

## 知識表現の枠組の統合に関する考察

松原 仁 , 井上 博允 (東京大学工学部)

### 1. はじめに

人間の持っている複雑な知識をどのようにコンピュータ上に表現してそれを制御(利用)するかという問題を扱うのが知識表現であるが、この分野は現在の人工知能における中心的な研究課題である。一般に人工知能のプログラムにおいては、データ構造が知識表現に、アルゴリズムが推論に相当するが、人工知能の歴史の初期の頃は研究者の興味はもっぱらアルゴリズムに集中しており、データ構造にはほとんど関心は払われなかった。当時は人工知能のプログラム、すなわち人間のふるまいをまねるプログラムを作成することは容易であると考えられていたのである。その後人工知能のプログラムを作成することの難しさが正しく認識されるようになり、アルゴリズムのみに注目しては方法論として限界があるということで、データ構造に重点を移したプログラムを作成する傾向に変わってきた。知識表現の枠組についても、(形式)論理、セマンティック・ネットワーク(以下SNと略す)[1]、プロダクション・システム(以下PSと略す)[2, 3]、フレーム[4]などが提案されてきている。人工知能のいろいろな分野のプログラムがこれらの枠組を用いて作成され、その経験によってこれらの枠組にはそれぞれ特徴となる長所と共に短所も存在することが判明した。最近はこちらに代わる新しい枠組を設計する試みが盛んである。その際に既存の枠組とまったく異なるものを設計するのは、当然のことながら難しく、また既存の枠組の長所を直接取り入れることができないので、既存の枠組のいくつかを統合して新しい枠組にするという方針のものが多い。

本稿では、まず代表的な既存の枠組に対して評価を行なう。次に枠組の統合の方式として現在最も一般的な「枠組混在方式」について考察を行ない、さらにその考察に基づいて「枠組融合方式」を提案する。

### 2. 既存の知識表現の枠組

我々の立場を明確にするために、既存の枠組に対して主観的な評価を行なう。既存の枠組の詳しい説明については省略する[5, 6]。枠組の特徴には、インプリメンテーションの仕方によって変化するものがある。枠組が動的であるか静的であるかという特徴はその例である。ここでは比較的インプリメンテーションに不変と思われる特徴を選んで取り上げた。(形式)論理、SN、PS、フレーム(Schank, Abelsonのスクリプト[7]はフレームに含める)を考慮の対象とした。

#### (1) (形式)論理

人間の(意味のある)知的活動の一部は論理的な推論に基づいているという説がある。また、結論自体は論理的な推論によって得られているのではなく、その結論の正

当性を他人に納得させる手段として論理的な推論に基づいてその結論が得られたふりを（多くの場合は無意識に）しているだけであるという説もある。このどちらの説が正しいのかという問題は、人間の知識表現を考える上で重要な点となるだろうが、どちらにせよ、論理によって結論の正当性を他人に納得させることができるのは事実のようである。人間は論理の中でも特に三段論法の正当性を確信しているように思われる。このことは人間の知的活動（の一部）と三段論法との間に何らかの関係があることを示唆している（三段論法についてはたとえばJohnson-Lairdの議論[8]を参照されたい）。この関係の存在は、枠組としての論理の長所と見なすことができる。また、論理は数学的に厳密に定義されているので、確固とした基盤の上で知識を扱うことができるということも長所である。

論理にはこのような長所が存在するが、論理そのものを知識表現の枠組とすることは不適當であると考えられる。論理ではいわゆる例外というものを扱うことができない。論理を拡張すればそれまで扱えなかった例外を扱えるようにできるかもしれないが、また新たにその論理では扱えないような他の例外が出てくる。人間の知的活動の大部分は、論理的な推論と直接には関係がなく、しかもその部分に知能の本質があると考えている。論理は知識をその形式にのみ着目して扱っている。人間は知識を形式のみに着目して扱っているのか、それとも形式と共に意味にも着目して扱っているのかという議論がある[9]が、我々は意味にも着目して扱っていると考えている。

もっとも論理はアルゴリズムを記述するのに有用な道具立てであることにはかわりはない。論理以外の枠組でも、内部に埋め込まれた形で論理が使われているのである。論理はこのように表面には直接出てこない形で使うことが適當と思われる。

## (2) セマンティック・ネットワーク (SN)

知識表現の枠組は一般にデータ構造本体とそのデータ構造へのアクセスの手段とから成る。他の枠組には両者が揃っているのに対して、SNにおいてはアクセスの手段が決まっていない。SNを枠組にする場合は、データ構造を記述しているプログラム言語を用いて、アクセスの手段を直接プログラムとして書くことになる。データ構造をどのように解釈するのかが前もって決まっていないということは枠組としての重大な短所であると思われる。知識をどのように表現するかという問題と、その知識をどのように制御するかという問題は、切り離して考えることはできない[10]。知識は制御の方法が決まって初めてその意味が与えられるのであるから、制御の方法を欠いたデータ構造は知識表現とはいえない。

SNのデータ構造が持つ連想記憶との類似性は長所と見なすことができるが、この長所はフレームに発展的に取り入れられている。

## (3) プロダクション・システム (PS)

PSの長所は、知識の局所性や、空間的な効率のよさや、人間のモデルとしての心理的実在性 (psychological reality) などが挙げられる。PSにおいて知識はルールとして表現されるが、一つ一つのルールは独立しているので、ユーザーが知識を管理するのが容易である。また、ルールのインタープリターとして前向きと後ろ向きの両方を用意しておけば、一つのルールを両方向に制御できる。このことは、たとえばフレームと比べた場合の長所となる。PSの動作は短期記憶、長期記憶に対する刺激

反応系と見なすことができ、さらにはルールは心理学のデモンのモデルにもなっている。ユーザー・インターフェースの面から見ると、PSには知識の書き易さ、読み易さ、理解し易さという長所が存在する。

PSの最大の短所は時間的な効率の悪さである。現在の逐次型コンピュータではルールの条件部を一つずつ調べていく必要があるので、実行に非常に時間がかかってしまう。また、PSでは基本的に階層的な知識を表現できないということも短所である。メタ・レベルのPSにすれば階層的な知識を扱うことは可能であるが[11]、メタ・レベルにすると厳密な意味でのPSとはいえない。知識の多くは階層構造を成しているから見なすのが自然なので、PSを枠組とする場合には問題となる。

なお、PSと論理は形がよく似ているので同一視されることもあるが、我々は明確に別個のものとして取り扱う。PSでは知識を形式と共に意味にも着目して扱うという点で論理とは異なる。さらには、人工知能では知識を手続き型と宣言型とに分類するが[12]、論理では宣言型のみを扱うのに対してPSでは両方とも扱うという点でも異なっている。

#### (4) フレーム

フレームの長所は、人間のモデルとしての心理的実在性や時間的な効率のよさなどが挙げられる。デフォルトや性質の継承のメカニズムはプログラムを記述する上で有用な特徴であるだけでなく、人間のモデルとしても非常に興味深い。デフォルトのメカニズムによって、人間の常識、連想、予想、失望、驚き、誤解などの仕組みを説明することができる。性質の継承のメカニズムは人間の知識の階層性を反映している。さらに付加手続きのメカニズムは、データ駆動型の記述の手段としてだけでなく、心理学のデモンのモデルと見なすこともできる。人間の知的活動の中でコンピュータによる実現が最も困難と考えられている芸術に関しても、フレームがよいモデルとなるという研究もある[13]。また、フレームを枠組とする場合には、フレーム一つ一つに対して中にそのフレームと関係のあるフレームの情報を記述しておくので、そのフレームからただちに関係のあるフレームに制御を移すことができる。

フレームの短所の一つは空間的な効率の悪さである。たとえばフレームAとフレームBの間に何らかの関係がある場合を考える。この場合フレームAの中にフレームBとの関係を記述しておくと同時に、フレームBの中にもフレームAとの関係を記述しておかなければならない。実質的に同じ知識を両方に記述しておかなければならないのである。このことは、時間的な効率のよさの裏返しといえる。また、フレームにおける知識は非局所的なので全体の構造がわからないと知識が記述できないという点も短所である。

なお、最近注目を集めているobject-orientedの概念に基づく枠組は、フレームの別名ないしはフレームのインプリメンテーションの一種と見なしているのも、あえてフレームと区別することはしない。

PrologやSmalltalkなどの汎用プログラミング言語を直接知識表現の枠組と見なす研究者もいる。確かにPrologは論理やPSに、Smalltalkはフレームに形が非常によく似ている。しかし汎用の言語では自由にプログラムが書けるので枠組としての統制

が取れない。知識表現の枠組は枠組として必要かつ十分な記述能力を持っているべきである。汎用の言語では記述能力の範囲が一般に広過ぎる。PrologやSmalltalkは枠組をインプリメントするのに適した言語と考えるのが妥当であろう。

### 3. 枠組混在方式

最近では既存の枠組をいくつか統合して新しい枠組を設計する試みが盛んである。既存の枠組には第2章で指摘したようにそれぞれ一長一短があるので、統合することによって長所を取りまとめようという意図によると思われる。これまでに既存の枠組の統合という形で設計された枠組の多くは、知識の実体を表現する方法として既存の枠組を複数用意しておいて、場合によってはフレームで場合によってはPSでというように、対象となる知識やユーザーの好みに応じて複数の既存の枠組を使い分けるといった方式のものである。

この方式で設計された枠組としてはLOOPS [14, 15]が代表的である。この枠組はProcedure、Object、Data、Ruleと呼ばれる4個のparadigm（枠組とほぼ同じ意味で使われている）を、Objectのparadigmを中心にしてInterlispというProcedureのparadigmの上に統合したものである。LOOPSのユーザーはこの4個の枠組を自由に使い分けすることができる。さらにある枠組の知識から他の枠組の知識を参照することもできる。知識の実体が4通りの方法で表現されているのである。たとえばMANDALA [16]やLOOKS [17]なども、設計者の好みによってどの枠組を中心に置くか（それに応じて枠組にobject-orientedとかlogic-basedとかの名前が付けられている）や既存の枠組の中からどれを選ぶかなどが異なっているけれども、その設計方針には大差なく、基本的にはLOOPSと同じ方式と見なすことができる。現在までに既存の枠組の統合（もしくは結合・融合・組み合わせなど）という形で設計された枠組のほとんどがこの方式である。

知識の実体が複数の方法で表現されているということは、実は複数の枠組が単に混在しているに過ぎない。我々は知識表現の枠組として最も重要なのは知識が統一的に表現されていること、すなわち知識の実体は何らかの一つの統一されたデータ構造として表現されていることであると考えている。この方式ではその原則が損なわれてしまっている。この方式は設計者が主張している枠組の統合（integration）ではな

く、単なる枠組の混合（mixture）であると思われる。そのような意味を込めて、ここでは「枠組混在方式」と呼ぶことにする。枠組混在方式の概念図を図1に示す。ユーザーからは混在している複数の既存の枠組が見えることになる。

ここで混在といっているのは、知識表現の枠組のレベルにおける混在という意味である。LOOPSにおいても最終的にはProcedureのparadigmによって4個のparadigmを実現しているので、そのレベルにおいては実体の表現方法は一つということ

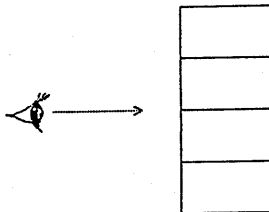


図1 枠組混在方式

になる。さらに機械語のレベルを考えればコンピューター上のプログラムは全て統一された方法で表現されていることになる。知識表現の枠組は機械語や汎用のプログラミング言語 (ex. Interlisp) よりも高いレベルにある。知識表現の枠組のレベルで知識がどのように表現されているかを問題にすべきである。

#### 4. 枠組混在方式の問題点

「枠組混在方式」は、枠組の統合という問題に対して初めて何らかの解答を与えたという点では評価できるが、枠組の設計方針としては必ずしも適当ではないと考えられる。ここではその理由として枠組混在方式の三つの問題点を取り上げる。

##### (1) 統合の方式として

統合とは二つ以上のものを一つにまとめることである。第3章でも述べたように、枠組混在方式では枠組が一つにまとまってはいないと思われる。混在とはいえ個々の枠組は完全に分離しているのではなく、相互に参照できるようにはなっているが、有機的にまとまっている状態と見なすことはできない。枠組混在方式で設計された枠組に統一的な規範が存在するとは思えない。

枠組混合方式の発想は、「Fortran、Pascal、Lisp、Prologなどのプログラミング言語にはそれぞれ一長一短があるので、プログラミング環境として複数の言語を混在させて、ある言語から他の言語を呼び出せるようにしておく。」という発想によく似ている。このプログラミング環境が決してプログラミング言語の統合ではないのと同じ意味で、枠組混在方式も枠組の統合にはなっていない。枠組混在方式は知識表現のレベルでの統合ではなく、その下のプログラミング言語のレベルでの統合に過ぎない。枠組の統合とは複数の枠組を混合することではなく、まして一つのプログラミング言語で複数の枠組をインプリメントすることでもないのである。

##### (2) 人間の知識表現のモデルとして

知識表現の枠組を設計する際は、コンピューター上のデータ構造の一種なので、まず当然のこととして空間的・時間的な効率を考慮する必要がある。さらに、それと同様にもしくはそれ以上に枠組に人間のモデルとして意味があること、別の言い方をすれば心理的実在性があることが重要である。

人間の知識の実体の表現方式がどうなっているかという問題は心理学や大脳生理学そして人工知能の興味深い研究対象である。しかし残念ながらその詳細についてはよくわかっていない。第2章で我々が人間のモデルとしてPSとフレームを高く評価していると述べた。表面上の仕組が相当に異なっているフレームとPSが共に人間のよいモデルであるとするならば、人間の認知や思考のプロセスにはいろいろな側面が存在して、それぞれが異なる側面のモデルになっているということになる。(形式)論理やSNも人間のモデルとしての特徴を持っている。コンピューター上の単なるデータを特に知識と名付けて知識表現の枠組というレベルを設定する以上は、人間の知識表現との関連を問題にしなければそのレベルの存在の意味がないと思われる。

枠組混在方式で統合された枠組は、人間のモデルとしては不自然で心理的実在性に欠けると考えられる。混在している個々の枠組にはそれぞれ人間のモデルとしての特

徴が存在するにもかかわらず、それらを混合したことによって、個々の枠組の特徴が失われてしまっている。混在している枠組間の相互の關係に、人間のモデルとしての意味を見い出すことができないのである。枠組混在方式ではある知識を表現する際に混在しているいくつかの枠組の中からどの枠組を選ぶかという問題が生じる。この問題と人間の知識表現との關係が不明である。また、ある枠組の知識から他の枠組の知識を参照することが人間のモデルとしてどのような意味を持つかがわからない。

### (3) ユーザー・インターフェースとして

知識工学の道具すなわちエキスパート・システム作成ツールとして枠組混在方式を支持する研究者は、複数の枠組を混在させることによってユーザーに対する柔軟なインターフェースを提供すると主張している。しかし知識をどのように表現して制御するかという問題と、ユーザーに対してどのようなインターフェースを提供するかという問題は、基本的には異なるものである。たとえば我々日本人はインターフェースとしては日本語を用いているが、日本人の脳の中の知識表現が直接日本語によって書かれているとは考えられない。インターフェースと内部の知識表現はまったく独立のものではないが、同一のものではない。枠組混在方式によって統合された枠組においては、両者の扱いが混同されていることが多いように見受けられる。

複数の枠組が混在している状態は、ユーザーに対して無用な混乱を引き起こす原因となるので、インターフェースとしても必ずしも適当ではないと思われる。知識表現の枠組と同様に、インターフェースも一つに統一されていることが望ましい。その際にインターフェースの枠組は知識表現の枠組と同一である必要はない。(形式)論理はその有力な候補である。

## 5. 枠組融合方式

我々は、枠組混在方式とは異なる方式によってPSとフレームを統合した枠組UNITYを設計・試作している。PSとフレームを選択した理由は第2章で述べた通りである。これら二つの枠組は相補的で、一方の短所がもう一方の長所になっているので、うまく統合すれば優れた枠組が設計できると考えている。UNITYにおいて知識は一つの統一された実体(データ構造)として表現されている。この実体はEntityと呼ばれており、EntityはPSと見なして解釈することもフレームと見なして解釈することもできる。一つのソースコードに対して二つのインタープリターが存在すると見なすことも可能である。UNITYの概念図を図2に示す。基本的には、ユーザーからEntityの実際の姿は見えなくなっている。ユーザーからはPSまたはフレームのどちらかに見える。同時に両方の枠組が見えること

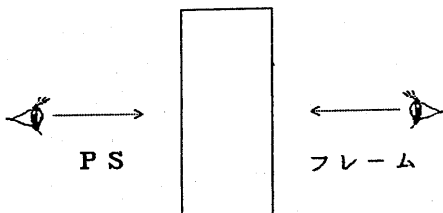


図2 UNITY

は決してない。一つの知識の実体に対して複数の視点が存在するUNITYの方式を「枠組融合方式」と名付けた。知識の実体の表現方法が知識表現の枠組のレベルにおいて一つか複数かという点で、枠組融合方式と枠組混在方式は大きく異なっている。

UNITYは第4章で示した枠組混在方式の問題点をふまえて設計してあるので、当然のことながらこれらの問題点は解決済みである。枠組融合方式は、二つ（以上）の枠組を知識表現の枠組のレベルにおいて一つにまとめているという意味で、枠組の統合といえると考えている。また、ある視点からはPSのように見えてある視点からはフレームのように見えるというUNITYの特徴は、同じような特徴を持つ人間の知識表現のよいモデルになっている。UNITYにおける知識の実体の表現方法は、元のPSやフレームそして枠組混在方式で統合された枠組に比べて複雑ではあるが、本来複雑なはずの人間の知識表現のモデルとしては妥当である。UNITYは知識工学の道具として設計したものではないが、インターフェースも一つに統一されている。

## 6. おわりに

既存の知識表現の枠組の統合の方式として「枠組混在方式」と「枠組融合方式」の考察を行なった。我々の提案する枠組融合方式の方が枠組混在方式よりも統合の方式として適当と考えている。UNITY及び枠組融合方式の詳細についてはまた別の機会に譲りたい。

## 参考文献

- [1] Quillian, M.R.:  
"Semantic Memory", in Minsky, M. (ed.), "Semantic information processing", MIT Press, pp.227-270(1968)
- [2] Newell, A. and Simon, H.A.:  
"Human problem solving", Prentice-Hall(1972)
- [3] Post, E.:  
"Formal reductions of the general combinatorial problem",  
American Journal of Mathematics 65, pp.197-268(1943)
- [4] Minsky, M.:  
"A framework for representing knowledge", in Winston, P. (ed.),  
"The psychology of computer vision", McGraw-Hill, pp.211-277(1975)
- [5] Barr, A. and Feigenbaum, E.A. (eds.):  
"The handbook of artificial intelligence vol.1", William Kaufmann(1981)
- [6] 中島 秀之:  
"知識表現とProlog/KR", 産業図書 (1985)
- [7] Schank, R.C. and Abelson, R.P.:  
"Scripts, plans, goals, and understanding", Lawrence Erlbaum(1977)

- [ 8 ] Johnson-Laird,P.N.:  
 "Mental models in cognitive science",Cognitive Science 4,pp.71-115(1980)
- [ 9 ] Dennett,D.:  
 "The logical geography of computational approaches: a view from the East Pole",Conf. of Philosophy and Cognitive Science,MIT(1984)
- [ 1 0 ] Newell,A.:  
 "The knowledge level", Artificial Intelligence 18,pp.87-127(1982)
- [ 1 1 ] 松原 仁、井上 博允:  
 "プロダクション・システムによる線画の解釈",  
 情報処理学会, コンピュータービジョン研究会26-1(1983)
- [ 1 2 ] Winograd,T.:  
 "Frame representations and the declarative/procedural controversy",  
 in Bobrow,D. and Collins,A.(eds.),"Representation and understanding",  
 Academic Press(1975)
- [ 1 3 ] 川野 洋:  
 "コンピュータと美学",東京大学出版会(1984)
- [ 1 4 ] Bobrow,D.G. and Stefik,M.J.:  
 "The LOOPS manual",Intelligent Systems Laboratory, Xerox Corporation(1983)
- [ 1 5 ] 丸一 威雄他:  
 "言語/システムLOOPSのオブジェクト指向からのアプローチ",  
 情報処理学会, ソフトウェア工学研究会35-4(1984)
- [ 1 6 ] Furukawa,K. et al.:  
 "MANDALA:a logic based knowledge programming sytem",Proc. of FGCS'84,ICOT,  
 pp.613-622(1984)
- [ 1 7 ] Mizoguchi,F. et al.:  
 "LOOKS:knowledge representation system for designing expert systems in  
 a logic programming framework",Proc. of FGCS'84,ICOT, pp.606-612(1984)