知識工学と人工知能 44－6
（１９８６．１．２０）

混成型問題解決機ＰＳＡについて

小川　均，北村泰彦，田村進一
（大阪大学基礎工学部）

１．はじめに

問題解決システムの構築に必要なものは問題の表現法，問題の分割，問題解決のためのオペレータ，推論方法の４項目である．したがって，問題解決システムを作成する場合，システム作成者は上記の４項目すべてを熟知している必要がある．一般的に，問題をよく理解しているある分野の専門家は，問題の表現法および推論方法には詳しくなく，人工知能研究者はその逆である．専門分野の問題解決システム（すなわち，エキスパートシステム）の作成の支援のため，多くのエキスパートシステム構築ツールが開発されている．ツールは問題の表現法を制限し，特定の推論方法を提供するので，作成されるシステムはその表現法および推論の能力に強く規定される．たとえば，ＯＰＳ５[19)20)]はルールに基づいた前向き推論を使用している．Ｍ１[6)]やＥＭＹＣＩＮ[17)]はルールに基づき，確信度を扱う後向き推論を使用しており，可換プロダクションシステム[13)]を形成する．可換プロダクションシステムとは，適用可能な複数のルールがその適用により互いに影響しない単調さと，ルールの適用順序を適用可能なようにどのように置換してもよい半可換の特徴を持つプロダクションシステムである．各問題に対して，適切なシステムを柔軟に構成するための混成型（hybrid型）ツール[6)]としてＥＳＨＥＬＬ[15)]，ＫＥＥ[8)]等が提案されている．これらのツールでは，知識表現・推論機構が複数用意されており，使用者による指示，または，ＬＩＳＰにより新たな作成により，エキスパートシステムが構築される．このことは，システム構築の柔軟性が得られる反面，使用者に多くの概念やＬＩＳＰの知識が要求され，負担が大きい．

問題解決システムの構成支援には目的に対応できる柔軟な構造が望まれる．しかし，知識表現や推論方法について詳しく知っていなくても問題解決システムを実現できる環境も必要である．本稿においては，種々の問題解決に対して望まれる知識表現および推論方法を必要に応じて統合して使用できる混成型問題解決機ＰＳＡ（Problem Solving Agent）を提案する．問題解決システムの構成支援を他の観点から考えると，ルール作成・維持のための援助，知識入力時の意味的整合性の確認，マンマシンインタフェースもまた重要であるが本稿では扱わない．

２．ＰＳＡについて

種々の問題解決機の能力を実現を目的としたＰＳＡ（Problem Solving Agent）を提案する．

２．１　ＰＳＡの概略

ＰＳＡの構造は図１に示されるように，推論制御部，データを蓄積するブラックボード，および，複数のルールグループからなる．推論制御部には推論制御，

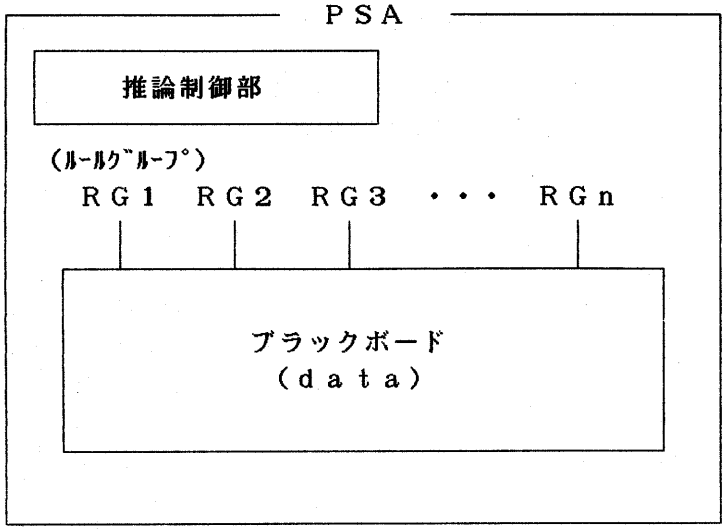

図１．　ＰＳＡの構造

デモン処理，メッセージ処理を行なうルールが記述される．推論制御部中のルールをメタルールと呼ぶ．推論のためのルールは目的や対象毎にグループにまとめられている．ブラックボードには，対象問題に関する情報，および，仮定や目標，推論による途中結果等のデータが蓄積される．

ＰＳＡはメッセージを受理することにより起動される．メッセージに対応する動作は推論制御部で定義される．すなわち，データの登録，推論の実行，推論結果のＣＲＴへの表示，他のＰＳＡへのメッセージ発信等である．推論はルールグループと推論方法を指定することにより実行される．１つのルールグループは複数のＰＳＡによって使用できる．推論用ルールの実行時の副作用（取り消しができない実行）はブラックボードに対してデータの断言と削除のみである．

ＰＳＡは論理型プログラム言語Prolog[1]により実現されているので，変数を英大文字を先頭に持つ語で表わし，その他は英小文字の語で表わす．

２．２　ＰＳＡの表現方法

（１）ＰＳＡの構成

ＰＳＡは推論制御部，データを蓄積するブラックボード，および，複数のルールグループからなる．ＰＳＡの各部分の初期値はそれぞれ別のファイルに蓄積される．ファイル名は以下の形式でなければならない．

ａ）＜ＰＳＡ名＞．ＭＥＴ：　推論制御部のルールを蓄積するファイル名．
ｂ）＜ルールグループ名＞．ＲＵＬ：　推論用ルールを蓄積するファイル名．
ｃ）＜ＰＳＡ名＞．ＤＡＴ：　ＰＳＡのデータを蓄積するファイル名．

（２）データの表現

推論の初期値，途中結果，仮説，目標等すべてのデータはブラックボードに蓄積される．データには以下の表現形式を使用するが，ルール，および，メタルー

ル中では＜述語式＞を記述するだけで参照される．

data(＜ＰＳＡ名＞，＜述語式＞，＜確信度＞)．

確信度の取る値は，一般的に，０から１，または，－１から１であるが，ＰＳＡを記述するプログラミング言語Prolog-KABA[5] では実数が扱えないために確信度の取る値は０から１００，または，－１００から１００である．どちらの値を使用するのかは推論方法に依存する．

［データの例］　ＰＳＡtestfwcaのデータを示す．

```
data(testfwca,fact(has(feather)),90).
                ...  羽毛を持つという事実がある．（確信度０．９）
data(testfwca,fact(has(egg)),95).
                ..... 卵を持つという事実がある．（確信度０．９５）
data(testfwca,fact(can(fly)),70).
                ..... 飛べるという事実がある．（確信度０．７）
data(testfwca,fact(color(black)),80).
                ..... 色が黒いという事実がある．（確信度０．８）
```

（３）ルールの表現

ルールは以下の表現形式により記述される．

rule(＜グループ名＞，＜ルール名＞，＜条件式＞，＜結論式＞，＜確信度＞)．

＜グループ名＞はルールが含まれるルールグループの名前である．＜ルール名＞はルール固有の名前であり，他のルールとの区別やコメントとして用いることができる．＜条件式＞および＜結論式＞は論理式または［と］で囲まれた論理式の集合で表現される．［と］で囲まれた論理式は左から処理される．又，論理式では and，or，not を使用することが出来る．＜確信度＞は０から１００で表わす．ルールにおいてブラックボードに対するデータ処理のためにassertdata，delete data，noexistが使用できる．これらの述語はメタルールで使用できるものの一部である．

［ルールの例］　ルールグループ名testcのルールを示す．

```
rule(testc,rc1,
     [fact(ako(bird)),fact(color(black))],
     assertdata(fact(name(crow))),70).
                ..... 色が黒い鳥はカラスである．（確信度０．７）
rule(testc,rc2,
     [fact(has(egg)),fact(can(fly))],
     assertdata(fact(ako(bird))),90).
                ..... 卵を持ち，飛べれば，鳥である．（確信度０．９）
rule(testc,rc3,
     fact(has(feather)),
     assertdata(fact(ako(bird))),80).
                ..... 羽毛を持っておれば，鳥である．（確信度０．８）
```

（４）メタルールの表現と機能

メタルールの表現形式は，

metarule(＜ＰＳＡ名＞，＜ルール名＞，＜条件式＞，＜結論式＞)．

であり，ルールと比べて＜確信度＞はない．メタルールの＜条件式＞は唯一の述

語式であり，ブラックボード上のデータを参照するのではなく，動作や出来事に関する記述が書かれる．メタルールの機能は他のＰＳＡと交換するメッセージの処理，および，ルールの使用法（推論法）の指示，メタルールの変更・維持，データ処理，デモン処理（ルールの動作に対応した処理）を行なう．これらの実行や推論状況の表現のために以下の述語が使用できる．メタルールは前向き推論で評価される．

[動作や出来事に関する述語]

・deaddata(D) ..... データＤが削除された．

・envelope(M,S) ..... ＳからメッセージＭが送られた．

・exec(P) ..... 述語式Ｐが実行された．

・fail(R,P,I) ..... ルール・グループＲを使用し推論方法Ｉで推論した結果，Ｐが得られなかった．Ｒ，Ｐ，Ｉはそれぞれ変数や変数を含む述語であってもよい．この場合は，Ｒ，Ｉに照合するルール・グループ，推論方法を用いたときＰに照合する結果が得られなかったことを示す．

・newdata(D) ..... 新しいデータＤが断言された．

・success(R,P,I) ..... ルール・グループＲを使用し推論方法Ｉで推論した結果，Ｐを得た．Ｒ，Ｐ，Ｉはそれぞれ変数や変数を含む述語であってもよい．この場合は，Ｒ，Ｉに照合するルール・グループ，推論方法を用いたときＰに照合する結果を得たことを示す．

[メッセージの処理]

・mail(T,M,S) ..... ＳからＴにメッセージＭを送る．

[推論に関する述語]

・inference(R,P,I) .... 問題Ｐをルール・グループＲを使用し推論方法Ｉで解決する．ここでルール・グループ名を省略すると，ＰＳＡ名をルール・グループ名としてルールが使用される．現在，使用できる推論方法は，ｆｗ（前向き推論），ｆｗａ（前向き推論で全ての目標を求める．），ｆｗｃ（前向き推論ＣＦ付き），ｆｗｃａ（ＣＦを用いた前向き推論で全ての目標を求める．），ｆｗｆ（前向き推論ｆｕｚｚｙ[18]付き），ｂｗ（後向き推論）の６つである．

・noretry ..... execで示された述語式を再実行しない．Prologの！（カット）と同じ働きをする．

[データ処理]

・assertdata(D) ..... データＤをブラックボードに書き込む．

・assertdata(D,CF) ..... 確信度CFのデータＤをブラックボードに書き込む．

・assertdata1(D) ..... データＤをブラックボードに書き込む（ただしＤはリスト）．

・asserted(D) ..... データＤがブラックボードに存在する．

・asserted(D,CF) ..... 確信度CFのデータＤがブラックボードに存在する．

・cleardata(D) ..... deletedata(D)と同じであるが，常に成功する

・deletedata(D) ..... データDをブラックボードから削除する．
・deletedata(D,CF) ..... データDの確信度を減少する．
・initdata ..... ＰＳＡ内のデータをクリアーする．
・noexist(D) ..... データDがブラックボードに存在しない．

［メタルールの例］ ＰＳＡ名testfwca中のメタルールを示す．

```
metarule(testfwca,mfwca1,
        envelope(achieve(A),S),inference(testc,A,fwca)).
          ..... メッセージachieve(A)が送られてくれば，
                inference(testc,A,fwca)（ＣＦを用いた前向き推論
                で全ての目標を求める．）を実行する．
metarule(testfwca,mfwca2,
        success(testc,B,fwca),mail(me,solution(B),psa1)).
          ..... 推論が成功すれば，使用者にsolution(B)を送る．た
                だし，Ｂには解が代入されている．
metarule(testfwca,mfwca3,
        fail(testc,B,fwca),mail(me,fail,psa1)).
          ..... 推論が失敗すれば，使用者にfailを送る．
metarule(testfwca,mfwca4,
        newdata(fact(A)),
        [asserted(fact(A),F),write(A),write(' (確信度 '),
         write(F),write(' ) '),write('  が得られた。'),nl]).
          ..... fact(A)に照合する新しいデータがデータ部に断言さ
                れれば，ＣＲＴにメッセージを出力する．
```

２．３ 実行例

前節の例で示したＰＳＡtestfwcaのメタルールとデータ，および，ルールグループtestcを用いた例を示す．testfwcaにメッセージachieve(fact(name(A)))を送ることにより，以下のようにＣＲＴに表示される．

```
ako(bird)（確信度 63 ）  が得られた。
name(crow)（確信度 44 ）  が得られた。
ako(bird)（確信度 72 ）  が得られた。
name(crow)（確信度 50 ）  が得られた。
```

３．考察

ＰＳＡによりどのような能力を実現できるか考察する．

３．１ 問題解決パラダイムとＰＳＡ

（１）プロダクションシステム[7]

ＰＳＡの動作はルールによって定義する．１つのＰＳＡを用い，適当な推論方

法を指定すれば，プロダクションシステムは簡単に構成できる．

（２）ブラックボード・モデル[3]

ＰＳＡでは，ブラックボード上に蓄積しているデータに対して，適用するルールグループと推論方法を指定することができ，柔軟な処理が可能である．

（３）オブジェクト指向

問題解決システムは複数のＰＳＡから構成できる．オブジェクト指向型プログラミングの観点から見ると，ＰＳＡはオブジェクトに対応する．オブジェクト指向型プログラム環境としてsmalltalk[4]が代表的である．ＰＳＡはsmalltalkのような小さなオブジェクトを対象とせず，一つの問題（たとえば，定理の証明やロボット・プランニング）を解くのに十分大きなオブジェクトに対応する．関係した研究にブラックボードシステムを分散システムに適用したものがある．[2)16)]

問題解決のための大まかな方針，または，モデルが与えられている場合，個々の処理に対応するＰＳＡを構築すればよい．このことは，与えられた問題を副問題に分割し，各副問題を専門的に解決することを示している．

（４）データ指向

デモン処理用の述語deaddataとnewdataにより，特定のデータの変化に対する処理が記述できる．これにより，ＬＯＯＰＳ[14]のActive Valueと同等の能力を定義することができる．

３．２　知識表現とＰＳＡ

（１）述語論理式[10]

ＰＳＡの表現は論理式を基本としている．従って，対象としている問題の意味，状態，目標を述語式により表現できる．ただし，表現の簡単化のため限定子を必要としない節形に従っている．また，ＰＳＡは論理型プログラム言語Prologにより実現されているので，変数を英大文字を先頭に持つ語で表わし，その他は英小文字の語で表わす．

（２）ルール表現

ＰＳＡの行動知識はルールによって表現される．ブラックボード上のデータを参照し，動作を決定するプロダクション・システムの機能を持っている．

（３）意味ネットワーク[11]

意味ネットワークは，事象や項目をノードとして表現し，それらの関係をアークで結んでいる．すなわち，意味ネットワークは２つのノードを始点と終点にしたアークの集合である．論理式を用いてこの関係を表わすことができる．アークに対応して述語を使用し，ノードを引数とすればよい．このような論理式の集合は意味ネットワークを表わしている．

（４）フレーム[12]

ある項目のノードに対して模範的なアークを選定し，それらをまとめたものがフレームの基本的考えである．さらに，フレームではデータの獲得やそれに付随した処理を手続きの形で表現できる．現在のＰＳＡでは，フレームと同様な付加手続きの表現はできない．しかしながら，同じ機能を実現することはできる．データ獲得のための手続きはルールとして表現する．データの付加および削除に対するデモン的処理は推論制御部において定義することができる．

４．３　多様な推論方法

推論方法を分類する項目には推論の方向，解の個数，解の探索方法，ルール選択方法，および，あいまい推論がある．これらについて，パラメータ指定による推論方法の決定はプログラムの複雑さ，実行時の効率低下の問題がある．そこで，ＰＳＡではそれぞれ目的に合うコンパクトで効率のよい推論方法をPrologで構築し，名前で指定するようにした．したがって，一般のＰＳＡ使用者はあらかじめ用意された推論方法から選択しなければならない．システム述語inferenceを用いて推論方法が指示できる．また，新たに推論方法をPrologで作成する場合は，簡単な条件（推論名を述語とし，ＰＳＡ名，ルールグループ名，目標を引き数とする．）を守るだけでよい．

４．４　幅広いレベルの利用者

ＰＳＡの利用者は，初心者，一般使用者，研究者の３種類に分けることができる．以下はそれぞれの使用法について述べる．

（１）研究者：知識の表現法や推論方法について研究する者である．４．３で述べた，新たな推論方法を定義し，実験を行なう．このため，推論方法をＰＳＡに追加する際の制限を知っている必要があり，また，Prologについて精通しているのが望ましい．

（２）一般使用者：メタルール，ルール，データを自由に定義することにより，希望する問題解決システムを構成し，利用する．推論方法に関しては，あらかじめ用意されたものから選択する．新らしい推論方法が必要な場合には，研究者にその開発を依頼しなけらばならない．

（３）初心者：問題解決の状態に合わせて適切な推論方法を選択するというよりは，単にルールとデータからどのような結果が得られるかを知りたい者である．メタルールの定義に関しては，研究者または一般使用者の援助を必要とする．

５．むすび

問題解決に対して望まれる知識表現および推論方法を必要に応じて統合して使用できる混成型問題解決機ＰＳＡ（Problem Solving Agent）を提案した．ＰＳＡはNEC PC-9801上のProlog-KABAで実現されている．ＰＳＡを用いて分散型人工知能システムやセメントのキルンの制御エキスパートシステムの構築が始められている．問題解決システムの構築には，ルール作成・維持のための援助，知識入力時の意味的整合性の確認，ヒューマンインタフェースも必要であるが，ＰＳＡ自体はこの能力はない．しかしながら，ＰＳＡを用いてこれらを実現することは可能である．

ＰＳＡ構築に関してシステム綜合開発株式会社の大木宗一氏の協力に感謝する．

参考文献

1)Clocksin W. F. & C. S. Mellish: Programming in Prolog, Springer-Verlag

(1981).
2)Corkill, D. D. & U. R. Lesser: The Use Of Meta-Level Control For Coordination In A Distributed Problem Solving Network, Proc of IJCAI-83, pp.748-756 (1983).
3)Erman, L. D. et al:The Hearsay II, Speech Understanding System: Integrateing Knowledge to Resolve Uncertainty, Computing Survey, Vol. 12, pp.213-253 (1980).
4)Goldberg Adele & David Robson: SMALLTALK-80: the language and its implementation, Addison-Wesley Publishing Company (1983).
5)萩野達也等：Prolog KABA Reference Manual, アステック (1985).
6)Harmon, Paul & David King: Expert Systems, John Wiley & Sons, Inc. (1985).
7)小林重信：プロダクション・システム，情報処理， 26, 12, pp.1487-1496 (1985).
8)Kunz, J. C. etal: Applications Development Using a Hybrid AI Development System, AI Magazine, 5, 3 (1984).
9)Minsky, M.: A Framework for Representing Knowledge, in The Psychology of Computer Vision, edited by P. H. Winston, McGraw-Hill Book Company, New York (1975)（白井，杉原訳：コンピュータービジョンの心理，産業図書 (1979)）.
10)中島秀之：論理に基づく知識の表現，情報処理， 26, 12, pp.1512-1519 (1985).
11)岡本敏雄：セマンティク・ネットワーク・システム， 情報処理， 26, 12, pp.1504-1511 (1985).
12)小川　均：フレーム理論に基づく知識表現言語，情報処理， 26,12, pp.1497-1503 (1985).
13)Rich, Elaine: Artificial Intelligence, McGraw-Hill (1983)（広田，宮村訳：人工知能Ⅰ，Ⅱ，マグロウヒルブック㈱ (1984)）.
14)Stefik, M. etal:Knowledge Programming in LOOPS: Report on an Experimental Course, AI Magazine, 4, 3 (1983).
15)多田，中島：ＥＳＨＥＬＬ，人工知能ウォーズ，日刊工業新聞社, pp.100-101 (1985).
16)Yang J. D., M. N. Huhns and L. M. Stephans: An Architecture for Control and Communication in Distributed Artificial Intelligent System, IEEE, SMC-15, No. 3 (1985).
17)van Melle, W.:A Domain-independent Production Rule System for Consultation Programs, Proc. of IJCAI-79 (1979).
18) Zadeh, L, A.: Probability measures of fuzzy events. Journal of Mathematical Analysis and Applications. 23, pp.421-427 (1968).
19)VAX-11 OPS5 ユーザーズ　マニュアル，日本ディジタルイクイプメント㈱ (1984).
20)VAX-11 OPS5 ユーザーズ　ガイド，日本ディジタルイクイプメント㈱ (1984).