプログラムがあいついで開発され

ふるかわ こういち，ふち かずひろ
(財)新世代コンピュータ技術開発機構

た．それから，問題解決手法の一般化への努力が始まり，事実や定理などを与えると自動的に解を導く汎用問題解決機の研究開発がなされた．

しかし，このようなアプローチには大きな障害のあることが次第に明らかにされた．それは，解決を見つけ出すときに起こる探索空間の“組合せ的爆発”とよばれる現象である．いわゆるしらみつぶしの方法は，調べる範囲が指数関数的に増大する問題に対してはまったく歯が立たない．そして，世の中で実際に役立つような問題のうちほとんどのものはこの性質をもっている．

そのような反省から生まれたものが知識工学である．問題解決には推論能力の他に，知識が大切な役割を果たすことは容易に考えられる．知識工学のアプローチは，問題解決能力を向上させるために，知識の比重を高めようということである．推論能力と知識のバランスが大切であり，そのどちらか一方に片寄っていては十分な問題解決能力を獲得することはできないが，知識工学が生まれたときの状況は，あまりにも知識が軽視されていたので，知識工学のスローガンとして，“推論よりも知識へ”ということが標榜されたわけである．問題解決能力は，推論の軸と知識の軸から成る平面を考えると，図1のように，その平面の面積に比例すると考えるのが自然であろう．この平面で，広い面積を獲得する種々の試みがなされてきた．知識の軸上の値を大きくするには，多くの，質のよい知識の扱いが必要となる．例として，通勤の手段を考えてみよう．勤務先が変わったり，あるいは引っ越しをしたとき，通勤経路を決めるのに何をするであろうか．まず，データを集めることから始めるであろう．交通手段が一通りならそれ以上考えることはないが，いく通りもある場合には，その中から最適なパスを選ぶことが問題となる．最適なものの評価規準は，時間，コストの他に，座れるかどうかなども重要である．さらに，そのパスの信頼性，時間のバラツキなども判断材料となる．そして，そのようなパスを一度決

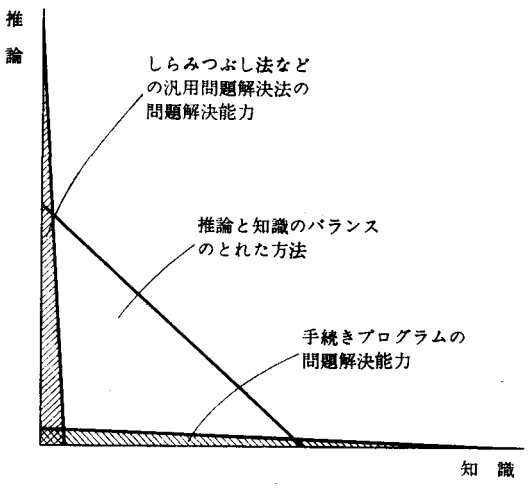

図 1 推論-知識平面と問題解決能力

めたら，毎日の通勤は，その決定にしたがって行動すればよい．毎日同じ問題を解く必要はない．そして，その手続き自身を，経験の積み重ねによって，さらに改良していくことができる．電車の乗り換え時などに，その道をいちいち探すことはないし，電車を確認することもない．それらはすべて習慣化された知識として覚えこまれているわけである．このような知識がいかに大切であり，問題解決(毎日の通勤も一種の問題解決である！)にとっていかに重要な働きをしているかは，たとえば電車が故障してふだんの経路が使えなくなったときによくわかる．そのような異常事態では，通勤経路をさがすという問題を初めから解き直し，他の解を求めることが必要となる．運よく他の解が見つかったとしても，それにしたがって行動するとき，それはいつものとおりスムーズにことが運ぶというわけにはいかない．道をさがしたり，バスの停留所をさがしたりといった小さな問題を解決しながら行動することが必要となる．

この例からもわかるように，習慣化された知識は問題解決の効率をいちじるしく高めるが，それで処理しきれないような異常時の対処も大きな問題である．完全に習慣化された知識に相当するのが手続きプログラムである．そして，その中には高度な推論はほとんど要しない．頭を使うことと

いえば単に習慣化された知識をその場その場で引き出すことぐらいである．それは手続きの実行に相当している．そのほかに必要なことは，異常事態が発生していないことを確認することぐらいであろう．これも，実は常にそのことに注意を払っていることは必要ではない．

手続きプログラムは，高効率ではあるが，それが扱うことのできる問題は非常に限られる．上で述べた異常事態に対処できないことはむろんのこと，電車の発着ホームの変更などの環境のちょっとした変化にも追随できない．そして，このように応用のきかない硬直化した“知識”は，ものの役に立たず，知識とよぶのにふさわしくない．

手続きプログラムを柔軟にする一方法に，プロダクション・システムとよばれる方法がある[2]．プロダクション・システムは図2のように，推論部，知識部，行動部の3つの部分系から成る．プロダクション・システムが手続きプログラムと異なるのは，手続きがルールの形で細分化されて知識部に格納されている点である．ルールは，

IF ～ THEN ～

の形をしている．IF の部分は，そのルールを起動するための条件が書かれており，THEN の部分には，そのときに何を行なうかが書かれている．たとえば，駅のホームの例では，

IF 電車の発駅のホームが変更されていてかつ これからの電車に乗るときは，
THEN 電車の発駅のホームを調べてからホームに行く．

といったルールになる．

ある状況で，どのルールが適用できるのかを調べ，実際にそのルールの実行を行なうのが推論部である．人間でいえば，状況を判断して次にどのような行動をすればよいかを決めることに当たる．行動部では，実際にロボットを動かしたり，あるいはディスプレイ画面上に表示したりする．

プロダクション・システムの個々の知識は，各状況に専用の知識であるが，状況を判断する部分

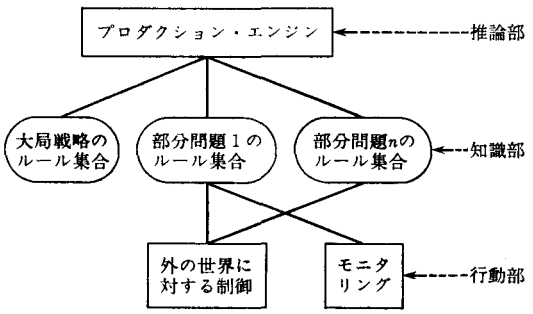

図 2 プロダクション・システムの構成

を推論部にもたせることによって，システム全体としては非常に柔軟になる．このことを手続きプログラムで実現しようとしたら，プログラムの中味の大部分が判定と制御の移行で占められ，とても実用にならないであろう．

プロダクション・システムの各ルールが知識らしく見えるもう1つの理由は，それが断片的であり，そのため人間にとっても理解が容易だからである．

ところが，プロダクション・システムだけでは，予測もできないような異常事態に対処することはできない．この対処には，問題自身の本質的な理解と問題解決能力が必要となる．そこで再び汎用の問題解決機が欲しくなるが，組合せ的爆発は防ぎたい．このような問題を扱う方法として注目されているのが，対象の構造を反映した知識表現の枠組と，それにもとづく問題解決法である．対象の構造を反映させてプログラムを作る方法は，システム・シミュレーションではよく行なわれるが，それを問題解決にも応用しようというわけである．問題解決をシステム全体を対象として行なおうとすると，たちどころに組合せ的爆発の壁に直面するが，システムの構成要素ごとに問題解決機能を分散させる．そうすることによって，各問題解決機が解決すべき問題は小さくなり，実用に耐える時間で解を得ることが可能となる．

この種のアプローチとして，Minsky の Frame の考え方が有名である[3]．Frame では，物事を表わすのに標準的な“物”とそれからのずれの

2つを用いる．そして，各成分特有の機能（すなわち，ずれ）の一部として，その物に関連したある種の問題を解決するためのプログラムをつけることもできるようになっている．これが局部的な問題解決機である．システム全体の問題を解決するには，各問題解決機の活動をうまく制御し，その間で起り得る矛盾を排除していかなければならない．これは実はこれからの研究課題であり，いまだ基礎研究の段階である．

2. 第5世代コンピュータ・プロジェクトの概観

第5世代コンピュータは，知識情報処理技術の確立を第1の目標としてかかげている．知識情報処理技術とは何であるかを明確にしなければ，その内容を推し測ることは困難かもしれないが，われわれは，それを知識工学を中核技術とした幅広い関連技術ととらえている．たとえば，知識工学に適したプログラミング言語やシステムの開発などもその中に含めて考えている．さらに，人間とのコミュニケーションを円滑にするための自然言語や図形・画像を介した会話技術などもその中に含めて考えている．

第2の目標は，ソフトウェアの諸問題を根本的に解決することである．ソフトウェアの生産性は，ハードウェアのそれに比べて，その向上率が小さく，そのためシステムのコストはその比重がソフトウェアに大きく片寄ってきている．この問題を，プログラムの部品化，モジュール化技術によって解決するのがそのねらいである．

この2つの目標を達成するための不可欠な土台としてわれわれが選択したのがPrologをひな形とする論理プログラミング言語である[4]．それは，知識情報処理にしても，プログラムのモジュール化技術にしても，必要とするのは強力なリスト処理機能であり，柔軟なデータ構造操作機能だからである．そしてこれらの機能をもつ言語としてわれわれはLispをスキップしてPrologを選択した．この選択の技術的根拠をここで詳細に述べる余裕はないが，要約すると次の4点になる．

(1) パターンの一致による手続き呼出しの機能．
この機能により，ある時点で呼出される手続きを複数個用意でき，非決定的な選択点を含んだ試行錯誤プログラムが容易に実現できる．

(2)パターンの一致によるパラメータの受け渡し．
パターンは，一般のリスト構造で表わされる複雑なものも許され，手続きを呼出すほうと呼出されるほうのどちらに変数や構造が現われてもよい．すなわち，パラメータは呼出す側と呼出される側で対等に扱われる．これにより，強力なリスト処理機能が実現される．

(3)変数を含んだパラメータが許されていること．
変数を含んだパラメータの効用は，無限構造の実現などに用いられ，Prologにルーチン機能を与えるための拡張で本質的な役割を果している[6]．また，この性質を利用したProlog特有の多くのプログラム作成技術が開発されている．DCGとよばれる構文解析プログラムがその代表例である．

(4) データベースと手続きの概念の統合．
Lispではデータベースの実現は手続きの実現と別の仕組みで行ない，そのための命令(属性リストとよばれているアトムに付随した特別のリストを操作する命令）がいくつか用意されている．一方，Prologでは，データは手続き本体が常に“true”である手続きとして定義されている．この機能により，データベースの規模が小さい範囲では，関係データベースもPrologの一部に含んでいるといえる．

以上の技術的理由のほかに，Prologあるいは論理プログラミング言語の囲りでの研究開発が，いかに速く進展し，しかもその範囲をいかに急速に拡大しつつあるかにも注目すべきであろう．

第5世代コンピュータ・プロジェクトでは，Prologを拡張した新たなプログラミング言語の開発と，その言語の専用コンピュータの開発をめ

ざしている．そして，そのコンピュータが，実は1990年代の汎用コンピュータになるであろうと考えている．それは，その言語が柔軟性に豊んでおり，柔軟なものこそが文字どおり“汎用”の名にふさわしいと考えているからである．

第5世代コンピュータ・プロジェクトの全体像を描くためには，ハードウェアの計画にもふれることが必要である．その特徴は，超並列コンピュータ・アーキテクチャと VLSI 技術の 2つである．超並列コンピュータ・アーキテクチャは，性能の飛躍的な向上をはかるために残された数少ない有力な道の1つであるが，これまでは，主にプログラミング言語との整合性の悪さが原因で，汎用コンピュータとしてはあまりメリットが出ないと考えられてきた．われわれが論理プログラミング言語を選択した理由の1つは，それがこの問題を解決してくれると考えたからである．R. Kowalski の言葉を借りると，論理プログラミング言語は，知識情報処理と超並列コンピュータ・アーキテクチャをつなぐ missing link である．これらの全体の関係は，図3に示すとおりである．

3. 第5世代コンピュータの知識工学での役割

知識工学は，これからの発展が大いに期待されているが，現在のところいまだ研究室内の実験システムの域を出ていない．Feigenbaum はそれを産業レベルのシステムにするためにはソフトウェアとしての品質の向上と十分な（既存のLispマシンの100倍程度の）処理速度の専用ハードウェアの開発が必要不可欠であることを指摘した．

われわれは，第5世代コンピュータがこの条件を満たす優れた道具となりうると考えている．第1に，超並列推論コンピュータが完成すれば，少なくとも処理速度に関しては十分である．本格的な超並列推論コンピュータが開発されるまでにはまだかなりの時間を要するので，それまでのあいだ何とかしたい．われわれが本プロジェクトの前

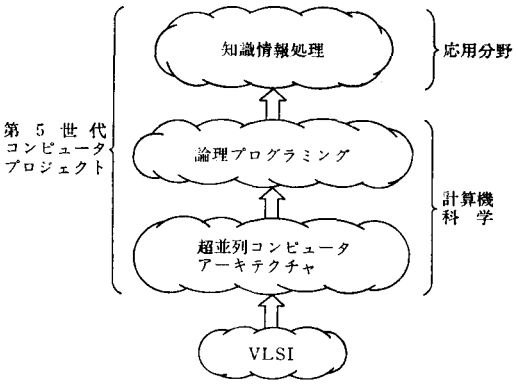

図3 第5世代コンピュータ・プロジェクトの概念図

期3年で開発をめざしているものに，逐次型の Prolog マシンがある．このマシンは，超並列推論コンピュータに比べるとその性能は2ケタ程度落ちるが，それでも現在の Lisp マシンよりも性能のよいものになると考えている．

いずれにせよ，処理速度については，現行の技術体系の中でも素子技術の発達によって数年のうちに1桁以上の性能向上は十分に望めるわけである．問題は，第2の「ソフトウェアとしての品質の向上」をどのように達成するかである．これはそれほど単純な問題ではない．むしろ，ソフトウェア工学の最大の問題であり，そこでも本質的な解決を見ていない困難な問題である．

ところで第5世代コンピュータ・プロジェクトでは，目標の1つとしてソフトウェアの諸問題を根本的に解決することをかかげていることは，すでに述べたとおりである．そしてわれわれは，その目標を達成するために Prolog をひな型とする論理プログラミング言語をプロジェクトの中核に据えることとした．知識工学の研究がこれまでに明らかにした事実は，ソフトウェアとして使える道具にはプロダクション・システムがあり，今後の発展を考えるとシミュレーション言語流に対象世界をモデル化し，問題解決機能をシステムの各構成要素に分散する方法の研究開発と，そのためのプログラミング・システムが必要となるということである．さらにそれらのシステムが概念的

に整理された形で実現され，プログラミングの効率化が十分にはかられることも大切な点である．

これらの技術的要求に，論理プログラミング言語が応えられるかどうかが次の問題である．Prolog を使ってこれまでいろいろな実験を行なってきたが，その経験に照らしていうと，見通しは明るいといえよう．溝口ら[7]は，Lisp と Prolog によるエクスパート・システムを開発し，その比較を行なっているが，彼らの結果は，プログラム・コードの量にして，Prolog が Lisp の約1/5という結果であった．コードの量にほぼ比例した開発時間を要するので，ソフトウェアの生産性が5倍向上したことになる．この事実は，実はそれ以上の意味をもっている．1つは，アイデアの実証が即座にできることである．従来は，よいアイデアがあってもそれをプログラム化するのが大変困難で，研究の進展がそこでとどこおってしまうことがよくあったが，プログラムの作成時間が短縮することによって，その障壁がとり払われたことになる．

Prolog によるプログラム作りがなぜこのように能率がよいか，その理由を一言でいえば，それが人間の思考によりマッチしているからであろう．プログラミング言語というと何か特殊な暗号のかたまりみたいに思っている人もいるかと思うが，実は，これもれっきとした言語の一種であり，その言語を十分に習得すると，その言語自身で物事を考えることができるようになる．そして，それは，アセンブラ言語でも，Fortran でも Lisp でも Prolog でも同じである．ただ違う点は，それらの言語の違いによって，どんなことが考えやすいか，すなわち得意とする対象分野が異なってくることである．そして，計算機からより離れた抽象度の高い言語ほど，人間の思考にマッチしたものとなる．その意味で，Prolog は他のどの言語よりも人間の言葉に近いといえよう．

論理プログラミング言語で，分散問題解決方式をどのように実現するかが，残された大きな問題である．分散問題解決方式は，“対象指向プログラミング”とよばれているプログラミング法に深く依存している．それは計算の“対象”を中心にして物事を考え，プログラムはその対象に対する各種のオペレーションであるとする方法である．分散問題解決でいえば，問題を解決すべきシステムの各構成要素が“対象”であり，それに付随する各問題解決機はその対象に関するオペレーションの一種と考えられる．このような計算法では，対象間の通信機能が重要な役割を果たす．複数の人で共同作業をするときに非常に緊密なコミュニケーションが必要となるのと同じことである．

論理プログラミング言語のうえでこのような“対象指向プログラミング機能”を実現する研究が，最近筆者のグループなどで，積極的に展開されており，その基本的な方式が明らかにされつつある[8]．

4. おわりに

知識情報処理技術は，コンピュータの新しい応用分野を切り拓く大きな可能性を秘めている．そして，第5世代コンピュータ・プロジェクトは，その目標の1つとしてそれを現実のものとすることをねらっている．それは，いくつか考え得るアプローチの1つであるが，われわれは，それが最も可能性の高い，技術的に質のよいものであると考えている．

われわれがこのアプローチを採っていくに当り，今後解決していかなければならない最も大きな問題は，この新たな文化の普及である．より具体的には，Prolog の質のよい処理系を現在出まわっている商用計算機上で実現し，誰もがそれを利用できる環境を整備することである．現在，すでに国内でもいくつかの処理系が作られて，その利用者も徐々に増えてきているが，いまだ本格化してはいない[9]，[10]．

また，知識工学自体でいえば，これからは，ますます各分野の専門家との協力が必要となるであ

ろう．すなわち，学際的なアプローチがぜひとも不可欠である．この特集をきっかけとして，そのような気運が起こり，共同研究が生まれることを期待して，本稿を閉じることにしたい．

参考文献

[1] Feigenbaum, E.A. "The Art of Artificial Intelligence : Themes and Case Studies of Knowledge Engineering", Proceeding of the 5th International Joint Conference on Artificial Intelligence, MIT, Cambridge, Massachusetts, USA, 1977

[2] 辻井 "プロダクションシステムとその応用", 情報処理, 20, 8, pp.735-743 (昭54-08)

[3] Minsky, M. "A Framework for Representing Knowledge", in Psychology of Computer Vision, McGraw Hill, 1975

[4] 横井, 淵 "推論機構を内蔵した述語論理型言語 Prolog", 日経エレクトロニクス,「人工知能」特集号(No.300), 1982年9月27日号

[5] 古川 "第5世代計算機のソフトウェア技術の展望", 電子通信学会誌, Vol.66, No.4,「ソフトウェア生産技術の現状と将来」特集号, 1983

[6] 竹内, Shapiro, E. Y. "論理型言語によるコンカレント・プログラミングについて", 情報処理学会ソフトウェア基礎論研究会資料4, 1983年3月

[7] Mizoguchi, F., Miwa, K., and Honma, Y. "An Approach to PROLOG Basis Expert System", Proc. of the Logic Programming Conference '83, Tokyo, March 22-24, ICOT, 1983

[8] 竹内, 古川, Shapiro, E. Y. "Concurrent Prolog によるオブジェクト指向プログラミング", ibid

[9] Nakashima H. "Prolog/KR User's Manual, METR 82-4, Dept. of Mathematical Engineering, Univ. of Tokyo, 1982

[10] 梅村 他 "文字列処理を導入した Prolog : Shape Up について", Proc. of the Logic Programming Conference '83, ICOT, 1983