愛知工業大学研究報告
第22号B　昭和62年

Prologによる性格診断

小田哲久・尾関守

Personality Test with Prolog

Tetsuhisa ODA and Mamoru OZEKI

In the present paper, an AI (Artificial Intelligence) approach to knowing one's personality is discussed . PROLOG, which is a typical AI oriented computer language, is used to implement a personality test on a computer as a kind of expert system. Y-G (Yatabe-Guilford) personality test is chosed a subject to be implemented, since it is one of the most popular peronality tests in Japan. Through the implementing jop, following problems were awared. They were about (1) the presentation order of the questions, (2) detecting false responses by a subject, and (3) composing reasonable diagnostic messages. The program list, shown in the appendix, is a part of the program of the core system of the AI computer based personality test system. To complete the total system, heuristic diagnosis module should be additionally developed.

1．はじめに

性格診断に質問紙法形式が採用されるようになって，久しい年月が過ぎ去った。その歴史における発展の契機は，第一次世界大戦参戦時の米国で，欧州戦線へ派兵するに当って，応募兵の中より，兵員としての適性を欠く人物をスクリーニングする為に，性格調査目録（Personality Inventory）を作成して，この用にあてた事にあるといわれる。

日本では，J．P．ギルフォードの性格目録を基礎として，矢田部達郎・辻岡美延らによって，日本人向けに作成されたY-G検査（矢田部ギルフォード性格検査）が，極めて広範囲に使われて来ている。これ以外にも，東大式サーストン気質検査，辰見敏夫氏診断性向検査，改訂バーンロイター人格目録，児玉氏MMPI，東大式MMPI等，各種の性格検査が作成されているが，産業への応用としては，もっぱら，Y-G検査が利用されている。

ところで，我々は，ここ数年来，人工知能用言語Prologの潜在的可能性に着目し，その応用化研究を行って来ているが，本研究報告では，その一環としての，「Prologによる性格診断プログラム」を取り上げる。Prologの応用の対象として，性格診断をとり上げた理由は，以下の諸点による。

①人工知能は，専門家の知識をルール化した知識ベースを備え，現実の症状（状態）を元に，推論機構を働かせて診断（あるいは設計）を行う，いわゆる「エキスパートシステム」の型が当面，有望であると考えられており，一方，性格診断は医師の診断と類似している，あるいはその一部とも考えられ，実現の可能性が高いと思われる。

②性格診断は高度にルール化されており，文献を利用して，一応の水準のエキスパート・システムを構築できそうである。

③人工知能研究という次元から，「性格診断」という課題を考えることによって，性格診断の方法論に新しい知見がもたらされるのではないか。少なくとも，質問紙というメディアに制約されている部分が浮き彫りにされるのではないか。

そして，各種性格検査法のうち，まずはじめに，Y-G検査をとりあげた。この検査は我々経営工学を学ぶ者にとって，最もなじみ深いものであり，又，詳細な診断事例に関する文献が公刊されており，我々のねらいに合致するという点も，その広範・長期にわたる実績と共に，最初の研究対象とした理由である。

2．研究方法

2.1　Prolog処理系と文法

Prologには各種の処理系があり，それに応じて，文法に若干の相違点がある。本研究では，世界的標準とみなされる，エジンバラ大学版のDEC-10 Prologの文法に準拠する事を目標としたが，本学に於て現実に利用可能なハードウェア・ソフトウェアの範囲で，可能な限り目標に近づける事とした。特に，Prologは述語単位での開発が可能である事から，小部分については，CP/M-80版の，Expert Systems社製，Prolog-1をもっぱら利用し

た。しかし，全体はとてもCP/M-80マシンではメモリに格納し切れないため，CP/M-68K上の，ハイパーウェア社製，Hyperware-Prologを用いた。又，改良作業は，TRS-XENIX上にて，シニック社製，K-Prologを用いた。

2.2 使用計算機

上記処理系は，各々，以下の計算機上で稼動した。カッコ内はＯＳ名である。

①沖電気製 if-800/20 （CP/M-80）
②タロス社製 TAROS-68K （CP/M-68K）
③タンディ社製 TRS-16B （TRS-XENIX）

2.3 システム開発の方針

①辻岡美延著，「新性格検査法」[1]をＹ-Ｇ検査の基本とし，江口恒男著，「性格診断マニュアル」[2]を，産業面への応用事例集として活用する。

②可能な限り，質問紙による場合との診断結果の比較がしやすいように工夫する。

③当面，ユーザーインターフェースは，ビデオディスプレイ上への表示ならびにキーボード入力とする。

④質問や診断結果の表示は，当面，ローマ字表記の日本語とし，将来的には漢字表記が可能な環境下へと移す。

3．結 果

3.1 Prologプログラムの冗長性

図１に，プログラムの一部を示す。このプログラムは，まだまだ冗長であり，各型名をtype Ａなどと，それぞれ独立のアトムで表現しているが，この事が災いして，各型ごとに出力の述語を用意する必要が生じるなどしている。その様な訳で，データー構造に関して，大いに改良の余地がある。

3.2 システム開発の方針と，開発の現状

被験者の応答結果を集計して類型分類する事は，極めて機械的な作業であり，コンピュータによる実現は容易で，特別の問題点はない。問題は，そこからさらに一歩ふみこんだ診断を下そうとする所にある。我々のプログラムにも，その種の診断メカニズムがとり入れられるように準備だけは行ってある。ただし，現時点では機能していない。

正規のＹ-Ｇ検査と比較しやすいように，という方針については，充分満足されるにはほど遠い状態である。開発方針④に端的に表されているような開発環境の制約，さらにProlog処理系の入出力の貧弱さが，その理由であり，今後，何らかの便利なユーザーインターフェースを考えてゆく必要があろう。

4．考 察

4.1 エキスパートシステム化のための重要事項

結果の類型分類にとどまらず，システム開発を進めて，さらにエキスパートに近いシステムを完成するには，次のような方針のうち，いくつかを採用する必要がある。

①専門家による「診断」のメカニズムを，診断事例を元にして，ヒューリスティックなルールを抽出し，これをシステムに組み込む。
②適切な診断ルールが抽出できない場合には，数多くの診断事例をそのままケースとして取り入れ，診断すべきデータとの距離の近いケースを選び出すようにする。
③プロフィールを入力して，専門家の診断結果と対比できるように，システムを分割し，プロフィール入力から診断出力までを，１つのモジュールとして考える。
④プロフィール・レベルでの入力と表示は，診断ルール確立のうえでも重要なので，ＶＤＴ上にグラフィック表示できるようにする。

4.2 性格診断のエキスパートシステム化による問題点と新たな可能性

性格診断法の問題点は，様々な視点から論ずる事ができる。（例えば，日本臨床心理学会編，「心理テスト――その虚構と現実」[3]など。）しかし，本研究では，質問紙の「紙」（Ｙ-Ｇ検査の場合は，カーボンペーパー）から，ビデオディスプレイとキーボードを入出力装置とし，推論機構を備えたＡＩコンピュータシステムへとメディアが大きく変化した事に起因する技術的な問題に的をしぼりたい。しかしながら，当初，技術的と思われた問題が，やがて他の分野にも影響を及ぼす事は，ありうる事と考える。

我々のシステムは，まだ充分な出来栄えでなく，とても質問紙法にたち打ちできる水準にないが，これまでの開発過程で，次の様な問題が着想されたので，それらについて考察を加えた。

①質問の順序を固定する必要はないので，被験者ごとに順序を変更する事が出来るようにしたらどうか。
②虚偽の反応に対して，新たな対策は考えられないか。
③機械で診断を行うと，診断の文章が，断片的な文章の形式的な組合せの作文になり，ナンセンスなものにならないだろうか。

これらの問題については，思考実験に過ぎないが，一応，以下の様な予想を立てた。

①単に，各試行ごとに変更する事には意義は認められない。ただし，被験者側の発言（訴え，問いかけ，質問に対する回答，無意識の発声など）に応じて，適切な

質問が次々と出せるならば，それはそれで，一つの理想的な診断システムの入力部となると言えよう。これは，かのELIZAプログラムを強化したていどのものでも，可能かもしれない。

②質問が表示されてから，キーボードを押し下げするまでの反応時間を検出する事によって，あるていど真実性の疑わしい反応が発見できると思われる。又，応答者の確信度については，音声応答を採用すれば，反応時間あるいはピッチ変動の検出によって，キーボードへの反応と同等か，さらに多くの情報を得る事が可能となる。一般的にいって，音声入力は実現が容易ではないが，「はい」，「いいえ」，など，語が決まっている場合には，極端に困難というほどではなく，技術的に実現が可能な範囲と考えられる。

又，別の見方として，特別な対策をとらなくても，検査結果を，他者である専門家ではなく，「機械」が判定し，その結果についてのプライバシーが守られる状況であれば，虚偽の反応は少なくなろう。

③専門家による「診断」と異なり，機械は正に組合せ論的な手順で「診断」を下すと考えられ，従来の例にないパターンについては，項目間の相互作用を無視するような誤りを犯すかもしれない。しかし，そのような例外については，必ず，専門家による診断も行い，両者の診断が合っていなければ，ルールの改訂，あるいは新事例の記録として処理してゆけば，いずれはそのような例外の発生が少なくなってゆくと考えられる。又，余りに例外ばかりが出現するならば，ルール化が不充分であるか，専門家の「診断」が非定形であるかのいずれかの可能性があり，専門家にとっての反省材料となろう。

5．結　び

性格診断をコンピュータ上で行えるようにする，という試みは，古くから行われており，それを新しいコンピュータ言語で行ったとしても，特別な意義はない。本研究の特色は，人工知能システムという次元から，性格検査法の技術的問題点に対して考察を加えようとする点にある。現時点では，単に類型判定のできるプログラムを開発した段階に過ぎず，今後，専門家の診断事例を組み込むなどして，システムを発展させたい。

又，Y-G検査以外の検査についても，Prolog化を試みてみたいが，Y-G用のプログラムが完成すれば，これをツール化して，他の検査に適用する方が，ゼロから始めるよりは容易であろう。

今後，上述のような作業を進める中で，人工知能的視点で，新しい性格検査についてのビジョンを形成してゆく事が，我々にとって，最も重要な課題である。

参考文献

1）辻岡美延：新性格検査法――Y-G性格検査実施・応用・研究手引――，竹井機器工業，1969.
2）江口恒男：性格診断マニュアル，テクノ，1976.
3）日本臨床心理学会：心理テスト・その虚構と現実，1979.
4）井村恒郎：臨床心理検査法（第2版），医学書院，1967.
5）依田　新，岡本栄一，福屋武人：心理学における実験と測定，日本文化科学社，1978.

付録 プログラムリスト（一部）

```
/*      DISPLAY INSTRUCTIONS    */

begin :-
        nl,
        tab(4) ,
        instruction,
        nl,
        tab(4),
        fail.
begin :- start.

instruction :-
        write('kono PROGRAM wa Y-G TEST wo sankounishite tsukurareta').
instruction :-
        write('seikaku shindan TEST desu.').
instruction :-
        write('shitsumon ni subayaku okotae kudasai.') .
instruction :-
        write('kotae ga "hai" naraba "y." to kii wo oshite kudasai ') .
instruction :-
        write('y no atoni "." wo wasure naide kudasai.') .
instruction :-
        write('sorekara RETURN kii wo oshite kudasai .') .
instruction :-
        write('kotaega "iie" naraba "n." to kii wo oshite kudasai.') .
instruction :-
        write('doushitemo kimerarenai toki niwa "e." to oshite kudasai. ') .
instruction :-
        write('n , e no ato ni "." wo wasurenai de kudasai') .
instruction :-
        write('sorekara RETURN kii wo oshite kudasai .') .
instruction :-
        write('soredewa tesuto wo kaishi shimashou.') .


/*      FIRST START PROCESS     */

start :-
        data([dep,cyc,inf,ner,sub,coop,agr,gea,rha,thex,asce,soe],Xy,Xx) ,
        Wx=[dep,cyc,inf,ner,sub,coop] ,
        Wy=[agr,gea,rha,thex,asce,soe] ,
        comp_ks(Xx,Wx,Wy,[A,B,C,D,E],[Ax,Bx,Cx,Dx,Ex]) ,
        variable([Ax,Bx,Cx,Dx,Ex],Xxa,Xxb) ,
        Xxb=[X1,X2,X3,X4,X5] ,
        comp_lv(Xs,X1,X2,X3,X4,X5) ,
        alpheee(Xs,Xx) .

variable([],Xxb,Xxb) .
variable([X|Y],Xxa,Xxb) :-
        (not nonvar(X) ,
         Xx2=0
        ;
         X=..[X1,Xx1,_] ,
         Xx2=Xx1
        ) ,
        flatten([Xxa,[Xx2]],Xxc) ,
        variable(Y,Xxc,Xxb) .

alpheee(Xs,Xx) :-
        list(Xx,N1,N2,N),
        dispersion(N,Xx1,Xa) ,
        sqrt(Nn,Xp,Xn,Xx1) ,
        pook(Xs,Xx,Xn,Xa) .

data([],Xx,Xx) .
data([X|Y],Xy,Xx) :-
        play(X,Xx1) ,
        counter(Xx1,Xy,Xz) ,
        data(Y,Xz,Xx) .
```

```
play(X,Xx1) :-
        rule(X,G) ,
        discover(G,Yx,Xx,Xy) ,
        search(X,Xy,Xx1) .

discover([],Xy,Xy,Xy) .
discover([X|Y],Yx,Xx,Xy) :-
        ask(X,Xx,XXx) ,
        count(Xx,Yx,Xz) ,
        discover(Y,Xz,_,Xy) .

count(Xx,0,Xx) :-
        ! .
count(Xx,Xy,Xz) :-
        integer(Xx) ,
        integer(Xy) ,
        Xz is Xx+Xy ,
        Xy\=0 .

counter(Xx,0,[Xx]) :-
        ! .
counter(Xx,Xy,Xz) :-
        flatten([Xy,[Xx]],Xz) .

search(X,Xy,Xx2) :-
        table(X,Xz,Xx1) ,
        Xx=(Xy,Xy1) ,
        member(Xx,Xz) ,
        Xx2=..[X,Xx1,Xx] .

ask(X,Xx,Xy) :-
        ( rule0(X,Xy)
        ;
          rule1(X,Xy)
        ;
          rule2(X,Xy)
        ;
          rule3(X,Xy)
        ) ,
        tab(3) ,
        write(Xy) ,
        nl ,
        tab(3) ,
        write(:) ,
        ! ,
        complete(X,Xx,Xy) .

complete(X,Xx,Xy) :-
        repeat ,
        read(Y) ,
        ( Y=y ,
          Xx is 2
        ;
          Y=e ,
          Xx is 1
        ;
          Y=n ,
          Xx is 0
        ) ,
        ! .

flatten([],[]) .
flatten([X],X) .
flatten([X,Y|Z],V) :-
        append(X,Y,U) ,
        flatten([U|Z],V) .

append([],X,X) .
append([X|L1],L2,[X|L3]) :-


        append(L1,L2,L3) .

member(X,[X|_]) .
member(Y,[_|Z]) :-
        member(Y,Z) .


/*      COUNT LOOP A--E          */

countA(A,Ax) :-
        A=..[a,0,1] ,
        Ax=..[a,1,_] ,
        ! .
countA(A,Ax) :-
        A=..[a,A1,A2] ,
        A2 is A1+1 ,
        Ax=..[a,A2,_] .

countB(B,Bx) :-
        B=..[b,0,1] ,
        Bx=..[b,1,_] ,
        ! .
countB(B,Bx) :-
        B=..[b,B1,B2] ,
        B2 is B1+1 ,
        Bx=..[b,B2,_] .

countC(C,Cx) :-
        C=..[c,0,1] ,
        Cx=..[c,1,_] ,
        ! .
countC(C,Cx) :-
        C=..[c,C1,C2] ,
        C2 is C1+1 ,
        Cx=..[c,C2,_] .

countD(D,Dx) :-
        D=..[d,0,1] ,
        Dx=..[d,1,_] ,
        ! .
countD(D,Dx) :-
        D=..[d,D1,D2] ,
        D2 is D1+1 ,
        Dx=..[d,D2,_] .

countE(E,Ex) :-
        E=..[e,0,1] ,
        Ex=..[e,1,_] ,
        ! .
countE(E,Ex) :-
        E=..[e,E1,E2] ,
        E2 is E1+1 ,
        Ex=..[e,E2,_] .
```

```
comp_ks([],Wx,Wy,[Ay,By,Cy,Dy,Ey],[Ay,By,Cy,Dy,Ey]) .
comp_ks([X|Y],Wx,Wy,[A,B,C,D,E],[Ay,By,Cy,Dy,Ey]) :-
        X=..[Xx,Yx,Zx] ,
        comp_lp(Xx,Yx,Wx,Wy,A,B,C,D,E,Ax,Bx,Cx,Dx,Ex) ,
        comp_ks(Y,Wx,Wy,[Ax,Bx,Cx,Dx,Ex],[Ay,By,Cy,Dy,Ey]) .

comp_lp(Xx,Yx,Wx,Wy,A,B,C,D,E,Ax,Bx,Cx,Dx,Ex) :-
        ( Yx=1
        ;
          Yx=2
        ) ,
        ( member(Xx,Wx) ,
          countC(C,Cx) ,
          countD(D,Dx) ,
          Ax=A ,
          Bx=B ,
          Ex=E
        ;
          member(Xx,Wy) ,
          countC(C,Cx) ,
          countE(E,Ex) ,
          Ax=A ,
          Bx=B ,
          Dx=D
        ) .
comp_lp(Xx,Yx,Wx,Wy,A,B,C,D,E,Ax,Bx,Cx,Dx,Ex) :-
        ( Yx=4
        ;
          Yx=5
        ) ,
        ( member(Xx,Wx) ,
          countB(B,Bx) ,
          countE(E,Ex) ,
          Ax=A ,
          Cx=C ,
          Dx=D
        ;
          member(Xx,Wy) ,
          countB(B,Bx) ,
          countD(D,Dx) ,
          Ax=A ,
          Cx=C ,
          Ex=E
        ) .
comp_lp(Xx,Yx,Wx,Wy,A,B,C,D,E,Ax,Bx,Cx,Dx,Ex) :-
        Bx=B ,
        Cx=C ,
        Dx=D ,
        Ex=E ,
        Yx=3 ,
        countA(A,Ax) .


/*      TYPE SEARCH PROCESS 2 sorting        */

newsort([H|T],S) :-
        split(H,T,U1,U2) ,
        newsort(U1,V1) ,
        newsort(U2,V2) ,
        append(V1,[H|V2],S) .
newsort([],[]) .


split(H,[H1|T1],[H1|U1],U2) :-
        H1>H ,
        split(H,T1,U1,U2) .
split(H,[H1|T1],U1,[H1|U2]) :-
        H1=<H ,
        split(H,T1,U1,U2) .
split(_,[],[],[]) .
```

```
/*      TYPE SEARCH PROCESS 1          */

comp_lv(Xs,X1,X2,X3,X4,X5) :-
        newsort((X1,X2,X3,X4,X5),(W1,W2,W3,W4,W5)) ,
        ( W1>W2 ,
          singular_max(Xs,X1,X2,X3,X4,X5,W1)
        ;
          W1=W2 ,
          ( W1=X2 ,
            W1=X3
          ;
            W1=X4 ,
            W1=X5
          ) ,
          Xs=exception ,
          passion(exception)
        ;
          W1=W2 ,
          plural_max(Xs,X1,X2,X3,X4,X5,W1)
        ) .

/*      SINGULAR_MAX TYPE       */

singular_max(Xs,A1,B1,C1,D1,E1,W1) :-
        ( A1>=9 ,
          Xs=typeA ,
          passion(typeA)
        ;
          B1>=8 ,
          Xs=typeB ,
          passion(typeB)
        ;
          C1>=7 ,
          Xs=typeC ,
          passion(typeC)
        ;
          D1>=9 ,
          Xs=typeD ,
          passion(typeD)
        ;
          E1>=9 ,
          Xs=typeE ,
          passion(typeE)
        ) .
singular_max(Xs,A1,B1,C1,D1,E1,W1) :-
        ( A1=8 ,
          Xs=typeNA ,
          passion(typeNA)
        ;
          ( B1=7
          ;
            B1=6
          ) ,
          B1=W1 ,
          Xs=typeNB ,
          passion(typeNB)
        ;
          C1=6 ,
          C1=W1 ,
          Xs=typeNC ,
          passion(typeNC)
        ;
          ( D1=8
          ;
            D1=7
          ;
            D1=6
          ) ,
          D1=W1 ,
          Xs=typeND ,
          passion(typeND)
        ;
          ( E1=8
          ;
            E1=7
          ;
            E1=6
          ) ,
          E1=W1 ,
          Xs=typeNE ,
          passion(typeNE)
        ) .
singular_max(Xs,A1,B1,C1,D1,E1,W1) :-
        A1=W1 ,
        ( A1=7
        ;
          A1=6
        ;
          A1=5
        ) ,
        B1=<4 ,
        C1=<4 ,
        D1=<4 ,
        E1=<4 ,
        Xs=typeNNA ,
        passion(typeNNA) .
singular_max(Xs,A1,B1,C1,D1,E1,W1) :-
        A1=W1 ,
        ( A1=7
        ;
          A1=6
        ;
          A1=5
        ) ,
        ( A1=B1 ,
          C1=<4 ,
          D1=<4 ,
          E1=<4
        ;
          A1=C1 ,
          B1=<4 ,
          D1=<4 ,
          E1=<4
        ;
          A1=D1 ,
          B1=<4 ,
          C1=<4 ,
          E1=<4
        ;
          A1=E1 ,
          B1=<4 ,
          C1=<4 ,
          D1=<4
        ) ,
        Xs=typeNNA ,
        passion(typeNNA) .
singular_max(Xs,A1,B1,C1,D1,E1,W1) :-
        A1=W1 ,
        ( A1=7
        ;
          A1=6
        ) ,
```

```
        ( ( B1=5 ,
            E1=5
          ;
            B1=5 ,
            D1=5
          ) ,
          Xs=typeAB ,
          passion(typeAB)
        ;
          ( C1=5 ,
            D1=5
          ;
            C1=5 ,
            E1=5
          ) ,
          Xs=typeAC ,
          passion(typeAC)
        ;
          D1=5 ,
          Xs=typeAD ,
          passion(typeAD)
        ;
          E1=5 ,
          Xs=typeAE ,
          passion(typeAE)
        ) .


/*      PLURAL_MAX TYPE */

plural_max(Xs,A1,B1,C1,D1,E1,W1) :-
        ( A1=6 ,
          B1=6 ,
          A1=W1 ,
          Xs=typeNB ,
          passion(typeNB)
        ;
          A1=5 ,
          B1=5 ,
          A1=W1 ,
          Xs=typeAB ,
          passion(typeAB)
        ;
          A1=6 ,
          C1=6 ,
          A1=W1 ,
          Xs=typeNC ,
          passion(typeNC)
        ;
          A1=W1 ,
          ( A1=6 ,
            D1=6
          ;
            A1=5 ,
            D1=5
          ) ,
          Xs=typeAD ,
          passion(typeAD)
        ;
          A1=W1 ,
          ( A1=6 ,
            E1=6
          ;
            A1=5 ,
            E1=5
          ) ,
          Xs=typeAE ,
          passion(typeAE)
        ;
          B1=D1 ,
          B1=W1 ,
          Xs=typeAB ,
          passion(typeAB)
        ;
          B1=E1 ,
          B1=W1 ,
          Xs=typeNB ,
          passion(typeNB)
        ;
          C1=D1 ,
          C1=W1 ,
          Xs=typeAC ,
          passion(typeAC)
        ;
          C1=E1 ,
          C1=W1 ,
          Xs=typeNC ,
          passion(typeNC)
        ) .


/*      PRINT PERSONALITY TYPE      */

passion(typeA) :-
          tab(2) ,
          ruletA1(X) ,
          write('anata wa A gata
              ( Average type ) desu.') ,
          nl ,
          tab(2) ,
          write('seikaku wa --') ,
          write(X) ,
          ! .
```

（受理 昭和62年１月25日）