解　説

エキスパート・システム概論†

上　野　晴　樹††

1. はじめに

エキスパート・システム (expert systems) は，対象とする問題領域の専門知識を利用して推論を行い，専門的に高度な問題の解決に関して，専門家（エキスパート）と同等の能力を持つ（ことを目標とする）知的問題解決システムをいう．従来型のソフトウェア・システムが手続き中心型であることと対照的に，知識中心型であるので，この点を強調して知識型エキスパート・システム (knowledge-based expert systems) と呼ばれることも多い．ただし，最近では必ずしも能力の高さにとらわれず，知識ベース型の問題解決支援システム，もしくはエキスパート・システム開発ツールを用いて開発されたシステムを総称して，エキスパート・システムと呼ぶ傾向が強くなってきた．

エキスパート・システムは専門家に代わって非専門家の仕事を支援したり，専門家自身の仕事を支援するために活用される．前者は能力を補うことが主たる目的であり，後者は生産性や品質を高めることが主たる目的である．前者の典型は，医学診断支援システムにみられるように，その分野についての非専門家の意思決定を，専門家の代わりに支援するためのシステムであり，後者の典型は，機器やシステムの設計や計画などの作業を支援するように働くシステムである．

このようなエキスパート・システムの対象問題はまた，医学診断，機器やシステムの故障診断，法律相談，政策決定，性能判定，能力評価などのような分析型問題と，CAD，システム設計，プログラム設計，VLSI 設計，薬品設計などのような合成型問題とに大別される．前者は，与えられたデータの分析に基づいて想定される仮説の中から最も可能性の高いものを選択するという問題であり，後者は，与えられた要求を最もよく満たすようなシステムを部品の合成によって生成するという問題である．両者は対照的問題であるが，共通な点は，いずれもその問題領域のエキスパートの知的能力を必要とすることである．ただし，これらの問題は知識表現や問題解決のアプローチにおいて，相互に関係が深く，特にいわゆる深い知識 (deep knowledge) に基づくシステムにおいては区別することの意味が少なくなるか，もしくは困難になると考えてよいであろう[18]．

エキスパート・システムの研究は，1965 年にスタンフォード大学の Feigenbaum らによって開始された化学構造式同定システム DENDRAL[1] に端を発することは知られているが，主として米国の大学や研究機関において 70 年代に盛んに研究開発が行われ，80 年代になってそれらの成果が産業界へ波及し始めた．今日では，世界的に先端技術の代表のように謳められ，企業化も急速に行われていることは，周知の事実である．しかしながら，現場で歓迎されると期待されていた医学エキスパート・システムがいまだに実験システムの域を抜け出られずにいるなど[2]，エキスパート・システムの実用化が当初予想されていたより容易ではないことが明らかとなり，見直しが行われた結果，どんな問題が容易でどんな問題が困難かということや，学問的研究として取り組むべき課題などが認識されるようになった．

ここでは，エキスパート・システムの本来の概念，歴史的考察，および技術的課題などについて概説し，本特集の各解説を読むための準備とする．

2. エキスパート・システムとは

エキスパート・システムの定義をまず行って，次に基本的概念や構造などについて論ずる．前述のように多少あいまいな点があるが，ここでは本来の考え方に従って，「エキスパート・システムとは，問題領域の専門家（エキスパート）から獲得された専門知識を用いて推論を行い，専門的に高度な現実の問題を，専門家と同等のレベルで解決する知的システムをいう」と定

† Expert systems by Haruki Ueno (Department of Systems Engineering, Tokyo Denki University).

†† 東京電機大学理工学部経営工学科

義しよう．キイポイントを箇条書きにすると，

—問題領域の専門知識を用いて推論を行うこと

—エキスパートから獲得された知識であること

—専門的に高度な現実の問題を対象とすること

—能力がエキスパートと同等であること

などである．つまり，エキスパートでなければうまく解決できないような問題を，エキスパートと同等のレベルで解決できる能力(performance)を持つことが重要なポイントである．(ただし，これを目標として開発されつつあるシステムもこの範疇に含めることにしよう．)

このような問題領域は，一般に複雑であいまい性を含んだ不完全な開いた世界(ill-structured open world)であり，科学的あるいは理論的に取り扱うことが困難である要素を多く含んでいるという特徴がある．エキスパートと呼ばれるほどの専門家は，その問題領域に関する高度に専門的な知識に精通しているばかりでなく，彼ら自身の永年の体験をとおして得たいろいろな経験則(heuristics)を利用して“うまく”推論していると考えられる．一般の人にとっては途方にくれるような難問をいとも簡単に解決してくれる場面に接したとき，“さすがにエキスパートは違う”とうならされる．また，身近にエキスパートがいてほしいと思うことが少なくない．

エキスパート・システムの目的の一つは，専門家に代わって非エキスパートの問題解決を支援することである．コンピュータ化されたコンサルタントと考えてもよい．意思決定支援エキスパート・システムのことをコンサルテーション・システムと呼ぶことがあるのはこのためである．コンサルタントが相談者に代わって意思決定することがないように，コンサルテーション・システムも支援する立場を守り，最終的意思決定は相談者がみずから行う．コンサルテーション・システムは，相談者に質問を発し，入力された情報を分析し，推論によって得た結論を示し，対策に関する助言を行い，結論の根拠を説明するなど，相談者の意思決定の支援となるいろいろな機能を持つ必要がある．

エキスパート・システムのユーザが専門家である場合もある．この場合には，問題解決に関する作業能率や信頼性の向上などの利点が大きい．現代のように取り扱う問題が大規模複雑になり，専門領域の細分化が進んでくると，一人の専門家だけで関連するすべての領域に関する十分な知識に精通することは困難となるばかりでなく，勘違いや見落としなどからくる作業能率の低下や信頼性の低下が避けられなくなっている．このような状況では，複数の専門家の知識を知識ベースに統合して彼の問題解決を支援することが有効となる．複雑な問題のコンサルテーション・システムや，各種設計支援システムなどこの類の応用は少なくない．

3. エキスパート・システムの構造

では，このようなシステムを実現するにはどんなソフトウェア構造をとることが必要かを考えてみよう．エキスパート・システムはソフトウェア・システムとしてみると，知識ベース・システムである．知識ベース・システムは，知識を管理する知識ベース(Knowledge Base)と知識ベース内の知識を利用して推論を制御する推論機構(Inference Engine)とを分離して持つシステムである．両者を分離することによって，知識の定義や管理を容易にし，応用エキスパート・システムの開発を容易にする．つまり，類似の推論形式が適用できる広範な応用に対して，知識ベースを入れ替えるだけで対応できるからである．これに対して，従来型の手続き中心型プログラムにおいては，知識は手続きの中に一体化されているので，個々の応用ごとにプログラムを開発する必要がある．ただし，知識の表現と推論制御の方式とは不可分の関係にあり，特定の知識表現を選べばそれに適した推論制御方式は限定されたものとなる．

エキスパート・システムは専門知識を用いて問題の解決を支援するためのシステムである．したがって，応用分野の違いによっていろいろな形態があったとしても，基本的に必要な機構はおのずと決まる．このような観点からエキスパート・システムに必要な構成要素とそれらの関係を考えてみよう．

典型的なエキスパート・システムは，**図-1** に示されるような構成要素と構造を持つ必要がある．すなわち，

—専門知識を表現しそれを統合的に管理する機構である知識ベース

—知識ベース内の知識を利用して，推論を実行するための推論機構

—ユーザとの応答をスムーズに行うためのユーザ・インタフェース

—エキスパートから専門知識を獲得し知識ベースを構築する作業を支援するための知識獲得支援機構

—ユーザの要求に応じて，推論で導いた結論の根拠を説明するための推論課程説明機構

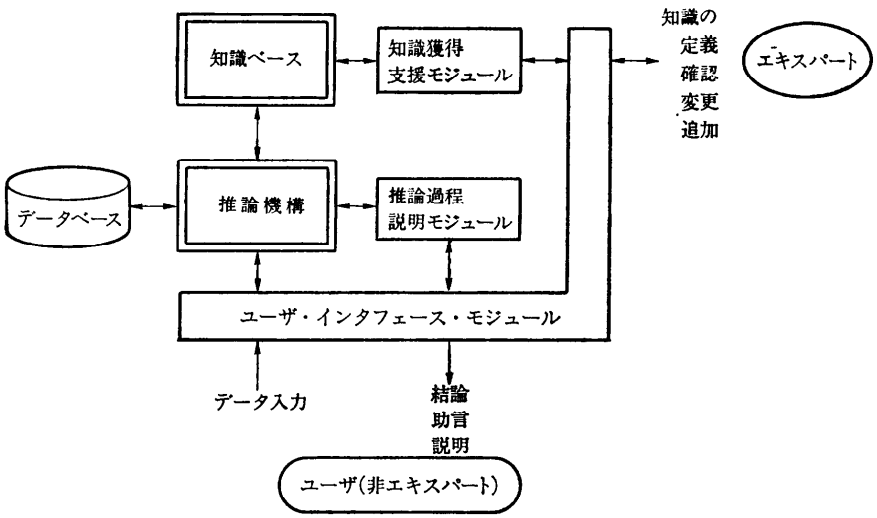

図-1 エキスパートシステムの基本的構造

などである．エキスパート・システム開発ツールは，これらの機能モジュールを持つソフトウェア・システムである．ただし，ここで示したものはごく基本的な構造にすぎず，実際のシステムは他のシステムとのインタフェースなど，問題によって付加的機構が必要となる．

4. 歴史的背景

ここで，エキスパート・システムを中心とする人工知能研究の歴史を簡単にながめてみよう．人工知能の研究者たちが推論中心から知識中心の研究へと方向転換を始めるきっかけになったのが，1965年にスタンフォード大学で研究が開始された前述の DENDRAL であることが知られている．これはエキスパート・システムと呼ばれる最初のものであり，構造が未知の有機化合物の分子式と質量スペクトル・データとを入力して，最も可能性の高い化学構造式を推定するシステムである．有機化合物の熱分解に関する専門知識や質量スペクトルと構造とを結びつけ，一致度の重要性を評価するための経験則を利用することによって，専門家とほぼ同程度の同定能力を実現できたという[1]．このシステムは三つの機能部分すなわち拘束条件生成器，構造生成器（仮説生成器）および構造検証器（仮説検証器）から構成されており，“計画，仮説生成および検証”（plan-generate-and-test）の枠組みで問題解決を行っている．たとえば，$C_8H_{16}O$ は1684のトポロジカルな構造をとりうるが，質量分析器から得られたデータ（ピーク値の組）から推論された拘束条件である“必要な部分構造のリスト”と“あり得ない部分構造のリスト”を適用すると，約40にしぼられ，さらにこれらから論理的に生成されたスペクトルと与えられたスペクトルとの類似度の比較により，数個の候補に絞ることができた．DENDRAL は上の例のような簡単な飽和非環式単能化合物のみを処理するいわゆるデモシステムであったが，AI システムという観点からは画期的なものであった．

この DENDRAL はプロダクション・システムとして知られているが，最初からそのような形態をとっていたのではなく，試行錯誤の結果としてたどり着いたようである．つまりシステムの問題解決能力を高めることに焦点をしぼってプログラムの改良等の努力の過程で，知識表現をプロダクション・ルールで統一することにより，理解しやすく，変更や拡張が容易となり，推論制御メカニズムも簡単な構造をとるなど，柔軟性に富んだシステムが実現できるという理由によるものであった．

70年代に入ると，DENDRAL の成功に刺激されて，知識型エキスパート・システムの研究が米国の大学で盛んに行われるようになった．MYCIN[3]，PIP，CASNET[4]，INTERNIST[5]，HEARSAY-Ⅱなどのシステムがこの頃開始されたものである．MYCIN は感染症の診断および投薬決定支援システム，PIP は腎臓病の診断支援システム，CASNET は緑内障の診断治療支援システム，INTERNIST は内科全般にわたる診断支援システム，HEARSAY-Ⅱは（エキスパート・システムではないが）話言葉理解システムである．こ

の時期の研究開発の重要な特徴は，応用問題指向型の研究であるということである．専門知識の表現モデル，推論制御機構や，コンサルテーション機能などに，主として能力をいかに高めるかという観点から研究の焦点があてられたといえる．また，図-1に示されるように，知識ベースと推論機構を分離するというソフトウェア構造が採用されるようになったことが，大きな特徴であった．

たとえば，MYCIN は最初からすべての知識を IF-THEN 型ルールで表現するプロダクション・システムとして設計されたが，ここではさらに，コンサルテーションに必要な HOW や WHY などの推論課程に関する説明機能や助言機能が組み込まれた．この MYCIN は実際の医学データに基づくテストではかなり高い診断能力を示したが，現場のユーザには受け入れられるまでには至らなかった．また，このシステムが初めて採用したあいまいさを取り扱うための CF (確信度係数) とその評価法やプロダクション・システムそのものの長所短所などについて，さまざまな議論がまき起こった．要約すると，MYCIN のようなプロダクション・システムの構造では，知識表現，能力，推論速度，あいまいさの取り扱い方などにおいて，複雑な問題への応用が困難である[6]というものであった．また，MYCIN は代表的な後ろ向き推論型プロダクション・システムであるが，このタイプの推論は原理的に診断や評価に適しているので，現在でも応用を限定してもしくはより汎用な推論制御の一部として使われている．

さて，これらのシステムで採用された知識表現モデルは，MYCIN がルール・モデル，PIP と INTERNIST がネットワーク・モデル，CASNET が因果ネットワークモデル，HEARSAY-Ⅱが黒板モデルなどである．これらはいずれも特定の応用分野のための専用システムとして設計された．したがって，知識表現形式や推論制御の方法に，それが対象とする問題そのものからくる特徴や，それを開発した研究者の好みなどがにじみ出ている．人工知能システムが人間の知能のモデル化という側面を強く持っていることから，これは当然のことであると言える．これらの研究成果はそれぞれ学問的に新しい知見をもたらし，その後この分野の研究に多大の影響を与えた．しかしながら，いずれも実験システム以上のものではなかったといってよい．

70 年代後半になるとこれら個別的応用志向のエキスパート・システム開発の経験をもとにして，汎用化された知識表現言語が開発され始めた．つまり，上述のシステムはいずれも一種の応用プログラムとして設計されたものであったが，個々の問題解決のために工夫された知識表現形式および推論制御技法は，それらが対象とした問題に類似の別の問題にも応用できるわけであり，かつ性格的に類似の問題が少なくないことが，これらのケース・スタディをとおして明らかとなってきたからである．別な表現を使えば，このようなことを念頭に置いて問題分野が選択されたといえる．AI 研究は一種の実験科学であると指摘されるのは，これが主な理由である[7]．つまり，適切な問題を選択し，それの解決をとおして一般性の高い原理を究明していくのが，いまだ完成されていない研究領域における適切な研究のありかたであり，これまでの人工知能の歴史である[8]．

さて，これらの研究をとおして，ルールやネットワーク，あるいはブラックボードによる知識表現と推論制御の普遍性と有用性がある程度確認されたが，汎用ツールに大きな影響を与えたモデルあるいは理論としては，このほかに Minsky のフレーム理論[23]，一階述語論理，意味ネットワークなどがある．

70 年代後半になって開発された知識表現言語の代表的なものは，MYCIN の汎用化版である EMYCIN[9]，CASNET での経験をもとにルール・モデル化された EXPART，HEARSAY-Ⅱを汎用化した AGE[10]，HEARSAY-Ⅲ，およびフレーム理論をもとに設計された FRL，KRL，UNITS[11] などがある．このほかにルール・モデル (プロダクション・システム) として，OPS，ROSIE，RITA などが知られている．我が国で開発された初期の知識表現言語の例としては，MECS-AI (プロダクション・システム)，COMEX[12] (簡易フレーム・モデル)，FMS[13] (フレーム・モデル) などがあるが，これらは 70 年代後半から 80 年代前半に開発されたものである．ヨーロッパで開発された知識表現言語に PROLOG がある．これは一階述語論理に基づいて設計された言語であるが，知識表現言語としても利用され，LISP と並ぶ人工知能用の汎用プログラミング言語として普及される傾向にある．なお，これらの言語の多くはエキスパート・システム開発言語として設計されたが，フレーム型言語のように，汎用性の高いことから，画像理解や自然言語理解あるいはその他に応用されているものもある．

知識表現言語が提供され始めたことによって，大き

な変化が起こり始めた．これらの言語を利用して比較的簡単にさまざまな応用エキスパート・システムが試作できるようになったことである．その中でも特にプロダクション・システムは比較的修得が容易であり，使いやすく，簡単な応用問題には十分な能力を持っていることが分かるにつれて，医学診断，機器・システムの故障診断や営業活動支援などに，それまで人工知能はリスクが大き過ぎるとして手を出さないでいた産業界が，次々と応用し始めた．

80年代になると，この傾向に加速度がつき，それまで大学の研究室内で行われていた研究成果を米国の産業界が先を争うように導入するようになった．これに対応して人工知能応用に的を絞ったベンチャ・ビジネスが出現し始めた．これらは大学の AI 系教授によって設立されたものである．一方，我が国では 82 年に発足したいわゆる第5世代コンピュータ開発計画と推進機関としての ICOT の設立が契機となり，大学や産業界におけるエキスパート・システム研究開発が盛んとなり，今日の隆盛に至っていることは良く知られるとおりである．ただし，研究の歴史，研究実績，研究者，専門技術者および教育研究環境などに関して，日米間の格差はいまだかなり大きいのが実情である．

これらの変化にともなって，エキスパート・システム研究開発環境の方も格段に改善されてきた．すなわち，LISP マシンに代表されるいわゆる AI ワークステーションの出現と，マイコン用から大型汎用コンピュータ用に至る各種のエキスパート・システム開発ツールの提供は，それまでの高価な共同利用型 AI コンピュータより優れた環境を各研究者レベルにもたらし，どこでも手軽にエキスパート・システムの開発が可能となった．

以上，簡単にながめてきたように，エキスパート・システムの分野は 10 年前とは大変な違いがある．実用性のあるエキスパート・システムも R1 (XCON)[14] など，少しずつ現れており，うまく問題を選べば実際に役立つシステムの実現も可能であることが実証されつつある．しかし，一方では基本的かつ重大な壁に直面していることも事実である．プロダクション・システムを使って解決できる問題が多い反面，当初，最も応用事例の多かった医学診断治療支援の分野において，いまだに現場で実際に利用されていないことに見られるように，当初の見込みに反して次々に困難な問題が現れ，解決の糸口をさがすために悪戦苦闘しているケースが少なくないのが現状である．これらは〔簡単なデモ・システムと実用システムとの間には，天と地ほどの差がある〕ことを示している．

5. エキスパート・システムの応用（分析，合成，制御）

エキスパート・システムの応用はきわめて広範にわたっており，いろいろな報告もあるので[15),16)]，ここでは応用分野について議論することはやめ，その代わりに，問題を代表的な分析型，合成型，制御型の三つのタイプに分けて，それぞれの性格の違いやアプローチの違いを論じることにする．また，具体的な議論はそれぞれの解説でなされているので，ここでは著者の個人的見方で比較することにする．なお，分析型は選択型あるいは診断型とも呼ばれ，合成型は設計型とも呼ばれる．まず分析型と合成型の問題およびエキスパート・システムについて論じ，最後に制御型の問題およびエキスパート・システムについて論じる．制御型は，分析型の性格を強く持っているが，実時間で応答することや，連続運転する必要がある点など，異なる面がいくつかある．

5.1 分析型と合成型

エキスパート・システムが対象とする問題は多種多様であるが，問題解決の性格から眺めてみると，お互いにきわめて対照的な二つのグループすなわち分析型問題 (analysis-based problems) と合成型問題 (synthesis-based problems) に分類できるようである．ただし，両者の境界は必ずしも明瞭に分割できるわけではない．ここでは一応この二つに分類できるものとして，両者の間の相違点について議論しよう．この違いは非常に重要であり，知識表現モデルおよび推論制御技法の選択に大きな影響を与えるだけでなく，実現の困難さが大きく異なる．

分析型意思決定問題は，あらかじめ設定された仮説の集合の中から，与えられたデータを最も良く説明する仮説を，データの分析に基づいて選択して結論とする類の問題である．たとえば，医学診断支援システムでは，あらかじめ設定された仮説（病名）の集合の中の一つ（あるいは複数個）を，患者データの分析に基づいて選択（診断）するという手段がとられているので，典型的な分析型意思決定の問題に入る．これと類似の問題としては法律相談，故障診断，政策決定，能力評価などがあげられる．

一方，合成型意思決定問題は，一定の拘束条件 (constraints) のもとで与えられた要求を満足する最適（ま

たは妥当）なシステム構成を生成する類の問題である．一般に設計では複数個の要素（部品）を組み合わせて，目的とする一つのシステムを構築するので，あらかじめ仮説として解の候補のすべてを設定しておくことはできない．分析型問題における処理はアナリシスが中心であり，合成型問題ではシンセシスが中心である．たとえば，プログラム作成は典型的な合成型問題である．ここでは，与えられた要求事項に基づいて設計仕様を作成し，これに基づいてプログラム構造（データ構造およびアルゴリズム）を設計し，これをプログラム言語で記述するというプロセスがとられる．複雑で大きな問題になると，この各段階で設計者のエキスパタイズが利用される．ただし，作られたプログラムは想定されたコンピュータ環境下で与えられた要求どおりに働く完全なものでなければならない．つまり，設計の各段階では設計者の経験則や趣向に基づいた意思決定（これは論理的に不完全なものである）が入るが，問題解決の結果としての製品は完全なものでなければならない．合成型意思決定の問題には，スケジューリング，LSI 設計，電子回路設計，プラント設計，実験計画，薬品設計，材料設計，構造物設計やいわゆる CAD が含まれる．医学における治療計画や CAI にも合成的色彩がある．

ただし後述するように，それぞれの問題の性格の違いによって，アプローチが大きく異なる．つまり，合成型問題といっても，問題分野によって全く異なった方法を必要とする．さらに，手順や方法が比較的標準化されている分野がある反面，まだ混沌としている分野もある．前者に対しては有効な手段が見いだされつつあるが[17]，後者に対しては AI 以前の問題が多いようである．これらのことが合成型問題への応用を困難にしている．

さて，分析型意思決定問題の特徴は，選択すべき仮説の集合をあらかじめ設定することができ，これと選択規則の集合によってつくられる問題空間（problem space）が限定できるので，モデルの作成が容易なことである．（逆に言えば，問題空間が限定できない場合は困難である．）典型的な方法は，所見と仮説とを結びつける評価規則を知識ベースとして準備しておき，与えられたデータに基づいて各仮説に対する評価値を求め，値の最も大きな仮説を結論として決定するというやり方である．AND/OR 木でモデル化できる場合が多い．これに対して，合成型意思決定問題では，あらかじめ解の候補を設定しておくことができないので，極端に言えば無限に近い仮説の中から解を決定することに相当し，問題空間が非常に大きくなり，推論モデルの設定が困難となる．ただし，設計は部分的に見ると選択で置き換えられるので，一定の拘束条件を満たす部品の組み合わせのみが解となる．この場合，解の候補が複数存在するとき，最適の候補を決定しなければならない．したがって，最適性の評価が必要である．

また，分析型問題では各仮説は記号で現すことができるので比較的簡単であるが，合成型問題では，設計対象の構造的モデル表現が必要となる．複雑なシステムのモデル表現およびその合成などの操作は難しい問題であり，これまでに開発されたエキスパート・システムの大部分が分析型問題を対象としたシステムであって，合成型問題はいまだ簡単な例が実現されているに過ぎないのは，対象のモデル（object model）の問題が十分解決されていないことも理由の一つである[18),19)]．

合成型問題へのアプローチは，さらに2種類に分類できるようである．第1アプローチは，一定の拘束条件の下で，与えられた要求を満たすシステムを合成するというやり方である．合成に基づく合成といえよう．要求がどんな形で与えられるかによって異なるが，与えられた要求を満たすようなシステムを，利用可能な部品をうまく組み合わせながら合成していくというやり方がその典型である．設計仕様の作成はこの種の代表的問題であると思われる．また，要求として明確な形で与えられる形態とは異なるが，与えられたデータから拘束条件を推論し，標準部品を組み合わせてシステムの合成を行うエキスパート・システム，たとえば，DENDRAL，HEARSAY-Ⅱ，R1などがこの範疇に含まれるものである．すなわち，DENDRAL は質量スペクトルのデータの中に，HEARSAY-Ⅱでは音声信号データの中に暗黙の要求が含まれており，R1 では注文表が要求を現していると考えることができる．いろいろな実現法が考えられようが，このような表現の要求から明確な要求を引き出す仕事（推論）を問題解決における最初のステップとすれば，スッキリとした推論制御の構造となろう．ここでは，専門家のエキスパタイズが利用される．次のステップが合成の処理となる．このように考えると，画像理解，音声理解，自然言語理解なども，これと類似の合成型問題であると言える．

第2のアプローチは，マクロな表現を詳細な表現へ展開（もしくは分解）することによってシステムの合

成を行うというやり方である．展開に基づく合成と言えよう．厳密な表現形式で記述された仕様書が与えられて，一定の拘束条件の下で，具体的システム設計に展開していくという設計のやり方がこの典型である．したがって，設計のプロセスが標準化されていなければならない．すなわち，設計作業が多段になっており，かつ各段階における入力の仕様記述と出力の仕様記述および展開の方法や評価の基準などが標準化されていることが必要である．たとえば，知的 VLSI 設計システムとして，与えられた仕様記述から，演算モジュール，論理回路，さらには電子回路へと，次々に自動的展開を行って詳細設計をするというアプローチは，この例である．自動展開処理において，専門家のエキスパタイズを利用した推論を適用して，無駄な探索をできるだけ避けるような工夫が重要となる．展開による設計のアプローチは，機械設計，建築設計の分野には応用可能であると思われるが，ソフトウェア開発や薬品設計，材料設計などのように，標準化が遅れているかもしくは困難な分野には適用困難であるはずである．後者に適用可能な方法は，第1のアプローチの考え方の延長となろう．

なお，合成型問題と分析型問題を明確に区別することは必ずしもできない場合がある．それは，合成に基づく分析 (analysis by synthesis) によって診断を行うという，高度な診断法がこの例である．この場合にも，与えられたデータに基づいて推論を行い対象のモデルを合成するという方法が採られる．診断は，このモデルを評価することによって行われる．したがって，高度な問題解決は合成型問題として取り扱われる必要があるはずである．

5.2 制 御 型

制御型の問題は，エキスパート・システムとしてとらえると，どちらかといえば分析型問題に近いが，制御に特有の問題も多い．各論は別の解説にゆずり[38]，ここではごく基本的な特徴について議論しよう．

簡単に言えば，システムの制御とは，システムを定められた一定の条件の中で運転することであると言えよう．制御対象のシステムは，一般に複雑であること，定常状態と過渡状態で異なった制御が必要であること，特に運転開始などの過渡状態には不安定な振る舞いをしがちであること，などの特徴がある．このようなシステムを，制御するには，連続的に監視し，異常を検出あるいは予測し，適切な制御行為をとる必要がある．また，いろいろな外乱の中で，望ましい挙動をとることも要求される．さらに，他のエキスパート・システムと大きく異なる点は，実時間で推論を行い，制御指令を出すという“速さ”が基本的要件であるということである．制御対象が複雑で変化が速くなると，現在のエキスパート・システムでは致命的問題となる．また，電力や石油化学などの大規模複雑なプラントの制御や，電車や船舶などの運転制御は，大きく異なる．したがって，当然異なったアプローチが必要となる．

なお，いわゆるフィードバック制御を中心とする古典的制御理論は存在し，設計や運転制御において応用されているが，上記のような条件は複雑すぎて，部分的応用を除いてはとても取り扱えない．コンピュータ制御はかなり複雑なシステムを対象として実用されて

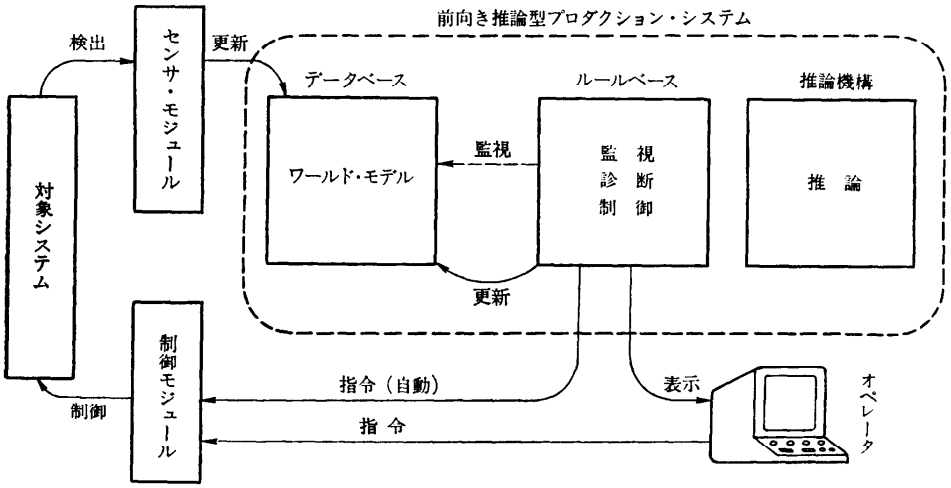

図-2 プロダクション・システムによるシステム制御の概念図

いるが，制御プログラムの開発や保守において問題がある．現在は熟練技術を持った人間が技術の不備を補っているが，熟練技能者を確保することは困難となりつつある．エキスパート・システムの概念や技法が注目されているのは，このようなことが背景にある．

さて，簡単な制御エキスパート・システムは，**図-2**に示されるようなプロダクション・システムによって実現できよう．高度なシステム制御においては，予測制御の機能が必要となり，数理モデルによるシミュレーションの機構を含む必要がある．すなわち，現在のシステムの状態とデータに基づいて，シミュレーションで将来を予測し，もし問題があれば，いくつかの対策を選択し，シミュレーションで評価を行い，最適な制御を決定することは，安全を保証するために不可欠である．ただし，これは将来の課題である．現在のところでは，いまだスピードさえも十分に克服されていないはずである．

なお，ファジー制御はエキスパート・システムか否か賛否両論があるが，著者はエキスパート・システム（すなわち AI システム）ではないと考えている．

6. 知識表現言語について

エキスパート・システムにおける知識表現言語に関する解説は別にあるので[39]，ここでは観点を変えて，概念的な知識表現および言語の意味を議論してみる．

従来型のプログラミング言語が手続き型であるのに対して，知識表現言語は多分に宣言型であることは，よく指摘される事実である[20]．宣言型のメリットは，知識を表現しやすい点にある．つまり，一つ一つの表現が簡潔であり，かつ柔軟であり，変更や追加に対応しやすいという特徴がある．逆に，実行においては，処理の大半が適切な知識の探索に費やされて，効率が非常に悪いという欠点がある．一方，フレーム型知識表現モデルのように，複雑な問題解決を効率良く行うシステムを実現するために，宣言型知識と手続き型知識とを組み合わせるやり方を積極的に採用しているケースもある[23]．これは，単にコンピュータの処理速度を上げるという技術上の目的からではなく，人間がシーンの理解や物語りの理解をきわめて速く行っているという事実を基に，これを実現できるための知識表現と推論の認知科学モデルとして提案されたものである．そこで，これらのことを前提にして，知識の表現と言語に関して，（多少独善と偏見を交えて）考察する．

まず，人間の知識構造を，“抽象一具体”の軸上で考えてみると，抽象レベルは宣言的かつ構造的であり，具体レベルでは手続き的であるように思われる．ここで，宣言的表現とは，“一は一である”，“一と一の関係は一である”とか，“もし一ならば一である”などと現される表現である．最初の二つは事実とその関係を表し，最後は判断を表す．事実間の関係が構造となる．すなわち，これらを階層的かつ網的に組み合わせたものが，抽象レベルにおける知識構造のモデルである．手続き的表現とは，フローチャートで表現できる具体的処理手順の記述をいう．一般的には，次のようなことが言える．抽象的であるほど（つまり宣言的であるほど）人間にとって表現が容易で理解しやすい反面コンピュータにとっては処理効率が悪くなる．これに対して，具体的であるほど（つまり手続き的であるほど）表現が困難で読解性が落ちるが，処理効率が高くなる．AI アーキテクチャを持つコンピュータの開発の目的は，この問題の解決にある．このような角度から代表的な知識表現言語の特徴を比較してみよう．

述語論理は完全に宣言型であるが構造的ではない．宣言型という点については，知識表現言語に向いていると言える．ところが，前述の主張と矛盾するようであるが，あまりにも厳密過ぎる点が，人間の知識表現および推論とは大きな距離があることがたびたび指摘されており，各種のあいまいさを含んだ複雑な問題解決を対象とする専門知識の表現には向かないという批判が，特に認知科学系の人工知能研究者から出されている[8]．一方では，学問としての理論の重要性を主張する研究者からは，数学的根拠に基づく体系を持つ唯一の言語である点が強調されており，矛軟な表現力などに乏しいなどの批判に対しては，それなりの拡張もされつつある[19]．最近では，ATMS に見られるように，述語論理の研究成果が他の知識表現の中に採り入れられるようにもなってきた．

PROLOG は宣言的かつ手続き的言語であるが，構造的ではない．述語論理のサブセットを基に，手続き的解釈の拡張をほどこして，汎用プログラミング言語としての機能を持っている．特にこの言語の性格に向いた問題に関しては，LISP に比較して圧倒的に少ない表現ですむという簡潔性が高く評価されている．ただし，常に PROLOG の性格を意識してプログラミングしなければならないことや，単純なミスでプログラムが暴走するなどの欠点がある．後者はいわゆるフラットな言語の共通の欠点である．ソフトウェア工学の研究は，大規模なプログラムにおいてはモジュール性

が不可欠であることを教えており，プログラムの構造化に努めてきた．したがって，著者はこの欠点を重要とみるが，そうでもないという意見もある．また，事実の構造的表現が望ましいという観点からも，この言語における事実の表現は十分ではないと思う．このような理由で，著者は，この言語の特長を生かし，弱点をおさえるために，フレーム型言語に組み込んだ試みを行ってみた[24]．

プロダクション・システムは宣言的かつ手続き的言語である．プロダクション・システムは，IF-THEN ルールの集合であるルールベース，事実（対象）の集合であるデータベース，およびインタプリタを基本構成要素として持つ．ルールは宣言型と手続き型の双方の性格を持っている．すなわち，各ルールは単独に宣言さるたものとみなすこともできれば，たいていの手続きはルールの集合で記述できるから手続きであると考えることもできる．AND/OR 木に基づく分析型プロダクション・システムは宣言型表現の性格が強く，認識一実行サイクルに基づく合成型プロダクション・システムは手続き型表現の性格が強い．また，データベースは対象世界のモデル表現であり，完全な宣言型知識表現と言える．これに文脈木やフレームを採用することによって，構造的表現も可能となる．このように，分析型か合成型か，さらにはデータベースの構造によって，性格や適用分野が大きく変わるのも，プロダクション・システムの特徴の一つである[24]．

フレーム・システムは，宣言的かつ構造的で，しかも手続き的である．フレーム・システムにおける知識の表現単位は，フレームと呼ばれる比較的大きなデータ構造であり，この中に事実の宣言とそれを取り扱う手続きとが組み入れられている．さらに，複数のフレームは抽象一具体関係に基づく階層構造を形成し，属性情報の自動的継承機能を持っている．これに加えて，フレーム間ネットワークを構成することもできる．したがって，このような機構をすべて備えたフレーム・システムは，最も汎用性の高い知識表現言語であるといえ，汎用知識工学環境として用いられているゆえんである[26]．

意味ネットワークは，宣言的かつ構造的であるが，手続き的ではない．意味ネットワークは対象をノードとし，このノード間に特定の意味を持つリンクをはることによって構成されたものである[25]．また，多くの例が抽象一具体関係に基づく階層構造を取り入れている．ただし，一般にノードのデータ構造は小さな単位であり，単なるメモリに近い．もともと意味ネットワークは自然言語処理などにおける一種の意味辞書として開発されたものであり，機械翻訳や CAI における知的辞書などとして応用されている．

なお，従来型プログラムにおけるように，手続きとしてプログラムを記述するやり方では，知識は人間によって前もって手続きにコンパイルされた形でプログラミングされる必要があるので，開発が当然困難となる．知識表現言語は，人間の専門知識を表現するためのモデル化言語であるから，知的問題解決プログラムの開発が容易となるわけである．ただし，多様な人間の知識をすなおに表現できるほどに完成された知識表現モデルはいまだ存在しない．多かれ少なかれ，人間が持つ知識を特定の表現に変換（コンパイル）しなければならない[26]．たとえば，IF-THEN ルールで知識を表現するにも，人間によるコンパイル作業が結構多くなる．したがって，問題の性格に合った知識表現形式を持つ言語を選択することが肝要である．

7. おわりに

エキスパート・システムは，60年代に芽生え，70年代に主として米国の大学の研究室内で急速に進歩し，80年代に入って産業界へ波及し始めた．現在いろいろな問題分野でエキスパート・システムの開発が盛んに行われているので，80年代の終わり頃までには応用例が増えて，実用化もかなり進むものと思われる．ただし，そのためには現在の技術レベルを考慮した適切な問題を選定することが前提である．エキスパート・システムは従来型のプログラムより知的意思決定の処理においてははるかに強力ではあるが，コンピュータ・プログラムであることには変わりがない．つまり，われわれに分からないことのプログラムの作成はいずれにしてもできないわけである．研究の場合にもそうであるが，特に短期間での実用システムを目的とする場合には，問題を良く理解して，十分に見通しのあるテーマに取り組むことが肝要であると思う．次に著者の私見を述べる．

エキスパート・システムほどに使う人によって意味するところの異なる用語は少ないのではないだろうか．専門家の認知モデル，深い知識構造，知識獲得，対象モデルなどの研究をエキスパート・システムの一環として行っている研究者がいる反面[18),27)~34)]，IF-THEN ルールで書かれただけのプログラムをエキスパート・システムと称してビジネス活動に利用してい

る人々もいる．前者はAIの基礎研究の一種であるが，後者はAI技術の応用の単なる試みに過ぎない．マスコミがこれらを区別しないで報道しているだけでなく，エキスパート・システム開発ツールを多少かじっただけの人が誇大宣伝をしているために，混乱が生じている．また，応用といっても，小さな練習問題から複雑あるいは大規模な知識ベース・システムの開発まできわめて多様であり，特に後者はAI研究の質的向上にとってきわめて重要である．つまり，複雑あるいは大規模な問題は，小手先の技術で誤魔化すことができず，根本的な技術開発や基礎的課題の解決を要求することが少なくないからである．AIのような若い学問では特にそうである．米国のアポロ・プロジェクトがシステム工学やコンピュータ科学における学問研究を大きく促進した事実はこの典型である．

我が国の研究者（と称する人々）の多くは，小さな問題を数学モデルで厳密に取り扱うことだけがピュアな学問であると誤解している傾向がみられる．新しいヒントを米国からもらいつづけた過去の歴史となんら変わらない．この種の研究者の傾向は，方法論に特に興味を持ち，現実の応用を避けて，流行語を追い続けることである．このようなやり方から，新しい発想や哲学が生まれるであろうか，著者は悲観的である．苦労の割りに目に見える成果をあげるまでに長時間を要することは避けられないが，適切な応用問題と組み合わせた研究が必要である．

参 考 文 献

1) Buchanan, B. and Feigenbaum, E.: DENDRAL and Meta-DENDRAL: Their Applications Dimension, AI 11 (1978).
2) 上野晴樹：メディカル・コンサルテーション・システムの可能性，第2回医療情報学連合大会論文集，pp. 179-182 (1982).
3) Shortliffe, E. H.: Computer Based Medical Consaltations: MYCIN, American Elsenier (1976).
4) Weiss, S. et al.: A Model-Based Method for Computer Aided Medical Desicion Making, AI 11 (1979).
5) Pople, H.: The Formation of Composite Hypotheses in Dyagnostic Problem Solving-an Exersise in Synthetic Reasoning, IJCAI 6 (1979).
6) Brooks, R. and Heiser, J.: Some Experience with Transfering the MYCIN System to a New Domain, IEEE TRANS PAMI-2, No. 5, pp. 477-478 (1980).
7) Feigenbaum, E.: The Art of Artificial Intelligence, Themes and Case Studies of Knowledge Engineering, IJCAI5 (1977).
8) Newell, A.: The Knowledge Level, AI Magazine Summer (1981).
9) VanMelle, W., Scott, A. C., Bennett, J. S. and Pears, M.: The EMYCIN Manual, STAN-CS-81-885 (1981).
10) Nii, H. P. and Aiello, N.: AGE (Attempt to Generalize): A Knowledge-Based Program for Building Knowledge-Based Programs, IJCAI 6 (1979).
11) Stefik, M.: An Examination of a Frame-Structures Representation System, IJCAI 6 (1979).
12) Ueno, H.: An End-User Oriented Language to Develop Knowledge−based Expert Systems, Proc. IEEE COMPCON FALL, pp. 523-529 (1983).
13) 上野晴樹：FMS：フレーム理論にもとづく汎用知識表現言語，電子通信学会 L 82-64 (1982).
14) McDermott, J.: R1: A Rule-Based Configurer of Computer Systems, AI 19, pp. 39-88 (1982).
15) 上野晴樹：知識工学入門，オーム社 (1985).
16) 森 健一：知識工学への期待，情報処理，Vol. 26, No. 12, pp. 1550-1553 (1985).
17) 長澤 勲：設計エキスパート・システム，情報処理，Vol. 28, No. 2, pp. 187-196 (1987).
18) 上野晴樹：対象モデルの概念に基づく知識表現について一深層知識システムへのアプローチ，電子通信学会 AI 86-4 (1986).
19) 大須賀節雄：知識表現に関する一考察，人工知能学会誌，Vol. 1, No. 1, pp. 20-29 (1986).
20) 大須賀節雄：知識情報処理，電子通信学会誌，Vol. 69, No. 11 (1986).
21) 高木 茂：テンプレートを用いた ALU 回路の自動合成，電子通信学会論文誌，Vol. J 68, D. No. 7, pp. 1369-1375.
22) 伊藤秀昭，上野晴樹：ZERO=Frame+Prolog, Logic Programming Conference (1984).
23) Minsky, M.: Representating Knowledge in Frames, in P. Winston (ed.): The Psychology of Computer ision, McGraw-Hill (1975).
24) 小林重信：プロダクション・システム，情報処理，Vol. 26, No. 12, pp. 1487-1496 (1985).
25) 岡本俊雄：意味ネットワークによる知識表現，情報処理，Vol. 27, No. 12 (1985).
26) 小川 均：フレーム理論に基づく知識表現言語，情報処理，Vol. 26, No. 12, pp. 1497-1503 (1985).
27) Chandrasekaran, B. and Mitta, S.: Deep Versus Compiled Knouledge Approaches to Diagnostic Problem Solving, Developments in Expert Systems, M. J. Combs ed., pp. 23-34,

Academic Press (1984).
28) Chandrasekaran, B. and Milne, R. ed.: Special Section on Reasoning about Structure, Behavior and Function, SIGART, 93 (1985).
29) Davis, R.: Expert Systems: Where Are We and Were Do We Go From Here, The AI Magazine, 3(2) (1982).
30) Hart, P. E.: Directions for AI in the Eighties, SIGART, 79, pp. 11-16 (1981).
31) Yamada, N. and Motooka, H.: A Plant Diagnosis Method Based on the Knowledge of System Description, J. of Information Processing 7, 3, pp. 143-148 (1984).
32) Buchanan, Bluce G.: Research on Expert Systems, HPP-81-1 (1981).
33) Buchanan, Bluce G. and Duda, Richard O.: Principles of Rule-Based Expert Systems, HPP-82-14 (1982).
34) 上野晴樹：知識工学の課題，情報処理，Vol. 26, No. 12, pp. 1554-1558 (1985).
35) 石塚　満：曖昧な知識の表現と利用，情報処理，Vol. 26, No. 12, pp. 1481-1486 (1985).
36) Shortliffe, E. H. et al.: ONCOSIN: An Expert System for Oncology Protocol Management, IJCAI7 (1981).
37) 小山照夫：医療コンサルテーションシステム，人工知能学会誌，Vol. 1, No. 1, pp. 38-47 (1986).
38) 舩橋誠壽，増位庄一：制御分野におけるエキスパート・システム，情報処理，Vol. 28, No. 2, pp. 197-206 (1987).
39) 新田克己：エキスパート・システムにおける知識表現と推論，情報処理，Vol. 28, No. 2, pp. 197-206 (1987).

(昭和61年12月25日受付)