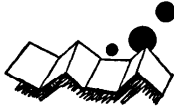


解説

診断型エキスパート・システム†



新井政彦†† 本位田真一††

1. はじめに

診断型エキスパート・システムは、従来より、各種のエキスパート・システムの中で最も早い実用化が期待されていたが、近年、産業界においていくつかの実用化システムが発表され始めている。しかし、これらのシステムにおいては知識ベースは小規模であり、知識表現および推論メカニズムも比較的単純であるのが実状であり、今後予想される応用範囲の広がりにとともにエキスパート・システムへの要求の複雑化に対しては、現状の技術レベルでは対処しきれないという危惧がある。このような産業界の要求に対応して、大学や研究機関では高度な知識表現、種々の推論メカニズムなどのエキスパート・システムの要素技術に関する研究が精力的に行われている。そこで本稿では、まず診断型エキスパート・システムの現状と問題点を踏まえて、これらの新しい技術がどこまで実用化に寄与できるのかの解説を試みる。もちろん、これらの問題点およびそれを克服する新しい技術の中には診断型に必ずしも限定されないものも含んでいるが、診断型エキスパート・システムの立場から述べることにする。

一方、適用事例の増大にとともに、作製されたエキスパート・システムが成功（実用化）するための問題点が指摘されている。たとえば、①何をエキスパート・システム化するのか、②成功するための規模は、③専門家からどのように知識を獲得するのか、などである。すなわち、エキスパート・システムの要素技術自体のみならず、一つのシステムとしての構築技術が問題となりつつあることにも注目する必要がある。一般的には、システムズ・アプローチ的な一般システム構築方法論で論ぜられるべきであるが、知識工学の枠内に限定したときのエキスパート・システム構築方法論およびその中で最も重要な知識獲得における筆者らの見解についても触れる。

2. 診断型エキスパート・システムの現状

診断型エキスパート・システムの研究開発は1970年代に米国の大学や研究機関において主に医療診断を中心に行われ始めた⁵²⁾。それらは感染症の診断支援システム MYCIN⁴⁸⁾、腎臓病の診断支援システム PIP⁵⁰⁾、緑内障の診断治療支援システム CASNET⁵⁵⁾、内科診断支援システム INTERNIST⁴²⁾ などである。しかしこれらの医療診断システムの多くは実用の域には到達していないのが現状である。むしろ、これらの医療診断エキスパート・システムに刺激を受けた形で産業界では1980年代に入って種々の故障診断システムが精力的に開発され始め、いくつかのシステムが実用の域に到達しつつある。国外の実用化レベルのエキスパート・システム例の一覧を表-1に示す。一方、国内においては実用化レベルに比較的近いものとしては、コークス炉消化車の故障診断（東京ガス）、大形直流発電機の故障診断（東芝）²⁷⁾、プロコンの故障診断（新日鉄）、LSI製造プロセスのプロセス診断（日立）、列車電子制御系の故障診断（三菱）、ミニコンOSの性能診断（東芝）¹⁸⁾などがある^{36), 44)}。

一般に診断型エキスパート・システムの対象となるのは、

- (i) 故障診断
- (ii) 医療診断

の二つがある。従来のエキスパート・システムの構築においては専門家（オペレータ）から、原因—症状に関する知識を収集するといった形態が故障診断、医療診断のいずれの場合においても採用されていたため、いずれの診断システムもほぼ同等に扱われていた。しかし、(i)は機械系を対象としており(ii)は人間系を対象としている。機械系の場合はその対象システムの設計者が、構造や基本的な機能を把握しているのに対して、人間系の内部モデルの構造はあまり解明されていないといった相違がみられ、最近の傾向として故障診断においては、構造や機能に関する知識を積極的

† Expert System for Diagnosis by Masahiko ARAI and Shin-ichi HONIDEN (Systems and Software Engineering Div., Toshiba Corporation).

†† (株)東芝システム・ソフトウェア技術推進部

表-1 実用化レベルの診断型エキスパート・システム例^{(1),(2)}

システム名	診断対象	使用ツール/言語	開発会社
AI-SPEAR	VAX シリーズ周辺機器	OPS5	DEC
ACE	電話ケーブル	OPS5	AT & T
COMPASS	電子交換機	KEE	GTE Lab.
CATS (DELTA)	機関車	Forth	GE
KBDS	ベースバンド交換システム	LES (社内ツール)	Lockheed Missiles & Space
Chest Pain	肺病	GC Lisp	Evlin Kinney
Diagnostics 1	精神病	EXSYS	Expert Ware
Mentor	空調システム	GC Lisp	Honeywel
Permaid	ディスク・ドライブ	LOOPS	Honeywel
TOGA	変圧装置	RuleMaster	Radian
TARBOMATIC	タービン機関	RuleMaster	Radian
Wheat Counselor	小麦		ICI

に採用するようになってきている。

一般にエキスパート・システムの機能を定性的に評価する尺度の一つとしてその利用の観点からのものがある¹²⁾。これには

(1) エキスパート・システムに全面的に従う

(2) エキスパート・システムの意見を参考にすると2レベルがある。診断型に限定した場合(1)に属するシステムは皆無であるのが現状である。診断型の場合に、その適用場面の性格からエキスパート・システムがすべてを decision して良いかどうかは疑問であり、むしろ、専門家が decision を下す場合に、エキスパート・システムの意見を参考にするとといったレベルが現実的である。これは、診断型エキスパート・システム固有の性格であり、たとえば制御の分野では、(1)のレベルのエキスパート・システムが要求されることは言うまでもない。

3. 実用化する上での制約事項

前章では、診断型エキスパート・システムの現状について述べたが、実用化レベルでとらえると、技術的には次のような点に問題が存在する。

(1) 適用範囲

従来の診断型エキスパート・システムの多くは、観察事象と故障原因の関係を知識ベースに貯え、推論機構が与えられた情報から故障原因を検出するという方式である。この方式においては観察事象と故障原因の関係が定義されていない事象に対しては全く対処できない場合がある。いわゆる専門家(オペレータ)の持つヒューリスティックな知識や教科書(マニュアル)的な知識は「浅い知識」と呼ばれ、そのような知識のみの場合には、エキスパート・システムの適用範囲を

限定化してしまうこともある。

(2) 知識表現

現在、実用に近い診断型エキスパート・システムはプロダクション・ルール型が中心であり、フレーム型のものもいくつかあるが、多くは単一表現である。また、これらのシステムが扱っている問題は比較的単純であり、知識ベースの規模も数百ルール程度であり、特に複雑な知識表現を必要としないことも事実である。しかし、対象とする問題領域が複雑になると、モデル記述力を高める必要が出てくる。また、診断型エキスパート・システムが扱う対象の持つ特殊性から、あいまい性、時間依存性の記述が必要となってくる場合があるが、これらに対する知識表現モデルは確立していない。

(3) 推論メカニズム

多くのエキスパート・システムの推論メカニズムは単調論理に基づいている。すなわち、知識ベースに格納されている知識はすべて正しいことを前提としている。しかし、診断の対象システムが時間経過によって変化する場合にはこの前提を覆す論理体系が必要である。また推論メカニズムは知識表現に強く依存しているが、知識表現自体が複雑化しているのに対する対応が遅れている。

(4) 説明機能

説明機能はそのエキスパート・システムが実用レベルに近いほど、重要な機能になってくる。特に前章で述べたように診断型エキスパート・システムの利用法としては参考意見を出させる形態になるため、ほかの種類エキスパート・システムと比較してその結論に対する説明機能が充実していることが要求される。しかし、多くのシステムの説明機能はその結論に至った

過程を示すような単純な機能にとどまっている。

(5) システム構築方法論

現在のエキスパート・システム作りは、ごく限られた専門家と、知識工学エンジニアの間で行われている。いわゆる家内工業的な手作りによる開発が大多数であり、ソフトウェア開発における1960年代を彷彿させる。したがって、ソフトウェア開発におけるソフトウェア危機と同等の問題が近い将来、生ずることも予想されるが、その最大の要因はシステム構築方法論の不備である。従来的一般システムやソフトウェア開発における構築方法論が踏襲できるのか、あるいは全く新しい方法論が必要なのかの議論も不足している。

(6) 知識獲得

知識獲得とは、エキスパート・システム開発の観点から次の2点でとらえることができる。

(i) システム開発の初期段階における専門家の知識の体系化および初期入力

(ii) ある程度知識ベースが構築された後の段階的な知識の追加、更新

これらのうち特に(i)については、システム構築方法論の問題の一部としてもとらえることができるが構築方法論の不備にともない、手法は確立していない。

(ii)については、学習型手法が数多く提案されている²⁸⁾が、実用に至ったシステムはきわめて少ない。したがって知識獲得はエキスパート・システムの構築上のボトルネックとも言われている。

(7) 診断型シェル

一般に商用化されているシェルの多くは、汎用型であるが、汎用型のシェルのみではいかば議論を呼ぶところである。少なくとも診断型シェルとして用いる場合に特に必要な機能には、①非単調推論、②高度な説明機能、③不確実性の表現、推論機能などがあるが、これらを実装したシェルは商品化の段階には至っていない。

4. 実用化へ寄与する技術

国内外の大学や研究機関では前章で述べた問題点を克服すべく提案が活発に行われている。ここでは、これらの提案の中で実用化に有効と考えられるものをいくつか選んで紹介する。

(1) 適用範囲

前章において、「浅い知識」のみではその知識ベースの適用範囲は限られていると述べたが、それに対す

る解決策として「深い知識」の有効性が広く強調されている。「深い知識」とは「浅い知識」が専門家のヒューリスティックスに対応していたのに対して、対象モデルの機能や構成に基づいた知識を意味している。「深い知識」を用いる利点は、過去に発生しなかった故障に対しても対処可能になること、システム開発時に知識ベースを構築できることである。

ここでは「深い知識」に基づいた問題解決手法として、定性的な式を用いた推論⁹⁾および物理法則に基づいた推論^{7),16)}の二つを取り上げる。

(i) 定性的な式を用いた推論

de Kleerらが提案している定性的な式を用いた診断とは、診断対象を一連の定性的な式を用いてモデル化し、その式を満足しているか否かを調べることにより、診断を行おうとするものである⁸⁾。定性的な式は診断対象が正常なときに満たされなければならない制約条件である。また、人間は定量的な微分方程式を解くことはせず、定性的な特徴をつかまえてシステムや機器の動作を推論、理解していることから、本方式を用いた診断は人間の行っている思考プロセスに近いと考えられる。定性的な推論の問題としては、あいまい性と不連続性が許されていない点があげられる。また、通常のシステムの多くはフィードバック・ループを含むが、基本的な手法では適用しにくい。実用に供するためにフィードバック・ループを含むシステム向けへの拡張に関して多くの提案がなされている。たとえばY. Iwasaki^{24),25)}はCausal Orderingの理論に加えて、Comparative Staticsを用いており、de Kleer⁸⁾はMythical Causalityの概念の基にヒューリスティックスを導入している³⁷⁾。さらに、西田³⁸⁾は、因果関係を直接表すデータ構造を利用した解析、変数の不連続変化の体系的な取り扱い、不完全な情報の管理を可能とする動的因果関係解析手法を開発している。

(ii) 物理法則に基づいた推論

専門家による診断は、観察される事象から故障原因を過去の経験に基づいて同定するといった方法のほか、診断対象の構造・機能に関する知識を用いて行われる場合も多い。機能に関する知識は診断対象が満たされなければならない制約条件を定義したものと考えられる。また、システムの構造に関する知識を用いれば、どの要素が故障している可能性があるかが推論でき、各要素の制約条件から診断を行うことができる。このような構造・機能に関する知識は診断対象となるシステムの設計者の知識であり、一方、観察事象と故

障原因に関する知識は、そのシステムの運用者の知識ともいえる。また、物理法則に基づいた推論に関する研究を進めている R. Davis は論理回路の構成と振る舞いを表現している⁷⁾。振る舞いに関しては構成要素の入出力間の関係に着目して、入力から出力を得るためのシミュレーションルールと逆に出力から入力を推定するためのインフェレンスルールを用いている。他の代表例としては、M. Genesereth¹⁶⁾ が論理回路の診断システム DART を開発しており、また国内においても、元田³⁴⁾ は原子炉異常診断において、システムの機能に関する知識も利用している。

論理回路のようにインフェレンスルールが簡単に得られるものは良いが、一般の装置では簡単にルールとして記述できないものが多く、今後の課題である。また、機能や構成に関する知識のみの場合には、探索空間が膨大であること、冗長な診断ルールを生成する可能性があることなどの問題点があり、実用に供するためにはこれらの点を改善することが必要である。

(2) 知識表現

高度で複雑な問題を記述するために、種々のアプローチがある。大きく分類すると次の二つがある。

① フレームとルールを組み合わせるなどの複合型の知識表現

② あいまい性や時間依存性を記述する知識表現

①についてはマルチパラダイム型知識表現ともいわれ、事例としては研究レベルのものも含めると LOOPS⁴⁷⁾, ZERO²¹⁾, Mandala¹⁵⁾, MENDEL¹⁷⁾, CL⁵⁴⁾, ORIENT/84⁵¹⁾, ABCL⁵⁹⁾, CIL³³⁾ などがあるが、いずれのモデルの記述力が高いのか、また、真に対象システムに近いモデルはどれかを決定する基準、診断型には何が良いかの選択基準を構築することが望まれる。

②については医療診断に限定しているが、文献²²⁾に詳細に解説されている。あいまい性に関しては MYCIN 流の CF (確信度) の計算によってサポートしているシステム (商用シェル) が大部分であるが、CF のみであいまい性の表現が可能かどうかは疑問である。また MYCIN 流の計算手法においてもその計算値の妥当性は理論的には保障されていない。これに対して Dempster-Shafer 理論^{23), 45)} による確信度表現の応用への関心が高い。さらに、この Dempster-Shafer 理論に基づいた推論手法として evidential reasoning²¹⁾ が提案されている。診断型エキスパート・システムにおける evidential reasoning の位置付け

は、対象システムのセンサからの不完全で不確実な信号から理論的に妥当な確信度の付いた結論を導出することである。ところで、Dempster-Shafer 理論では、基本的には各仮説は独立であることを前提としているが、現実の対象システムにおいては、仮説間に因果関係が存在する場合が多い。このようないわゆる複合仮説の取り扱いが可能となるようにするためにいくつかの試みが実施されている^{40), 58)}。たとえば大和田、溝口⁴⁰⁾ は、時間論理に基づく仮説間の因果関係を付加して複合仮説を扱えるようにしている。

また、論理に基づく知識表現において時制の概念を組み込む場合には様相論理の一つである時制論理を採用することがごく自然な形態である。しかし、時制論理のなかで最も記述力の低い時制命題論理においても、その決定手続きの計算量が PSPACE 完全であることが知られている⁵⁶⁾。したがって時制論理表現を用いた場合にはシミュレーション手法が妥当であると思われる。すなわち、時制命題論理で考えるならば、そのオペレータとして

○ f (次の状態で f が真である)

◇ f (将来 f が真である)

$f_1 U f_2$ (f_2 が真になるまで常に f_1 が真である)

□ f (常に f が真である)

があるが、これらのオペレータをそのまま解釈実行することを意味する。最近の知識表現としての時制論理に関する研究対象としては、Allen の ITL (Interval Temporal Logic)¹⁾ をベースとした区間時間体系時制論理に集中している^{29), 30), 53)} が、診断型のエキスパート・システムの記述に対しても有効であると思われる。

(3) 推論メカニズム

オンライン診断を行う診断型エキスパート・システムにおいてはオンラインで新しいデータが入力されてくる。したがってそのデータに反応して迅速に推論の方向を修正することが要求される。この要求に対応するために ATMS (Assumption based Truth Maintenance System)¹⁰⁾ は有効である。もっとも知識ベースを外部の状況変化に反映させながら一貫性を保持するための機能は TMS (Truth Maintenance System)¹¹⁾ においても実現されているが、すべての正当化を満たす解を見つけるのに非常に多くの資源を費やすのでオンライン診断への適用は現実的ではない。一方、ATMS では知識ベースすべてに対しての無矛盾性を追及しているわけではなく、また正当化のための負荷も軽くな

っているので TMS より、一步実用レベルに近い。また、ATMS はすべての可能解を生成しているが、適用分野によってはその必要もない。そのためにバックトラック機能を ATMS に組み込むことも提案されている⁹⁾。

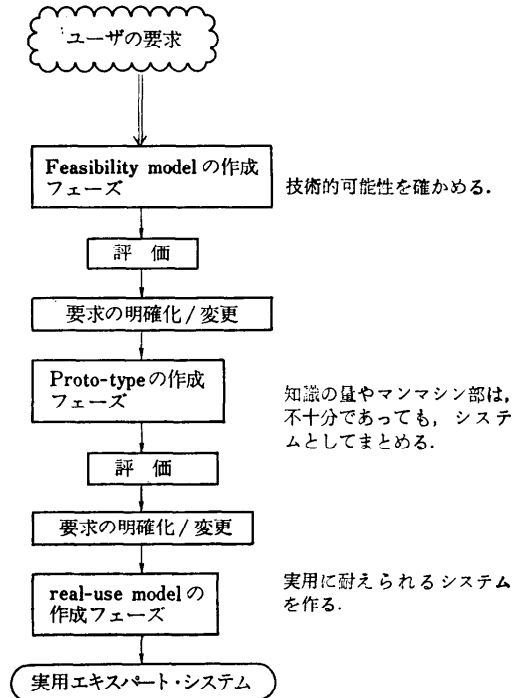
(4) 説明機能

深い説明機能を実現するためには、いわゆる診断に用いられるルール・ベース以外に、メタ知識や深い知識などと呼ばれる知識およびそれらを管理する機構が必要である。深い説明機能を有するシステムとして XPLAIN がある⁴⁹⁾。XPLAIN はジギタリス療法のためのエキスパート・システムであるが、深い知識として、ジギタリス療法における因果関係と手続き的概念を持っている。XPLAIN では一般の診断知識に加えて、これらの深い知識を結び付けることにより、ユーザに対する深い説明機能を実現している。一般により深い説明機能を実現するためには対象システムの設計知識が有効であるが、XPLAIN の後継システムである EES では、設計知識としてはそのシステムの開発過程を利用している⁴⁵⁾。

(5) システム構築方法論

ここでは著者らが開発中のエキスパート・システム構築方法論 TUPPS-ES を紹介する。エキスパート・システムといえども一つのシステムであり、一般のシステム構築方法論が生かされるべきである。しかし、さらに知識工学、エキスパート・システムという特殊性を反映した方法論として具体化しておく必要がある。一般的なシステム構築ではフェーズド・アプローチが採られており、それが基本となる。しかしながら知識工学手法の特徴としてプロトタイプング手法が容易に導入できる。そのため、図-1 に示すように3フェーズが考えられている。

第1のフェーズの Feasibility model の作成フェーズでは、漠然としたユーザの要求からシステムの果たすべき目標を明確化し、技術的な実現可能性を検討する。ユーザの要求は不明確で矛盾を含んでおり、断片的なものであるのが通常である。また、エキスパート・システムに過度の期待を持っている場合が多く、ユーザに妥当な実現イメージを持たせる必要がある。従来手法によるシステムではなく、エキスパート・システム化することの意味も十分つめておかなければならない。また必要であれば試験的に知識の抽出を行い、計算機にインプリメントし、技術的可能性をチェックする。その結果を評価し、最初のユーザの要求を



明確化するとともに、実現の無理な部分は変更する。

第2フェーズの Proto-type の作成フェーズでは、前フェーズでの結果を基に、実際にエキスパート・システムとして機能するプログラムを作成する。知識の量、性能、マンマシン機能などは実用レベルに達していなくてもよい。技術的に核となる知識、推論機能がともかく実現されておればよい。このプロトタイプを評価し、ユーザの要求を一層明確化し、より妥当な仕様に変更する。

第3フェーズの Real-use model の作成フェーズでは実用に耐えられるエキスパート・システムを作成する。知識の量は実際に使用しながら増していくが、エンドユーザが現場で使えるシステムとする必要がある。

各フェーズでは、図-2 に示すステップにそってエキスパート・システム構築を行う。

(6) 知識獲得

知識獲得については種々の提案があるが、ここでは図-2 に示す知識獲得ステップに基づいて述べる。

① 初期の知識獲得

診断型エキスパートに関しては、深い知識（システムの構造化）の獲得が重要であるが、それに関する研

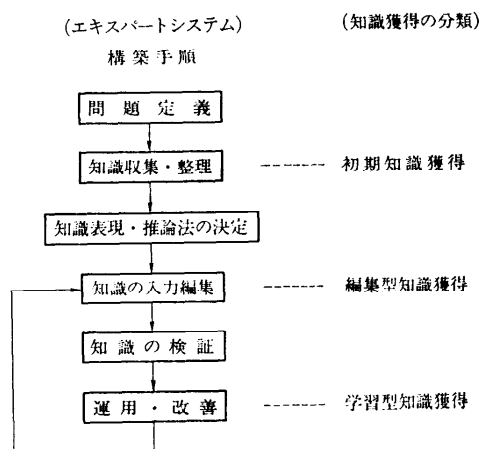


図-2 エキスパート・システムの知識獲得ステップ

究はきわめて少ない。その解決案の一つとしては、システム工学の適用が考えられるが、エキスパート・システム構築方法論 TUPPS-ES においても診断型エキスパート・システムにおける初期の深い知識獲得にシステム工学手法を用いている。初期の知識獲得の特徴として、エキスパートシステム構築者が診断対象システムに対してあまり明確なモデルを持っていないことがあげられる。そのためインタビューの手法の重要性が認識されているが、より体系的な技術とするためには、システム計画時に採用される各種の手法が一つの方向付けを与えると考えられる。一般のシステム構築技術では、システムのイメージが明確ではないシステム計画時点においては、システムの機能構造を明確にする手法が探られる。これは目的を持ったシステム、すなわちユーザの目的に合った特殊システムをより汎用的な部分システムに分解することにより、システム構築者の言葉で表現することであり、“What”を“How”で表現する深い知識獲得の第一段階である。この手法は一種の再帰的な手法となっており、システムを階層的にブレークダウンする手段を与えている。したがって、初期の知識獲得において、エキスパートシステム構築者または専門家自身が、診断対象システムに関する深い知識を得るための一つの表現法を与えていると言える。

システムを表現する場合、内包的表現と外延的表現がある。前者はトップダウン的記述、後者はボトムアップ的記述に対応する。また、これだけではなく、観点の相違によって表現/記述の仕方は変わってくる。したがって、実際にシステムを記述する場合には、さ

まざまなレベル、観点の記述が混在している。しかしながら、これらの記述は対象システムのさまざまな面(構造)を表しているわけで、逆にこのことを利用してシステムの構造(深い知識)を分析することが可能となる。このときに供される技術は、システム構造化技法と呼ばれている。代表的な例としては、システムをさまざまな言葉で表現し、その言葉の相関関係を分析するものであり、二種類ある^{2), 39)}。

(i) 自己相関関係

これは、言葉の集合をシステムの表現と考え、その言葉間にある関係を定義するものである。

x : 言葉 (Term)

s : システム s を表現する述語

として、

$$s = \{x | s(x)\}$$

であるとき、 x, y, s 間に二項関係 R を考え推移律を仮定するものである。すなわち、

$$R(x, y), R(y, z) \rightarrow R(x, z)$$

s を構造化するには、この推移律を使った同値関係(グラフ論的には、強連結成分)でグループ化を行う。しかしながら当然のこととして実際的には関係 R の定義が明確でなく、推移律が一般的には成立しない場合が多い。そのため、 R を多値化して利用する機会が多い。また R は一意的でなく、いくつもの関係が考えられるため、構造化の一つの例として考えた方がよい。

(ii) 相互相関関係

これは、一つのシステム s を二つの観点 X, Y から記述し、それらの記述間の関係の全体をシステム s を表す概念と考える考え方である。すなわち、一つのシステムを記述する方法を二つ考え、それらを結ぶ共通概念がシステムを表現すると考える。

$$s = \{R | R(x, y), x \in X, y \in Y\}$$

グラフ論的には二部グラフに対応し、システムは

$$(X_1, Y_1): \forall R \exists x_1 \in X \exists y_1 \in Y$$

$$\exists x \in X_1 \exists y \in Y_1$$

$$R(x, y_1) \vee R(x_1, y)$$

で表現されるとする。 $(\phi, Y), (X, \phi)$ もシステムの表現であり (X_1, Y_1) も一つの表現である(図-3)。この手法の特徴は、システム自体というよりはシステムの表現の構造化に主眼を置いてあり、上記のさまざまな表現に制約を課し構造を抽出しようとするものである。特に X, Y の大きさ $|X|, |Y|$ に制約を置き $|X| + |Y|$ が最小となるように構造化を行うと、二部グラフの既約表現になり、 X, Y がグループ化される。この方法

によれば、 X としてシステムの目的、 Y としてそれを達成するための機能とすることにより、直接システムの機能分解が可能となる。相互相関関係を使ったほかの構造化手法としては、一方の観点の言葉を、他の観点の言葉との関係を使って多次元空間内の平面に対応づけ位相空間として解析する手法も研究されている。

このように機能分解された結果を深い知識として使用するためには、これにデータ、コントロールの情報を追加する必要がある。さらに深いものにするには、システム構成要素との対応づけが必要となる。TUPPS-ESでは、これらを支援するツール群を用意しており、設計情報から深い知識を生成することを支援している。一つの例として図-4にPrologによる深い知識の表現例を示す。この結果一つのシステムの深い知識をあるレベルでモデル化することが可能となる。

診断型エキスパートシステムには、上記のモデルを使った推論機能が必要であり、モデルのインプリメンテーション方法とともに、さらに研究されるべきであると考えている。

② 段階的な知識獲得

診断型に限定された段階的な知識獲得支援システムの研究が確実に進んでいる^{19),20)}。この研究はTEIRESIAS⁶⁾が原点ともなっており、それを基礎として研究が行われている。SEEKでは「ルールの一般化」「ルールの特殊化」を用いて、知識ベースの性能を向上させている⁴⁾。MOREでは、知識ベースの洗練化において次の3ステップを支援している²⁶⁾。

1. 専門家とのインタビューをとおして知識ベースを作る。この際の個々の知識は仮説と徴候および条件間の関連を記述したネットワークとして構築される。
2. 8つの戦略を用いて洗練化する。
3. 確信度の洗練化をする。

MOREでは知識ベースのネットワークの静的解析によってどの知識が欠け

contain (f_a, f₁).
 contain (f_a, f₂).
 contain (f_a, f₃).
 receive (f₁, d₁).
 receive (f₁, d₂).
 receive (f₁, d₃).
 generate (f₁, d₂).
 generate (f₁, i₂).
 generate (f₁, i₃).
 consist (i₃, d₃).
 dfcd (DF, I, O) : -contain (DF, F), dfunction (F, I, O).
 dfunction (F, I, O) : -receive (F, I), generate (F, O).
 dfunction (F, I, O) : -receive (F, I), generate (F, OX), dfunction (FX, OX, O).
 receive (F, I) : -consist (I, IX), receive (F, IX).
 generate (F, O) : -consist (O, OX), generate (F, OX).

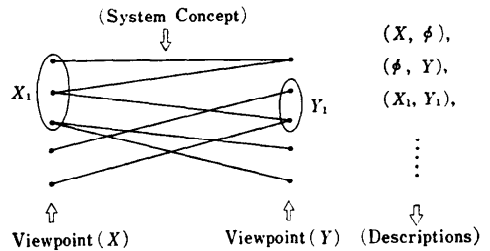
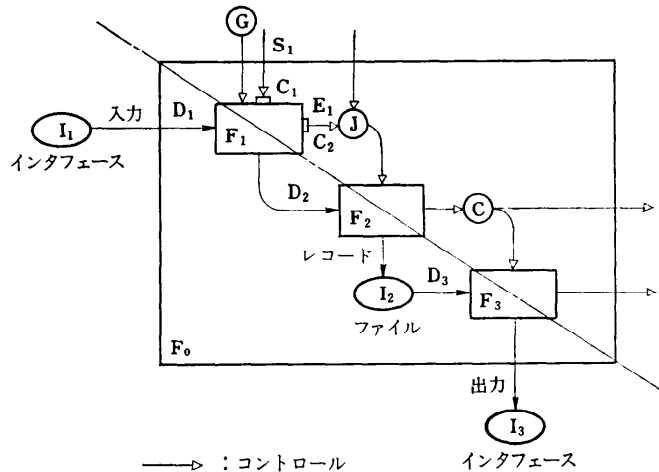


図-3 二部グラフによるシステム構造化



- ▷ : コントロール
- : データ
- : 機能
- : データセット
- : コントロールノード
- P : PARALLEL BEGIN (FORX)
- J : JOIN
- C : CONDITIONAL SELECT
- G : GENERATE CONTROL PERIODIC

図-4 Prologによる深い知識の表現例

ているかを検出する。したがってある程度、知識が入力された段階で不完全な知識を正しくするのに適している。しかし、初期入力にはあまり適しているとは言えないのが MORE の欠点である。MORE の発展形である MOLE では初期入力をも可能としている¹³⁾。そのため、対象となる世界や専門家から引き出した知識のあいまい性をなくすために専門家がどのように表現するかの知識を有している。したがって質問回数を減らすことが可能となる。いずれにせよ、これらのシステムはプロダクション・ルール型の知識獲得ツールであるが、知識表現の多様化への対応も今後要求される。一方、例題から、ルールを生成する帰納推論を用いた学習型知識獲得としては E. Y. Shapiro のモデル推論アルゴリズムがある⁴⁶⁾が、実用レベルで用いるためには計算量の問題を解決しなければならない。ごく単純な帰納推論を用いて知識獲得を実現した商品としては、EX-TRAN がある³²⁾。EX-TRAN では例題からルールとして Fortran プログラムを生成する。

(7) 診断型シェル

診断型の専用ツールとして商品化されている数少ないツールの一つとして IN-ATE がある⁴³⁾。IN-ATE の主要な特徴は①知識ベースの自動生成、②Shafer-Dempster 理論に基づいた推論、③最適テストの選択などである。①については論理モデルなどから帰納推論を用いて診断ルールを自動生成するものであり、④についてはトラブル・シューティングの過程で最良のテストを選択していくもので、ガンマ最小平均発見的探索手法や Shannon の情報理論を用いて実現している。ほかの例としては CSRL がある³⁾。CSRL は B. Chandrasekaran の提案しているエキスパート・システム構築方法論“Generic Tasks”⁶⁾の考え方に基づいた診断型（診断を分類作業ととらえている）シェルである。CSRL では知識は specialist と呼ばれるモジュールによる階層的な構成になっており、対象システムの知識表現が自然にできる。この適用事例としては医療診断システム MDX⁴⁾などがある。深い知識を利用して診断型、設計型を統合化するエキスパート・システムシェルの研究も行われている。その中では設計段階に得られる知識から「知識コンパイル手法」によって冗長性の少ない診断ルールを生成する手法が提案されている⁵⁷⁾。

5. むすび

本稿では、診断型エキスパート・システムの現状およびその問題点を明確にし、その解決案について述べた。ここで述べた解決案の中には診断型に必ずしも限定しないものも多いが、いずれにせよ、エキスパート・システムが実用化されることにより、多くのユーザの手に触れ多様な要求が生まれ、それらが研究者へフィードバックされるといったサイクルが今後とも続くであろう。

産業界において、エキスパート・システムへの潜在的ニーズはかなり高いと思われるが、これらのエキスパート・システムが開発されるためにも、知識獲得手法を含む構築方法論の確立およびそれに基づいた適切なシェルの早急な整備が望まれる。“Generic Tasks”⁶⁾の考え方は一つの方向を示唆している。

ここで述べた問題点以外にも一般のソフトウェア・システムとしての観点からとらえると、さらに

- ① ソフトウェア（知識ベース）の移植性
- ② エキスパート・システムの応答性
- ③ マンマシン・インタフェース
- ④ エキスパート・システムの開発環境

など多くの課題が残されている。おのおのの課題において一般のソフトウェア・システムの範疇では論じることのできないエキスパート・システム固有の問題点を洗い出し対応策を検討することが要求されてくるだろう。特に①に関してはシェルの汎用状態である状況から今後、大きな問題になることも予想される。

参 考 文 献

- 1) Allen, J. F.: Maintaining Knowledge about Temporal Intervals, *Comm. ACM*, Vol. 26, No. 11, pp. 832-843 (1983).
- 2) Arai, M. et al.: A Method for Structural Modeling of Complex Systems, *Proc. IEEE Int'l. Conf. on SMC* (1981).
- 3) Bylander, T. G. and Mittal, S.: CSRL: A Language for Classificatory Problem Solving and Uncertainty Handling, *AI Magazine*, Vol. 7, No. 3 (1986).
- 4) Chandrasekaran, B. et al.: An Approach to Medical Diagnosis Based on Conceptual Structures, 6th *IJCAI* (1979).
- 5) Chandrasekaran, B.: Generic Tasks in Knowledge-Based Reasoning: High-Level Building Blocks for Expert System Design, *IEEE EXPERT*, Vol. 1, No. 3 (1986).

- 6) Davis, R. and Lenat, D.: Knowledge-Based Systems in Artificial Intelligence, McGraw-Hill (1982).
- 7) Davis, R.: Diagnostic Reasoning Based on Structure and Behavior, Artificial Intelligence, Vol. 24 (1984).
- 8) de Kleer, J. et al.: A Qualitative Physics Based on Confluences, Artificial Intelligence, Vol. 24, pp. 7-83 (1984).
- 9) de Kleer, J. et al.: Back to Backtracking: Controlling the ATMS, Proc. of AAAI-86, pp. 910-917 (1986).
- 10) de Kleer, J.: An Assumption-based Truth Maintenance System, Artificial Intelligence, Vol. 28 (1986).
- 11) Doyle, J.: A Truth Maintenance System, Artificial Intelligence, Vol. 24 (1979).
- 12) Ernst, M. L. and Ojha, H. E.: Business Applications of Artificial Intelligence-Knowledge-Based Systems, ADL Research Letter October 1986/L861001 (1986).
- 13) Eshelman, L. and MacDermott, J.: MOLE: A Knowledge Acquisition Tool that Uses its Head, Proc. of AAAI-86, pp. 950-955 (1986).
- 14) Expert Systems Strategies, Vol. 2, No. 8, Cutter Information Corp. (1986).
- 15) Furukawa, K. et al.: Mandala: A Logic Based Knowledge Programming System, FGCS '84 (1984).
- 16) Genesereth, M.: The Use of Design Descriptions in Automated Diagnosis, Artificial Intelligence, Vol. 24, pp. 411-436 (1984).
- 17) 本位田, 内平他: 推論型システム記述言語 MENDEL, 情報処理学会論文誌, Vol. 27, No. 2, (1986).
- 18) 本位田, 内平他: MENDEL による 計算機 OS 性能診断エキスパート・システム, 電気学会, システム・制御研究会 SC-86-15 (1986).
- 19) 昭和60年度新世代コンピュータに関する技術開発動向及び適用分野等の調査研究報告書—発電設備診断システム—(財)新世代コンピュータ技術開発機構 (1986).
- 20) 昭和60年度新世代コンピュータに関する技術開発動向及び適用分野等の調査研究報告書—エキスパート・システムにおける知識獲得の様相—(財)新世代コンピュータ技術開発機構 (1986).
- 21) 伊藤, 上野: ZERO: Frame+Prolog. Proc. of the Logic Programming Conference (1985).
- 22) 伊藤他: 医療診断エキスパート・システムにおける知識表現, 情報処理, Vol. 27, No. 7 (1986).
- 23) 石塚: Dempster & Shafer の確率理論, 電子通信学会誌, Vol. 66, No. 9 (1983).
- 24) Iwasaki, Y. et al.: Causality in Device Behavior, Artificial Intelligence, Vol. 29, No. 1 (1986).
- 25) Iwasaki, Y. et al.: Theories of Causal Ordering: Reply to de Kleer and Brown, Artificial Intelligence, Vol. 29, No. 1 (1986).
- 26) Kahn, G., Nowlan, S. and MacDermott, J.: MORE: An Intelligent Knowledge Acquisition Tool, Proc. 9th IJCAI (1985).
- 27) 川北: 知識工学の産業システムへの応用, システムと制御, Vol. 30, No. 5, pp. 289-295 (1986).
- 28) 國藤, 古川: 学習システム研究の現状と課題, 計測と制御, Vol. 25, No. 9 (1986).
- 29) Ladkin, P.: Primitives and Units for Time Specification, Proc. AAAI-86, pp. 354-359 (1986).
- 30) Leban, B.: A Representation for Collections of Temporal Intervals, Proc. AAAI-86, pp. 367-371 (1986).
- 31) Lowrance, J. D. et al.: A Framework for Evidential Reasoning Systems, Proc. AAAI-86, pp. 896-903 (1986).
- 32) Michie, D.: The Superarticulacy Phenomenon in the Context of Software Manufacture, Proc. R. Soc. Lond. A405, pp. 185-212 (1986).
- 33) Mukai, K.: Unification over Complex Interminates in Prolog, Proc. of LPC '85, pp. 271-278 (1985).
- 34) 元田他: システムの機能に関する知識を用いた原子炉異常診断の試み, 計測と制御, Vol. 22, No. 9, pp. 791-796 (1983).
- 35) Neches, R. et al.: Explainable (and Maintainable) Expert System, 9th IJCAI, pp. 382-389 (1985).
- 36) 日経 AI, No. 1-No. 21 (1986).
- 37) 西田: 定性的推論, 第49回知識工学システム分科会 (1986).
- 38) 西田他: 動的因果関係解析法による電子回路の定性的解析, 情報処理学会, 知識工学と人工知能, 48-7 (1986).
- 39) 大内他: 擬順序関係の含意を求める効率的アルゴリズム, 電気学会論文誌C, 61-C12 (1986).
- 40) 大和田, 溝口: 因果推論を導入した複合仮説の診断, Proc. of LPC '86, pp. 35-42 (1986).
- 41) Politeks, P. et al.: Using Empirical Analysis to Refine Expert System Knowledge Bases, Artificial Intelligence, Vol. 22, pp. 23-48 (1984).
- 42) Pople, H.: The Formation of Composite Hypothesis in Diagnostic Problem Solving—an Exercise in Synthetic Reasoning, 6th IJCAI (1979).
- 43) Richard, R. et al.: IN-ATE: Fault Diagnosis as Expert System Guided Search, Automated Reasoning Corporation (1985).
- 44) 佐々木, 嶋田: 知識工学の産業への応用, 人工知能学会誌, Vol. 1, No. 1, pp. 64-71 (1986).
- 45) Shafer, G.: A Mathematical Theory of Evi-

- dence, Princeton Univ. Press (1976).
- 46) Shapiro, E. Y.: Inductive Inference of Theories from Facts, Yale Univ. Res. Rept. 192 (1981).
 - 47) Stefik, M. et al.: Knowledge Programming in LOOPS: Report on a Experimental Course, AI Magazine, 4, 3 (1983).
 - 48) Shortliffe, E. H.: Computer Based Medical Consultations: MYCIN, American Elsevier (1976).
 - 49) Swartout, W. R.: XPLAIN: A System for Creating and Explaining Expert Consulting Programs, Artificial Intelligence, Vol. 21, No. 3 (1983).
 - 50) Szolovits, P. and Pauker, S. G.: Categorical and Probabilistic Reasoning in Medical Diagnosis, Artificial Intelligence, Vol. 11 (1978).
 - 51) Tokoro, M. et al.: An Object-Oriented Approach to Knowledge Systems, Proc. of FGCS '84, pp. 623-631 (1984).
 - 52) 上野: エキスパート・システム—研究動向と技術的課題—, 人工知能学会誌, Vol. 1, No. 1, pp. 48-56 (1986).
 - 53) Vilain, M.: Constraint Propagation Algorithms for Temporal Reasoning, Proc. AAAI-86, pp. 377-382 (1986).
 - 54) 渡辺: CL: ドメイン言語構築機能を強化したフレーム型知識表現システム, 情報処理学会, 知識表現と人工知能研究会資料 40-2 (1985).
 - 55) Weiss, S. et al.: A Model-Based Method for Computer Aided Medical Decision Making, Artificial Intelligence, Vol. 11 (1979).
 - 56) Wolper, P. L.: Synthesis of Communicating Processes from Temporal Logic Specifications, STAN-CS-82-925 (1982).
 - 57) 山口他: 設計と診断を融合したシェルの構成, 情報処理学会第33回全国大会, 7K-1~3 (1986).
 - 58) Yen, J.: A Reasoning Model Based on an Extended Dempster-Shafer Theory, Proc. of AAAI-86, pp. 125-131 (1986).
 - 59) 米澤他: オブジェクト指向に基づく並列情報処理モデル ABCM/1 とその記述言語 ABCL/1, コンピュータ・ソフトウェア, 3, 3 (1986).

(昭和61年11月6日受付)