OUKA
The University of Osaka
Institutional Knowledge Archive

Title	知識工学概論
Author(s)	豊田, 順一
Citation	大阪大学大型計算機センターニュース. 1985, 58, p. 25-34
Version Type	VoR
URL	https://hdl.handle.net/11094/65657
rights	
Note	

The University of Osaka Institutional Knowledge Archive : OUKA

https://ir.library.osaka-u.ac.jp/

The University of Osaka

知　識　工　学　概　論

大阪大学産業科学研究所　豊　田　順　一

0. はじめに

本小論は、古典的人工知能研究より始めて知識工学の誕生、知識工学的問題解決、及び人工知能の応用例であるエキスパートシステムについて述べたものである。

1. 人工知能研究とは

人工知能の研究は、ある時点で知的であると判断されるような仕事を計算機で実現することを目標としている。したがって、人工知能研究が始められた頃（1960年代初頭）と現在とではその内容も大いに異なってきている。1959年、あの有名なオンラインコンピュータPDP-1 がDECより発売された（$120,000）当時、他の計算機会社が発売しているコンピュータはいずれも百万ドル以上であった。1963年になってやっとミニコンのベストセラー PDP-8が世に出たのであった。現在私達が手にしているパソコンと比較してみると隔世の感があろう。しかし、計算機本体などのハードウェアが百倍も千倍も進歩したように人工知能研究が進歩してきたわけではない。ある時は、その成果に失望し研究者が激減したこともあった。以下に人工知能研究の課題のいくつかを掲げておく。

- ゲーム（チェスやチェッカーをするプログラムの作成）
- 定理証明
- 一般問題解決
- 画像理解
- 自然言語理解
- 音声理解
- 応用人工知能——エキスパートシステム

2. 古典的人工知能研究と知識工学の誕生

古典的人工知能研究の多くは、人間が作った問題の表現、例えば状態記述に作用素を操作して、新しい状態記述を作り出すといったものであり、計算機の高速性を単に利用しているに過ぎなかった。しかもその研究方法は、階層構造の一番下の部分から積み上げて行こうとするものであり、問題の特性から来る制御の枠組を考慮していなかったため、処理手数の組合せ的爆発には対処できなかった。Toy-problemに属する簡単な問題のみが扱われたのも当然であった。もちろん、ヒューリスティックス（経験則）の導入も考慮されたが、極めて単純な常識を

前提とするに留まった。しかも、いくつもの知識をT.P.O.に従って適用するというのではなく、唯一つの知識をかたくなに適用していた。

初期のこのような人工知能研究の結果、人間の持つ能力を活用し、それらを合成するという観点からの研究の必要性がはっきりと認識されてきた。すなわち、人間の持っている専門知識、常識の活用が人工知能には不可欠であることが判明し、ここに人間の知識を有効に利用することを目指す知識工学が誕生したのである。

3. 知識工学の主要研究課題

知識工学の主要な研究課題には、人間の持つ専門的知識・常識をどのように計算機上にて表現するかという知識表現の問題、一つの知識から別の知識を導く推論の問題、知識の獲得・理解・学習の問題、及び計算機と人間の対話インタフェイスの問題などがある。以下、これらについて簡単に説明していく。

3.1 知識の表現

知識は、計算機で操作される形式で表現されていなくては、工学的に何の意味もない。本節では、知識の表現方法について述べる。

知識表現の大まかな分類として、手続き的知識表現と宣言的知識表現が考えられる。まず始めに手続き的知識表現の例を掲げておく。この例では、冷蔵庫（REIZOHKO）にEGG, MILK, BROTCHEN , BACON, LETTUCE, TOMATO, ICECREAM, SCHINKENが入っていることを（EGG MILK BROTCHEN BACON LETTUCE TOMATO ICECREAM SCHINKEN）と表わす。誰かが冷蔵庫の中の何かを食べた時、何と何が残るかということを管理するための知識は以下のようにLISPプログラム中に手続きとして埋め込むことができる。

```
(NOKORI (LAMBDA (A REIZOHKO)
 (COND  ((NULL REIZOHKO) ( ))
        ((EQ (CAR REIZOHKO) A) (CDR REIZOHKO))
        (T (CONS (CAR REIZOHKO) (NOKORI A (CDR REIZOHKO))))
        ) ))
```

先程の内容が入っている冷蔵庫からICECREAMを取り出すと何が残るかを知るためには、(NOKORI 'ICECREAM REIZOHKO)を実行すればよく、(EGG……TOMATO SCHINKEN)が得られる。このような知識表現形式は、動作や事象の関係を手続きとして記述するのに便利である。

次の例は、1平方マイル当りの人口密度を求めた上で、±5％の範囲で同一人口密度の国を

求めるPrologプログラムである。プログラム中では、いくつかの知識が宣言的に述べられている。

```
density (C,Dis) :-pop (C,P) , area (C,A) , Dis is (P*1000)/A.
answer (Cl, D1, C2, D2) :-density (Cl, D1) , density (C2, D2) , D1>D2, 20*D1<
21*D2.
pop (china, 825) .                    area (china, 3380) .
pop (india, 586) .                    area (india, 1139) .
pop (ussr, 252) .                     area (ussr, 8709) .
pop (usa, 212) .                      area (usa, 3609) .
```

このような論理的（一階述語論理）な形式には、数理的演繹法を即適用することが可能である。Prolog上で知識を表現すると、Prologが持つ推論機構を利用することができる。このような宣言的知識表現は、一般に知識の追加・修正が容易であり、表現構造が単純で理解しやすいという特徴がある。

宣言的表現と手続的表現の中間に位置する表現にプロダクションルールによる知識の表現がある。プロダクションルールでは「もし～ならば……である（……する）」という条件と行動の対で知識が表現される。次の例は、産研豊田研で行われている辞書の構造化に際して用いる知識をプロダクションルールで表現したものである。

RULE 3790：（入力データの先頭文字列がボールド体）
→（その先頭文字列を見出し語として登録）

RULE 3791：（入力データが（＜A）で始まる）＆（Aは言語名）
→（Aを見出し語の語源として登録）

プロダクションルールによる知識の追加、変更、及び削除は、その知識がモジュラーであれば簡単である。また、個々のルールが読解性に富んでいるので、システムが持っている知識を容易に知ることができる。

手続き的・宣言的という分類ではなく、以下のような知識表現分類がされることもある。

- 意味ネット
- 概念依存関係
- フレーム
- スクリプト
- 手続き

知識表現の問題は、いずれの方法も形式的かつ論理的評価に耐えるものではないという意味においてアドホックである。要は、①必要な知識を表現でき、②何らかの推論機構に載せるこ

とができ、そして③問題を解く時に使えるということである。

3.2 推論

ある一つの知識表現系で表現された知識から新しい知識を導き出すことを推論という。推論を眺めるのには、推論とその実行を分離して考えるか、一体のものと考えるかという二つの立場がある。前者は記号論理学に基づく明確な定義を持つが、後者は人工知能的システムでよく用いられ、必ずしも定式化されている訳ではない。まず前者について簡単に述べておく。

演繹的推論：問題とその問題の生じた世界などを論理学のような明確な体系、例えば一階述語論理で表現し、与えられた公理系から結論を導出したり、あるいは与えられた結論が公理系から導出できるか否かを決定することなどを主とする推論である。このためには三段論法とその拡張である導出原理が用いられる。しかし、これらの方法は推論規則が単純で、その操作が個々の問題に依存しないため計算機向きであるという反面、対象世界の知識を推論システムに持ち込めない。したがって、推論に要する探索の手数が爆発的に増加する危険性がある。プロダクションシステムによる演繹推論システムは、ある程度対象世界をそのルールに書き込めるため、多少はその欠点をのがれることが可能である。

帰納的推論：既知の知識からは直接には導出されないが、既知の知識とは矛盾しない知識を導出しようとする推論である。帰納的推論では前提は個別的であり、結論は前提を支配する一般則である。したがって、この推論は、一般化、抽象化の過程として重要である。

推定的推論：類推的推論とも呼ばれる。知識が充分構造化されている時、二つの世界の類似性を用いて、一方の世界で成立したことを他方の世界でも導出しようとするものである。

論理学に基づく推論は他にも考え得るがここでは省略する。特に指摘しておきたいのは、帰納及び推定は、不完全な知識から新しい知識を得ようとするものであり、今後より一層の研究が待たれている分野であるということである。

人工知能的システムで用いられる推論には、目標指向型推論などの語で述べられるものがあるが、一般的に定式化されるようなものではない。研究の対象に依存するものであり、研究対象そのものであると言えよう。

推論に関連して、推論の各時点における選択の非決定性の処理が問題となる。Prologプログラムでは、ユーザがステートメントをどのように並べるかによって一義的に決定されてしまうし、LISPプログラムではユーザが独自に用意しなければならない。一般的には、推論システムが対象世界と問題に対して持っているヒューリスティックスを生かした深さ制限付き縦型探索と横型探索が有効であろうと考えられる。

3.3　知識の獲得、理解、及び学習

知識表現は、知識の利用とその獲得の二つの側面から検討されるべきである。後者の知識の獲得問題は、エキスパートシステムの持つ知識ベースにどのようにして知識を蓄えて行くかという問題と直接結び付いているが、その体系化などの手法には確立したものはない。知識獲得を表現する用語として、「学習」、「帰納」、「抽象」、「一般化」、「構造化」、「理解」などが用いられるが、これらの用語は理論的根拠を持たず、経験的あるいは例証的に述べられているに過ぎない。ここでは、知識獲得のいくつかの段階について簡単に述べておく。

- 第一段階：体系化された知識をシステム内に知識ベースに単に加える。
- 第二段階：具体例によるシステム内知識の修正。
- 第三段階：入力された知識の一般化または法則化。

このような分類の他にもいろいろあると考えられる。特に第三段階では、高度な帰納的推論が必要になると考えられる。いくつかのシステム、特に大阪大学産研で行われているICAIに関する研究はこの点をはっきり認識したものである。知識の獲得、理解、及び学習の問題は、知識を獲得するためには、その理解を伴わなければならないし、またその知識の一般化が必要であるという意味において非常に困難な問題を含んでいる。

3.4　対話インタフェイス

人工知能的システムに限らず、人間がシステムと直接会話を交わす上でその仲介役を果す対話インタフェイスは、システム全体の使い易さを決めるという点において極めて重要である。対話インタフェイスに配慮を欠いたシステムは、システムの内部がいくら革新的であっても、使用者はいつかは離れて行くであろう。例えば、 Schlumberger 社の CRYSTAL システムと呼ばれるエキスパートシステムではプログラム全体の約40%近くがユーザインタフェイスに充てられている。

対話インタフェイスの理想的姿を Reddy, R は “gracefully interacting system” として描いた。そこでは、システムがあたかも人間と同じように受け答えしてくれる。そのような機能はなかなか実現できるものではないだろうが、その一部でも実現していきたいものである。

4.　エキスパートシステム

4.1　エキスパートシステムとは

1960 年代の中頃より、応用人工知能の分野に属する知識工学がかなりの成功を収め始めた。エキスパートシステムの研究がそれである。エキスパートシステムとは、熟練した専門家と同等の知識を予めシステムに与えておき、それに基づいて問題を解決するシステムのことである。専門家の持つ専門知識は、問題の周辺にある種々の知識や問題を解くための技巧などから成っ

ている。特に、人間の専門家は公刊された文献には出ていないような知識を用いて、問題解決への最も有望な道筋を見出し必ずしも完全でないデータを効果的に取扱うことのできる能力を持っている。したがって、エキスパートシステムを作り上げるためには、専門家の持つ知識・能力を記述するだけでなく、専門家の置かれた環境（世界）をも記述しておく必要がある。しかし、複雑な社会的あるいは物理的過程を正確に解釈し記述することは極めて困難であり、近似的なモデルしか存在しないことが多い。この意味で、完全に人間の専門家と同一の能力を持つエキスパートシステムを作り上げることは困難である。しかし、目標をある限られた一つの領域のしかも限られた作業を模擬するに留めると、それはそれで役に立つものを作り上げることが可能であり、現存のエキスパートシステムはこのレベルにあるものがほとんどである。

問題を解くアルゴリズムが一見ないような問題でも、知識の表現、抽象レベルにおける問題の記述、問題解決用資源の分配、協調するプロセスの制御、及び推論のおける種々の知識源の統合を考慮することによって問題解決アルゴリズムを与え得ることがある。しかし、エキスパートシステムとして実現するにせよその統一的な手法は確立しておらず、現在のところ case-by-case で処理する以外に方法がない。

一方エキスパートシステムが出来上ったとして、人間でもそうであったように、システムが知識を持てばシステムが自律的に成長して行くのではないかとの期待があり、過去に実現されたエキスパートシステムでもある程度実証されている。

エキスパートシステムの研究が進む過程で知識が実に貴重な資源であり、この利用が大きな富を生み出す可能性がある事が認識されて来たのも興味ある事実である。

4.2　エキスパートシステムにおける問題解決の特色

人間の専門家が持つ知識とは、一般にある領域における記述、関係、あるいは手順のことをいい、その表現形式は、概念、ルール、制約、経験則、発見的方法あるいは因果関係をとることが多い事は前に述べた。しかも、現時点では、ある領域における理論的及び経験的関係を規定する記号表現とこれらの記号表現を操作する手続きとして定義するより他に方法はないと考えられる。また知識と関連してしばしば議論される技能とは、正しい知識とその効果的利用法と考えておく。

エキスパートシステム作成に際して専門家から知識・技能を抽出するといっても、問題領域によっては、必ずしも厳密なアルゴリズムの形式でそれらを抽出できるとは限らない。したがって、必ず経験に基づいた判断が必要である。すなわち、実際の問題では知識が不正確であったり、データが誤っているために、評価すべき選択が二つ以上あることもあり、一つの問題に対する解が得られた場合、その解の得られた過程や根拠を説明する必要がある。

このような説明機能は通常の問題解決には必要ないものであり、その説明により納得して解を採用するなり、あるいはその結論が不適当な場合には既に蓄えられている知識を誤っているものとして改変することが行われる。

このように、システム自らが推論を行う過程に関する知識を持ち、自らが出した結論を説明したり、その正当性を証明できるというのは従来のＡＩシステムと大きく異なる点である。

4.3　エキスパートシステムの分類

本節では実際にエキスパートシステムが作成された分野とそれらの分野において問題となる事項を書き出しておく。

データの解釈：測定データは一般に雑音を含み、ある場合には測定誤りさえ含む。それらのデータより論理的に一貫した矛盾のない結論を抽出する必要がある。

診　　断：システムあるいは生体の欠陥は、同時に複数個の箇所に生じていることが多く、欠陥を見出すために積極的に働き掛ける必要がある。特に、生体については未だ解明されていない点が多く、その結果、解釈のために必要なデータが得にくいことがある。

モニタリング：各種の信号を常に監視して、問題のある場合に警報を出せばよいのであるが、相互に関連を持つシステムや時定数の長いシステムなどではそれなりの工夫が必要である。

予　　測：予測を正しく行うためには、正しい情報を元に推論を行う必要があるが、情報が不完全な場合にはどのように対処すべきかが問題となる。

計 画 立 案：効率的に目標を達成することが肝心であるが、全ての行動に対する因果関係をあらかじめ把握することは困難なため、ある程度試行錯誤的に行うより他に方法がない。また、因果関係を必ずしもトリー状に管理できるとは限らず、抽象化の原則をあらかじめ設定しておく必要がある。

設　　計：CAD／CAMで一般に遭遇する問題が全て当てはまる。大規模問題の目標の設定、分割、相互関連、及び設計結果の維持管理（データベース）などについては未だ定量的基準がない。

この他にデバッギング、修理、制御など分野があるが、これらの問題点は上述中に何らかの形で含まれると考えられるので省略する。

各分野に共通することは、巾広い解の選択可能性と知識あるいはデータの信頼性が低い場合における問題解決が重要だということであろう。

4.4　エキスパートシステムの典型的構成

エキスパートシステムの典型的な構成例を図1に示す。

対話インタフェイス部は、ユーザがシステムに知識を入れるなどの何らかの作業を行う場合にシステムとの対話を仲介するものである。単なるCRT画面の設計などではなくもっと広範なものを含んでおり、システムの使い易さを決定する大きな要素である。知識ベースは専門知識を管理し、知識間の推論を行う部分である。黒板は、解法の計画、仮説、中間結果、あるいは最終結果などを管理し、各コンポーネントはこれらを見ることができる。

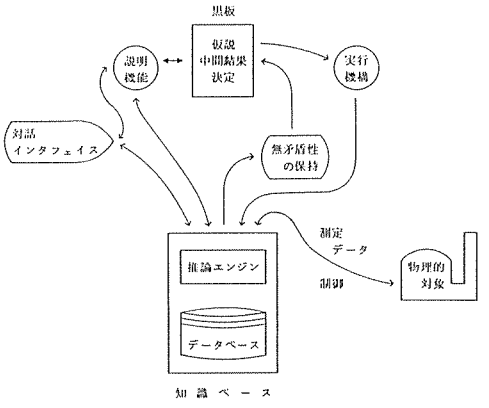

図1　エキスパートシステムの構成

無矛盾性の保持機能部は、システム動作中に設定される種々の仮説や暫定的な結論などの無矛盾性を調べる部分である。また、説明機能部は、システムのとった行動の正当性などをユーザに説明する。

どのようなエキスパートシステムにもこれらの基本的な構成要素が存在する。

4.5　エキスパートシステム作成上の問題点

エキスパートシステムを作り上げる上で問題となるのは、まず第一に「専門家からのエキスパートシステムへの専門知識の移入」である。この知識獲得に際して従来より行われている方法は、計算機の専門家といわゆる専門家が協力して、専門家の知識をシステムに入力するものである。この他、知識の表現法や推論のメカニズムがあらかじめ用意されており、専門家が自然言語による対話インタフェイスを通して計算機屋さんの助けを借りることなく自らの知識をシステムに入力する方法もある。

いずれの方法にしても、人間の専門家が行っている方法でシステムが知識を管理しているとは限らず、場合によっては専門家がシステムの知識管理に対して違和感を持つことがある。

第二の問題点は、システムを組み上げ実際に現場で運用した場合に散見されるであろう不都合な箇所の同定である。不都合な箇所を同定し、知識を更新してシステムを修正していくには、単なる説明機能だけではなく知識更新に対する積極的サポートを必要とする。すなわち、知識更新時の無矛盾性の保持、単調性の保持、及び真実の維持などの機能である。

この他、一般的問題として、(1)知識表現、(2)推論エンジン、(3)開発用ツールなどがあるが、

(1)については既に述べた。(2)については少々問題が大きすぎるので省略する。

4.6　エキスパートシステムの例

いくつかのエキスパートシステムの開発例を表1にまとめておく。

表1　エキスパートシステムの開発例

名　　称	機　　能	言　　語	開 発 機 関
DENDRAL	化学構造式の決定	INTERLISP	Stanford 大
MYCIN	感染症に抗生物質投与決定	INTERLISP	Stanford 大
PUFF	肺機能検査	EMYCIN	Stanford 大
CASNET	緑内障診断	FORTRAN-IV	Rutgers 大
DIP METER ADVISOR	油田探索分析	LISP	Schumberger Ltd.
CATS-1	機関車の故障診断	LISP	GE
MACSYMA	数式処理	MACLISP	MIT
YES／MVS	計算機のモニター	OPS5	IBM

5.　知識工学サポートツール

知識工学的システムの作成において、出発点である知識ベース作成を積極的に支援する枠組をどのように用意するかは重要な問題である。表2にサポートツールのいくつかを示しておく。

表2　サポートツールの例

ツール名	開　　発　　機　　関	サ　ポ　ー　ト　機　能
LISP	————	LISP プログラミング環境
PROLOG	————	PROLOG プログラミング環境
EMYCIN	Stanford 大	ルール主導型診断と説明機能
ROSIE	Rand Corp.	ルール主導型、手続き指向型汎用
KEE	Intelligenetics Ltd.	知識工学開発環境
KAS	SRI	ルール主導型診断と説明機能
OPS5	CMU	汎用、ルール主導型

6.　おわりに

知識工学のすべての話題についてではないが、いくつかのキーポイントについては述べたつもりである。エキスパートシステムは知識工学における数少ない（多分唯一）の成功例であるので、ＡＩビジネスとしてツマミ食いされているのではないかと思われる。個々の話題については諸先生方の項を参考にされたい。

参考文献

1）白井，辻井; 人工知能，岩波講座情報科学22，1982.

2）E.リッチ; 人工知能Ⅰ，Ⅱ，マグロウヒルブック社，1984.

3）Hayes-Roth, F., 他編 ; Building Expert Systems, ADDISON-WESLEY CO. INC., 1983.