修士論文

構造化プログラミング型知識表現に基づく

プラント異常診断用エキスパートシステム

指導教官　井上紘一教授　熊本博光助手

論文提出者　奥村明俊

摘要

プラント故障診断用エキスパートシステムにおいて，知識をどのように表現するかという問題は非常に重要である。従来の診断システムでは，ＩＦ－ＴＨＥＮルールによって専門家の知識が表現されることが多い。しかし，ＩＦ－ＴＨＥＮルールは，知識の断片であり全体像が把握しにくい。また，そのシステムの処理速度が落ちるなどの短所もある。

そこで本論文では，新たなエキスパートシステムを実現するために，専門家の知識をＩＦ－ＴＨＥＮ－ＥＬＳＥ－ＥＮＤＩＦを単位とする構造化プログラミング型で表現する。また，この表現に基づくエキスパートシステム構築ツールを作成する。まず，構造化プログラミング型知識表現を用いることの意義を論じる。次ぎに，作成したエキスパートシステム構築ツールの概要を述べる。さらに，具体的な故障診断システム作成にツールを使用し，従来のＩＦ－ＴＨＥＮルールの欠点が解消されるなどの有効性を具体的に示す。診断の対象としては，船舶のエンジン冷却系異常と自動二輪車の故障診断を取り上げる。最後に構築用ツールによって作成された故障診断システムの概要について述べる。

目次　　頁

1章　緒言

自動化の進んだ現在においても，その道の専門家にしか解けない問題が多くある。エキスパートシステムとは，これら専門家の知識の利用により，計算機に高度の問題解決能力を持たせようとするものであり，その適用分野は，医学，工学，教育などの多岐にわたっている[1,2,3]。

ところで，知識の表現法として典型的なものは，ＩＦ－ＴＨＥＮルールとよばれるものである[2,3,4]。しかし，ＩＦ－ＴＨＥＮルールは，知識の全体像が把握しにくい，処理速度が落ちるなど，多くの短所を持っている。そこで，フレーム理論をはじめとして様々な手法が提案されているが，今なお研究段階のものが多い[5]。これらは，知識表現問題と言われるものである。エキスパートシステムは，表現された知識を利用して構築される。従って，エキスパートシステムの構築において，知識の表現問題は非常に重要である。

ＩＦ－ＴＨＥＮルールは，個々の知識は単純で理解しやすいが，知識が断片に細分化されているので全体像がわかりにくい。また関連した知識は，まとまりとして扱った方が理解しやすく素早く処理できるのは当然である。このように知識をまとまりとして表現することを知識の構造化といい，多くの研究がなされている。

そこで本論文では，専門家の知識を構造化プログラミング型に表現することを提案する。また，この構造化プログラミング型知識表現に基づいて診断用エキスパートシステムの構築用ツールを作成すると共に，この表現を用いることの有用性を明らかにする。

本論文の構成は次のとおりである。まず２章で，知識表現問題の位置づけを示すために，知識工学について概説する[6]。３章では，ＩＦ－ＴＨＥＮルールや知識の構造化表現について論じる。４章では，構造化プログラミング型知識表現の意義について述べ，５章でその有効性の一端を示すため，故障診断システムへの適用を示す。６章では，パーソナルコンピュータ上に作成した故障診断システムについて述べる。７章は今後の展望である。

２章　知識工学

問題解決のために人間が持つ経験的な知識を計算機上で利用しようとする研究分野を知識工学と呼ぶが，この章では，知識工学の研究課題について概説する。

２．１　知識表現の問題

知識表現とは，ある専門領域について，教科書にのっているような事実的知識と特定の専門家だけが持っているような経験的な知識の両方を計算機で利用可能となる様に形式化する問題をいう。これは知識ベースの設計問題ともいわれる。

２．２　知識利用の問題

知識利用とは，計算機可読なように形式化された知識を与えられた問題解決のためにどのように利用するかという問題である。これは推論エンジンの設計問題ともいわれる。知識表現と知識利用は，データとアルゴリズムのように，表裏一体の関係にある。

２．３　知識獲得の問題

知識獲得とは，対象とする問題領域について，事実的な知識と経験的な知識の両方を知識ベースに移植する問題をいう。専門家からの知識の抽出の支援，知識ベースの完全性や整合性の維持，学習による知識の自動獲得などが主要な課題である。

３章　知識の表現と利用

この章では，知識のＩＦ－ＴＨＥＮルールによる表現と構造化表現について述べ，知識を構造化プログラムで表現することの意義について論じる。

３．１　知識表現

知識の表現については，宣言的(declative)表現と手続き的(procedual)表現の２つの立場がある。宣言的表現は，ものを知ること(knowing that)という立場から，事実に関する知識とこれを利用する手続きを分離して表現するものである。手続き的表現は，方法を知ること(knowing how)という立場から，事実に関する知識と利用する方法を区別しないで，一体として扱おうとするものである[6]。表３．１にそれぞれの表現の特徴を示す。

宣言的表現	手続き的表現
① 推論方法の変更による多目的利用が可能 → 柔軟性・経済性	① 問題解決プロセスのような手続き的知識 → 自然に表現可能
② 推論方法が独立 → 推論の完全性	② 推論過程を陽に指示 → 自然な推論
③ 知識と制御が分離 → モジュール性	③ 知識と制御が結合 → 知識の相互作用

表３．１　宣言的表現と手続き的表現

３．２　知識表現の条件

知識を計算機が利用可能となる様に形式化するためには，次のような条件を満たすように考慮しなければならい。

１）　表現能力

モデル化の対象となっている問題領域に関連する事実的および経験的知識の全てを表現できるだけの記述能力を持つこと。

2）　推論能力

知識ベース上に構造化された知識群を操作して，新しい知識を推論できる推論機構が実現しやすいこと。

3）　問題解決能力

問題解決の方向に推論の焦点を合わせて，推論機構を効率的に働かせうること。

4）　獲得能力

新しい知識の獲得，知識ベースの整合性の維持，知識ベースの拡張などが容易に行えること。

3．3　IF－THENルールによる表現

知識表現には，論理表現，意味ネットワーク，IF－THENルール，フレーム表現等がある。現在広く用いられているIF－THENルールについて述べる。

IF－THENルールは，現在では，エキスパートシステムの構築言語として，最もポピュラーな知識表現である。

IF－THENルールは，”前提→結論”または　”条件→行動”の形式で記述される知識モジュールであって，以下の様な形式をしている。

IF　[命題P_1] & [命題P_2] &　. . . & [命題P_n]
THEN　[命題Q]　　（3．1）

IF－THENルールに基づく推論システムには，つぎのような特徴がある。まず長所として，

1）知識の変更，追加が容易である。

2）システムが持っている知識を簡単に知ることができる。

3）入力データに応じて柔軟な処理を行える。

4）システムが結論を得る過程で用いた規則をたどることによって，なぜそのような結論を下したかをユーザーに説明することができる。もしその理由が納得できなければ，規則を変更すればよい。

つぎに，短所として，

5）規則の実行だけでなく，どの規則を適用したらよいかを決めるための処理が必要であるので，処理速度がおちる。

6）規則の集合から，システムの動作を予測できない。

7）一連の手続きを直接表現することができない。

8）知識の全体像が把握しにくい。

9）知識の信頼性の検証が困難である。

10）ルールの数が増すに連れて，知識の管理が飛躍的に困難となる。

といったことが挙げられる。

3．4　知識の構造化表現

人間が日常行っている常識的な思考は効率が良い。もし，常識に相当する論理式を全て計算機に与えたとしても，人間と同じように効率良く結論を引き出すことは困難であろう。人間と同じ程度の能力と速度を得るためには，知識の単位がもっと大きく，しかもそれぞれの知識のまとまり(chunk)が構造化されていなければならないと考えられる。つまり対象分野によっては，知識をＩＦ－ＴＨＥＮルールのように細分化しないで，項目ごとにある程度まとまった知識を与えておくほうがよいことがある。

このような考え方は，人間の問題解決を心理学的に説明しようとする研究や，自然言語理解の研究から生まれてきた。実際にこの考えに基づいたシステムの知識表現の枠組みとして，フレーム，意味ネットワーク，スクリプト，ユニットなどが使われている[2,3,5]。

４章　構造化プログラミング型知識表現

この章では，ブロックＩＦ文によって知識の構造化を行い，さらにモジュール化された知識群をサブルーチンとして扱う構造化プログラミング型知識表現について論じる。

４．１　知識の手続き化

ある事柄について判断を下すとき，熟練者と素人の違いは，たんにその知識の有無，多少ではない。例えば，素人に熟練者の知識としてのＩＦ－ＴＨＥＮルールの羅列を与えて故障診断をさせても，熟練者と同じように効率良く診断することはできないであろう。それは，素人は与えられたルールを試行錯誤的に追っていくのに対して，熟練者は羅列でなく整理された知識のまとまりを一気に処理するからである。つまり，素人は与えられたルールをしらみつぶしに処理するのに対して，熟練者は知識の全体像を見通して選択的に推論するからである。

ＩＦ－ＴＨＥＮルールを用いた従来の故障診断システムも，ルールの照合(matching)をワンステップごとに行っており，計算機というハードウエアによって処理速度を向上させるという面が強い。また，実行前にルールをリンクするという手法もあるが，そのようにリンクされた知識は理解しがたい場合が多い。そこで，知識表現の段階から知識を論理の流れに従って手続き化しておいたほうが良いという考えが生まれる。このことによって，知識は理解しやすくなるし，これを利用するエキスパートシステムの処理速度も向上しよう。

４．２　構造化プログラミングによる知識表現

手続き的表現で最も優れたものは，構造化プログラミング型知識表現であろう。そこで知識群を図４．１のような形式のブロックＩＦ文で表現する，また，一まとまりの知識群をサブルーチンとして登録できるようにする。

図４．１を従来のＩＦ－ＴＨＥＮルールで表すと，

ＩＦ　[命題P_1] ＆ [命題P_2] ＴＨＥＮ　[命題Q_1]．
ＩＦ　[命題Q_1] ＆ [命題P_3] ＴＨＥＮ　[命題Q_2]．
ＩＦ　[命題P_1] ＆ ＮＯＴ [命題P_2] ＴＨＥＮ　[命題Q_1]．

となってしまい，構造化されたものに比べ遥かにわかりにくくなる。

図４．１のように表現された知識群を１つのサブルーチンとして登録することもできる。サブルーチンの参照は　ｄｏ　サブルーチン名　としてｔｈｅｎ部の後に記述するようにする。ｅｌｓｅ部とｅｌｓｅの後の命題は必ずしも必要ではない。なお縦線は，つながりを明示するためのもので実際には引く必要はない。

```
if  [命題P1]
  if  [命題P2]
  then  [命題Q1]
    if  [命題P3]
    then  [命題Q2]
      do  サブルーチン名
    endif
      ・
      ・
      ・
  else  [命題Q1]
    if  [命題Pj]
      ・
      ・
    endif
  endif
endif
```

図４．１　構造化プログラミングによる知識表現の形式

４．３　ルールの羅列と構造化表現の比較の具体例

まず，船舶のエンジン冷却系の故障診断システムを例に挙げて説明する。

従来の知識表現では，式（３．１）の形式のＩＦ－ＴＨＥＮルールで，熟練者の原因探索知識を表現している。ルールの形式を

ＩＦ　　［Ｐ－ＳＴＡＴＥ］＆［ＦＡＣＴ－１］＆［ＦＡＣＴ－２］
ＴＨＥＮ　［Ｃ－ＳＴＡＴＥ］

と書き改めたときのルールベースを図４．２に示す。ＲＵＬＥ－ＮＡＭＥは，それぞれのルールの名前である。例えば，ＲＵＬＥ２は，

ＩＦ　［命題ＳＴ１］＆［命題ＦＴ２］　ＴＨＥＮ　［命題ＳＴ２］

というＩＦ－ＴＨＥＮルールである。この知識表現は図４．３のようなＡＮＤ／ＯＲ木でも表現できる。同図において，ルール１１と１２は次のとおりである。

RULE-NAME	P-STATE	C-STATE	FACT-1	FACT-2
RULE 1	ST 0	ST 1	FT 1	
RULE 2	ST 1	ST 2	FT 2	
RULE 3	ST 2	ST 3	FT 3	
RULE 4	ST 2	ST 4	FT 4	
RULE 5	ST 4	ST 5	FT 5	
RULE 6	ST 4	ST 6	FT 6	
RULE 7	ST 4	ST 7	FT 7	
RULE 8	ST 4	ST 8	FT 8	
RULE 9	ST 4	ST 9	FT 9-1	FT 9-2
FULE 10	ST 1	ST 10	FT 10	
RULE 11	ST 10	ST 11	FT 11	
RULE 12	ST 10	ST 12	FT 12	
RULE 13	ST 12	ST 13	FT 13	
RULE 14	ST 12	ST 14	FT 14	
RULE 15	ST 14	ST 15	FT 15	
RULE 16	ST 14	ST 16	FT 16	
RULE 17	ST 14	ST 17	FT 17	
RULE 18	ST 17	ST 18	FT 18	
RULE 19	ST 17	ST 19	FT 19	
RULE 20	ST 17	ST 20	FT 20	
RULE 21	ST 17	ST 21	FT 21	
RULE 22	ST 17	ST 22	FT 22-1	FT 22-2

図４．２　ＩＦ－ＴＨＥＮルールによる表現

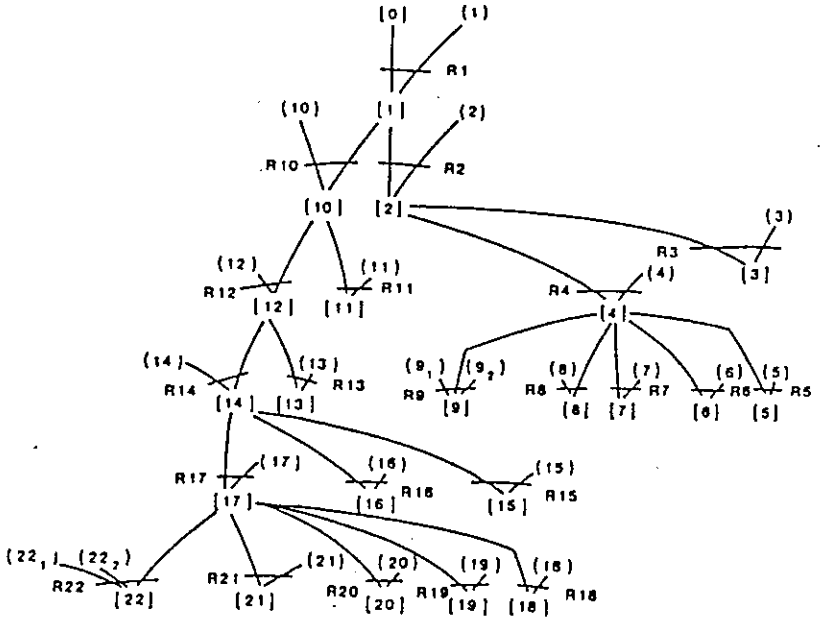


図４．３　ＡＮＤ／ＯＲ木表現

ルール１１：[10] AND (11) → [11]: R11

ルール１２：[10] AND (12) → [12]: R12

[10]：純水冷却ループの冷却材は充分である。

(11)：暖機弁が開いている。

[11]：暖機弁の閉じ忘れがエンジン過熱の原因である。

(12)：熱交換器から出てきた純水の温度が高い。

[12]：熱交換器が充分に働いていない。

熟練者は，Ｒ１１，Ｒ１２ の順でルールを作成した。まず暖機弁開閉の有無をＲ１１で確かめ，これが正常に閉じている場合にのみ，Ｒ１２で「熱交作動不十分」と結論している。ルールを１１，１２の順で適用する限りにおいては，特に問題は生じない。しかし，図４．３のＡＮＤ／ＯＲ木によれば，これら２つのルールに優先順位はなく，左からルール１２，１１の順の適用も許されている。この場合，エンジン過熱時には事実（１２）は常に成立するので，暖機弁開閉の有無にかかわらず，状態［１２］が真となる。つまり，暖機弁が開いていても熱交

換器の方へ探索を進めてしまう。上記の不都合が生じる原因は，ルール１２作成時に，熟練者が「暖機弁閉」を暗黙に仮定したことによる。ルールがある順序で作成される時，この種の暗黙の仮定が無意識になされることが多く，結果として得られるＡＮＤ／ＯＲ木は熟練者の知識を正確に表現しないことになる。

熟練者の性癖とも言える暗黙の仮定を忠実に表現するには，単なるＩＦ－ＴＨＥＮルールではなく，構造化プログラミングにみられるようなブロックＩＦ文，つまりＩＦ－ＴＨＥＮ－ＥＬＳＥ－ＥＮＤＩＦ構文のほうが適している。図４．３のＡＮＤ／ＯＲ木を構造化プログラム風に書き改めると図４．４となる。暗黙の仮定もＥＬＳＥ文で明示的に表現され，ＡＮＤ／ＯＲ木の場合のような混乱は生じない。熟練者の知識が単なるルールの羅列では表現しにくいという一例である。

次ぎに，船舶の主機起動系を例に取り，安全装置の正常作動や誤作動によるプラント停止時の異常診断を考えてみる。このときには，次の２種類のルールの集合があらわれる。

異常状態探索ルール：　安全装置の正常作動を発動するようなプラント異常事象の探索ルール群。

誤作動帰結ルール　：　プラントの異常事象が見つからない場合に，安全装置誤作動原因を帰結するルール群。

図４．５は，船舶の油圧ポンプが起動後すぐに停止することの診断知識を構造化プログラミング型で表現したものである。ここに，油圧ポンプＯＰ－１が起動するがすぐに停止し，油圧が２５ＫＧ／平方センチに達していないとする。図中のＡのルールは，プラント異常状態探索ルールである。また，Ｂのルールは安全装置の誤作動帰結ルールである。

通常ならば，油圧が２５ＫＧ／平方センチに達するまで９秒かかり，１０秒後に油圧がこの圧力に達していないときには安全装置（油圧スイッチ）が作動してポンプをとめるとする。もし５秒後にスイッチが作動したのであれば，当然油圧は２５ＫＧ／平方センチに達しておらず起動後すぐ停止する。これら２種類のルール群もＩＦ－ＴＨＥＮ－ＥＬＳＥ－ＥＮＤＩＦ構文で自然に結合できる。

```
##   エンジン冷却系診断システム

if [エンジンが過熱している]
  if [エンジン自身には原因がない]
  then [冷却系が異常]
    if [純水冷却系の圧力が低い ]
    then [純水系の冷却水が不足  ]
      if [純水ポンプが動いていない ]
      then [純水ポンプの停止 ]
      else [純水ポンプ作動のもとでの冷却水不足]
        if [純水ポンプ吸込側の圧力が低い ]
        then [純水ポンプ吸込弁が閉じている ]
        endif
        if [純水ポンプ吐出側の圧力が高い ]
        then [純水ポンプの吐出弁が閉じている ]
        endif
        if [純水ポンプのモータの電流計が異常値を示す ]
        then [純水ポンプのモータのパワー減少 ]
        endif
        if [充填タンクの水位が低い ]
        then [充填タンクの水量が不足し，冷却系に気泡ができた ]
        endif
        if [純水ポンプ吐出側の圧力が低い ]
          if [純水ポンプ吸込側の圧力は正常 ]
          then [純水ポンプの羽が故障 ]
          endif
        endif
      endif
    endif
    if [純水冷却系の圧力は正常 ]
    then [純水系の冷却水は十分である  ]
      if [環機弁が開いている]
      then [環機弁の閉め忘れ]
      else
        if [熱交換器から出てきた純水の温度が高い ]
        then [熱交換器が十分働いていない]
          if [熱交換器から出てきた海水の温度が低い ]
          then [熱交換器が汚れていて交換率が低い]
          endif
          if [熱交換器から出てきた海水の温度が高い ]
          then [海水系の冷却水が不足  ]
            if [海水制御弁が閉じている ]
            then [海水制御弁の故障 ]
            endif
            if [海水ポンプが動いていない ]
            then [海水ポンプの停止 ]
            else [海水ポンプ作動のもとでの冷却水不足  ]
              if [海水系の圧力が高い ]
              then [海水放出弁が閉じている ]
              endif
              if [海水ポンプ吸込側の圧力が低い ]
              then [ストレーナのつまりか，海水ポンプ吸込弁が閉じている]
              endif
              if [海水ポンプ吐出側の圧力が高い ]
              then [海水ポンプ吐出弁が閉じている ]
              endif
              if [海水ポンプのモータの電流計が異常値を示す ]
              then [海水ポンプのモータのパワー減少 ]
              endif
              if [海水ポンプ吐出側の圧力が低い ]
                if [海水ポンプ吸込側の圧力は正常 ]
                then [海水ポンプの羽が故障 ]
                endif
              endif
            endif
          endif
        endif
      endif
    endif
  else [エンジン自身に原因がある]
  endif
endif
```

図４．４　構造化プログラミング型知識表現

```
##油圧ポンプ起動後すぐ停止
  if［油圧ポンプＯＰ－１が起動するがすぐ停止］
  then［インタロック作動により油圧ポンプＯＰ－１が起動後すぐ停止］
    if［油圧ポンプＯＰ－１の油圧が２５ＫＧ／平方センチに達しない］
    then［油圧低による油圧ポンプＯＰ－１のトリップ］
-------------------------------------------------------------------
      if［主操作レバ駆動システムの油圧タンクのレベル低］
      then［主操作レバ駆動システム油圧タンクのレベル低による燃料運転不可能］
      else
        if［油圧ポンプＯＰ－１の吸い込み側圧力低］
        then［油圧ポンプＯＰ－１の吸い込み弁閉あるいはストレーナの詰まりのため燃
             料運転不可能］
        else
A         if［油圧ポンプＯＰ－１の周辺がら油もれ］
          then［油圧ポンプＯＰ－１からの油漏れのため燃料運転不可能］
          else
            if［油圧ポンプＯＰ－１から異音が聞こえる］
            then［主操作レバ駆動システム油圧配管内の泡のため燃料運転不可能］
            else
-------------------------------------------------------------------
              if［油圧スイッチ作動］
B             then［油圧スイッチの誤作動のため燃料運転不可能］
              endif
-------------------------------------------------------------------
            endif
          endif
        endif
      endif
      if［油圧タイマの作動］
      then［油圧タイマの早期作動のため燃料運転不可能］
      endif
    endif
  endif

endif
```

図４．５　油圧ポンプ起動後すぐ停止する場合の診断ルーチン　（抜粋）

```
##着火失敗による燃料運転不可能を診断するルールベースです．

if［主機の冷却水温度低］
then［暖機不足のための燃料運転不可能］
endif
##
if［燃料タンクの元弁が閉］
then［燃料タンクの元弁が閉のために燃料運転不可能］
##元弁は開であります．
else
  if［燃料ブースターポンプの吸い込み側の圧力低］
  then［燃料フィルタの詰まりのために燃料運転不可能］
##
  else
    if［燃料噴射ノイズの欠如］
    then［燃料噴射弁閉のために燃料運転不可能］
    endif
  endif
endif
##
```

図４．６　着火失敗による燃料運転不可能を診断するルーチン　（抜粋）

最後に，エンジンの着火失敗による燃料運転不可能を診断する例を図４．６に挙げる。ここで，燃料タンク，元弁，燃料フィルタ，吸い込み弁，ブースタポンプ，噴射弁は直列的に結ばれている。このような直列バルブ系の閉塞などでは，異常部分を上流部から順次探索していくのでルールの定型的パターンが現れる。これは構造化プログラミング中のｉｆ－ｔｈｅｎ－ｅｌｓｅのパターンに反映されてくる。

４．４　構造化プログラミング型知識表現の特徴

３．３で述べたＩＦ－ＴＨＥＮルールと比較すると構造化プログラミング型知識表現には，次のような特徴がある。

１）知識をプログラムとしてコンパイルでき，処理速度が向上する。

構造化プログラムで表現された知識は，容易にプログラムとしてコンパイル可能である。そのプログラムの実行は，データ駆動型推論（前向き推論，forward reasoning）の実行にほかならない。構造化プログラムで表現することにより，処理速度は著しく向上する。

２）知識の全体が理解しやすい。

構造化プログラムで表現された知識は，一連の手続きを直接表現したものでもある。エキスパートシステムの重要な性質として，システムの見通しの良さを意味する　”透過性(transparency)”を向上させると言える。推論中に構造化プログラムで表現された知識を直接表示することによって推論の現在位置や推論の展開を明示できる。

３）ルールの特質(character)を明示する。

ルールの相互関係，知識全体における位置が明示されるので，命題の従属関係を直接表現でき，どのルールが並べ換え可能であるかとか，違った構造で等価な知識は何かといったことも検討できる。

４）モジュール化

知識をサブルーチンとしてモジュール化できる。これによって大きな知識ベースでも取り扱いが容易になる。一つのモジュールの大きさは，その知識の性質や対象によって適切に定める。

５）複数原因の探索

原因探索が終了した後もプログラムの次のステップを実行することにより，探

索を続行し他の原因を調べることができる。

6）優先順位

構造化プログラムで表現された知識は，上位からシーケンシャルに実行されて行くため，表現された順がルールの優先順位である。

7）ソフトウエア工学との関係

知識獲得問題はプログラム開発問題に，知識の正当性検証はプログラムのそれにというふうに，ソフトウェア工学の成果を援用できる可能性がある。

8）知識獲得が容易

4．1で述べたように，暗黙の条件を無理なく明示的に表現でき，実際の故障診断のチェックリストを容易に構造化プログラミング型知識表現で表すことができる。

9）コメントの活用

事実の確認方法，推論の現在位置，プラントの復旧手段，ルールの正当性などをコメントとして記述することで知識の理解度が深まる。

以上主な特徴を挙げたが，実際知識を構造化プログラムで表現することは，ＩＦ－ＴＨＥＮルールの短所を解消するものである。構造化プログラム表現は知識工学でポピュラーなＡＮＤ／ＯＲ木の自然な拡張であり，原因探索知識の獲得と表現に適していると思われる。

５章　診断システムへの適用

この章では，診断システム構築ツールとこのツールによって作成された故障診断システムについて述べる。診断システム構築ツールは，構造化プログラムで表現された知識をこれと等価なＣ言語のプログラムへ変換し，診断システムを構築するものである。

５．１　概要

構造化プログラムで表現された知識を知識ベースとして故障診断システムを構築するツールを作成した。このツールは，Ｃ言語でかかれた故障診断システムを構築するもので，このツール自身もＣ言語でかかれている。故障診断の対象として，自動二輪車と船舶のエンジン冷却系を取り上げる。自動二輪車の故障診断知識を構造化プログラムで表現した例を付録２にしるす。サブルーチンは，ｄｏ　サブルーチン名　であらわされている。

構造化プログラムで表現された知識の編集は，エディタを用いてテキストとして編集する。

５．２　推論（reasoning）

推論は，データ駆動型推論（前向き推論，forward reasoning）で行われる。故障診断構築ツールによって，構造化プログラムで表現された知識は，等価なＣ言語のプログラムに変換される。図４．４をＣ言語に変換した例を付録３にしるす。

ｄｉａｇ（）という関数がこのツールによって知識が変換されたものであり，診断に必要な他の関数にリンクされる。このプログラムの実行は，データ駆動型推論の実行にほかならない。

５．３　バックワード(backward)機能

ユーザーはあくまで専門家ではないので様々な説明機能と共に推論のワンステップごとに訂正可能であること(revocability)が必要である。

本システムでは，推論の任意の段階で前進と後退(backward)が可能である。これにより熟練者の知識上を自由に”渡り歩くこと(loitering)”ができる。

後退は，変換の際にそれぞれのｉｆ－ｔｈｅｎ－ｅｌｓｅ文をｗｈｉｌｅ文で囲い，ｅｎｄｗｈｉｌｅをブロックの最後におくことによって可能にしている。

ｉｆ部の前のｗｈｉｌｅ文の条件部におかれている関数　ｆｆａｃｔ（ｊ）は，ｉｆ部にある命題の真偽を問いかける関数である。ここでバックワードを示すコマンド（ｂ）が入力されるとこのループに入ることが出来なくなる。そして，もう一段外を囲うｗｈｉｌｅ文によってそのｗｈｉｌｅ文の条件部へと戻ってくる。この条件部には，１つ前のｉｆ部にある命題の真偽を問う関数ｆｆａｃｔ（ｉ）があり，すなわちワンステップ後退したことになる。

５．４　説明機能

推論中に推論状況を構造化プログラム上に色分けして表示することで推論の過程や展開を示す。また，知識を構造化プログラムで表現するさいに，よりわかりやすくするためにコメント文や空行を認めている。コメント行は，＊＊を行中に置く。事実の確認方法，推論の現在位置，プラントの復旧手段，ルールの正当性などを記述することでさらに充実した知識ベースとなる。図４．４の知識にコメント文としてさらに知識を組み入れた例を図５．１にしるす。

５．５　自動推論機能(Quick Table)

ＩＦ部の命題の真偽をファイルに入力することによって，推論を自動的に行うことができる。センサから直接的に真偽を入力できれば，即座に診断を下すことができる。このファイルをクイックテーブルとよぶことにする。

また診断時の応答によって入力された真偽をそのファイルにセーブすることもできる。

５．６　データ検索機能

メタ知識をはじめとした様々な知識の管理に，データベースシステム　日本語ｄＢＡＳＥⅡを用いることもできる。推論中の任意の段階で，データベースにアクセスして様々な情報を得ることができ，説明機能を充実させている。また，ユーザーは，独自のファイルを持つことができる。

```
**   エンジン冷却系診断システム

if [エンジンが過熱している]
  **  エンジン自身か，冷却系に異常がある
  if [エンジン自身には原因がない]
  **  純水系か，熱交換器か，海水系に異常がある
  then [冷却系が異常]
    if [純水冷却系の圧力が低い ]
      **  圧力計 p160 を調べて下さい．１５kg／cm² 以下だと低いです
    then [純水系の冷却水が不足  ]
    **   純水系に異常がある
      if [純水ポンプが動いていない ]
        **  ポンプに触れてみると，良く判かります
      then [純水ポンプの停止 ]
      else [純水ポンプ作動のもとでの冷却水不足]
      **   ポンプ自身か，純水系閉塞か，冷却水補給不足が原因である
        if [純水ポンプ吸込側の圧力が低い ]
          **  圧力計 p161 を調べて下さい．２kg／cm² 以下だと低いです
        then [純水ポンプ吸込弁が閉じている ]
        endif
        if [純水ポンプ吐出側の圧力が高い ]
          **  圧力計 p162 を調べて下さい．２０kg／cm² 以上だと高いです
        then [純水ポンプの吐出弁が閉じている ]
        endif
        if [純水ポンプのモータの電流計が異常値を示す ]
          **  電流計 a15 を調べて下さい．正常値は１５から２０ｱﾝﾍﾟｱまでです
        then [純水ポンプのモータのパワー減少 ]
        endif
        if [充填タンクの水位が低い ]
          **  水位計  L2 を調べて下さい．１ﾒｰﾄﾙ以下だと低いです
        then [充填タンクの水量が不足し，冷却系に気泡ができた ]
        endif
        if [純水ポンプ吐出側の圧力が低い ]
          **  圧力計 p101 を調べて下さい．１５kg／cm² 以下だと低いです
          if [純水ポンプ吸込側の圧力は正常 ]
          **  圧力計 p111 を調べて下さい  正常値は０．５から１．０kg／cm²までです
          then [純水ポンプの羽が故障 ]
          endif
        endif
    endif
  endif
  if [純水冷却系の圧力は正常 ]
    **  圧力計 p160 を調べて下さい．正常値は１５kg／cm² 以上です
  then [純水系の冷却水は十分である ]
  **  冷却水が，誤って熱しられているか，十分に冷却されていない
    if [暖機弁が開いている]
      **  インジケータランプ 123 が赤ランプだと開いています
    then [暖機弁の閉め忘れ]
    else
      if [熱交換器から出てきた純水の温度が高い ]
        **  温度計 T15 を調べて下さい．１２０度以上だと高いです
      then [熱交換器が十分働いていない]
      **  熱交換器か海水系に原因がある
        if [熱交換器から出てきた海水の温度が低い ]
          **  温度計 T16 を調べて下さい．１００度以下だと低いです
        then [熱交換器が汚れていて交換率が低い]
        endif
        if [熱交換器から出てきた海水の温度が高い ]
          **  温度計 T16 を調べて下さい．１８０度以上だと高いです
        then [海水系の冷却水が不足  ]
        **  海水系に原因がある
          if [海水制御弁が閉じている ]
            **  インジケータランプ 122 が赤ランプだと閉じています
          then [海水制御弁の故障 ]
          endif
          if [海水ポンプが動いていない ]
            **  ポンプに触れてみると，良く判かります
          then [海水ポンプの停止 ]
          else [海水ポンプ作動のもとでの冷却水不足  ]
          **  ポンプ自身か，海水系閉塞が原因である
            if [海水系の圧力が高い ]
              **  圧力計 p201 を調べて下さい．２kg／cm² 以上だと高いです
            then [海水放出弁が閉じている ]
            endif
            if [海水ポンプ吸込側の圧力が低い ]
```

図５．１　コメント文を入れた構造化プログラミング型知識表現　（抜粋）

6章　パーソナルコンピュータ上に作成した診断システム

この章では，パーソナルコンピュータ上に作成した診断システム構築ツールとそれによって構築された診断システムについて実際例を挙げて述べる。

6．1　システムの構成

1）知識ベース

構造化プログラムで表現された知識自身をデータファイルとする。拡張子は，ＤＡＴである。そのファイルからＩＦ部とＴＨＥＮ部を集めたファイルが作られる。拡張子は，それぞれＦＡＴとＳＡＴである。

2）推論エンジン

構造化プログラムで表現された知識からＣ言語による推論プログラムが作られ，コンパイルされて実行型ファイルとなる。拡張子は，それぞれＣとＥＸＥである。

ｔｒａｎｓ　データファイル名

で実行型ファイルが作成される。構造化プログラムで表現された知識の書式にミスがあれば，コンパイル時にＣプログラムのエラーとしてメッセージが表示される。

3）ワーキングメモリ

パーソナルコンピュータのＲＡＭ(Random Access Memory)とフロッピィディスクを使用する。

4）データベース

現在観測可能事実の確認方法などを記録したファイルがあるが，容易に増やすことが可能である。

5）エディタ

知識ベースの編集用にＥｄｉｔｏｒ　Ｅが用意されている。ｄＢＡＳＥⅡのファイルの編集は，ｄＢＡＳＥⅡの編集コマンド　ｅｄｉｔ，ｂｒｏｗｓｅを用いる。

6．2　システム環境

このシステムは，ＮＥＣ　ＰＣ－９８０１の ＭＳ－ＤＯＳ　Version 2.0 上で作動する。コンパイラは，ＰＥＮＧＵＩＮ　Ｃ86　Ｃ　ＣＯＭＰＩＬＥＲ　を用いる。データベース管理システムとして日本語ｄＢＡＳＥⅡ，テキスト編集エディタとして　Ｅｄｉｔｏｒ　Ｅ を使う。

６．３　操作

ＭＳ－ＤＯＳ のシステムを立ちあげて ” ｔｒａｎｓ　データファイル名” と入力すると必要なファイルが作成され，ソースファイルがコンパイルされる。実行型ファイルが作成されると，そのファイル名を入力することにより推論を始めることができる。

推論中に使用可能なコマンドについて説明する。

操作は対話形式で行われ，すべてワンキー入力である。大文字，小文字のどちらでも良い。システムは，命題の真偽をオペレータに質問する。オペレータは，表示される命題が真であれば ” ｔ” 偽であれば ” ｆ” を入力する。

その他以下のようなコマンドがある。

ｂ：真偽応答の取り消し。前の状態に戻る。

ｃ：現在の推論ブロックを終了して，その後のブロックの推論を続行し他の原因を探索する。ブロックが無ければ終了する。

ｄ：ｄＢＡＳＥⅡのモードにはいる。様々なファイルとのアクセスが可能となる。” ｑｕｉｔ” と入力するともとに戻る。

ｅ：Ｅｄｉｔｏｒ Ｅのモードに入る。クイックテーブルに真偽を入力すると自動推論が可能である。０，１，２がそれぞれ命題の偽，真，未定である。

ｇ：推論状況を構造化プログラムで表現された知識で表示する。緑色はコメント文，黄色は真の命題，紫色は偽の命題，背景白色は現在位置をしめす。現在位置が，結論状態である時は背景赤色でしめす。２，８キーでそれぞれ，スクロールダウン，スクロールアップする。ＥＳＣキーで終了する。

ｈ：使用可能コマンドの説明。

ｉ：事実の確認方法などのデータを表示する。

ｍ：結論がでた後，その結論についてより詳細な診断サブルーチンがあれば，そのルーチンの探索をおこなう。

ｑ：現在推論中のルーチンを終了する。

ｓ：現在までの真偽応答をセーブする。

６．４　実行例

船舶エンジン冷却系と自動二輪車の故障診断システムの作成及び診断例をしるす。これは，画面コピーであり，実際はメッセージによって色分けされている。オペレータとの応答と共に解説を加える。

```
B>trans colc

 データファイルからＣ言語のプログラムを作成し，コンパイルします．
 診断に必要なファイル　colc.fat, colc.sat , colc.doc が作成されました
 Ｃ言語による診断システム　　colc.c が作成されました

  colc.C をコンパイルします

PENGUIN 漢字 C COMPILER V2.1g
copyright (c) 1984 penguin soft (Kyoto Japan)
C:A.c:
------------------------------------------------------------------------------
文法上問題はありません
The Microsoft MACRO Assembler
Version 1.12, Copyright (C) Microsoft Inc. 1981,82,83



Warning Severe
Errors  Errors
0       0

B>LINK colc a:fdiag a:dbas a:help ,colc,NUL,C:LIB+C:MATH

   Microsoft Object Linker V2.00
(C) Copyright 1982 by Microsoft Inc.

Symbol defined more than once: _MXFMEM in file C:LIB.LIB(_CONFIG)

There was 1 error detected.

B>ECHO



B>colc

              colc ** 診断開始 **

   **   エンジン冷却系診断システム



Fact Number 0 *** colc
[ エンジンが過熱している ]

                                              command please  : t

Fact Number 1 *** colc
[ エンジン自身には原因がない ]

                                              command please  : h

    b -- 前の質問に戻ります．戻らなければ再入力して下さい
    c -- 次のブロックの推論を行います．ブロックが無ければ終了します
    d -- dＢＡＳＥⅢのモードに入ります
    e -- エディトモードに入り，クィックテーブル等の編集ができます
    f -- 命題を偽とみなします
    g -- 推論の経過および方向を表示します．上下　２－８，ＥＳＣで終了します
    h -- コマンドの説明を行います
    l -- 確認方法などのデータを提供します
    q -- 現在推論中のルーチンを終了します
    s -- 現在までの応答をセーブします
    t -- 命題を真とみなします
```

システム構築ツール
ｔｒａｎｓによって
船舶エンジン冷却系
診断システムを構築
する。

書式にミスがあれば，
ここでエラーメッセー
ジが表示される。

システム作成に成功し
た。

船舶エンジン冷却系
に関する診断を開始す
る。

命題を真とみなし，
”ｔ”を入力する。

使用可能なコマンドを
表示する。

```
Fact Number 1 *** colc
[ エンジン自身には原因がない ]

                                                    command please  : t

State Number 0  *** colc
[ 冷却系が異常 ] が推論されました

Fact Number 2 *** colc
[ 純水冷却系の圧力が低い  ]

                                                    command please  : l

   data   圧力計 p160 を調べて下さい．  １５kg／cm² 以下だと低いです

Fact Number 2 *** colc
[ 純水冷却系の圧力が低い  ]

                                                    command please  : f

Fact Number 10 *** colc
[ 純水冷却系の圧力は正常  ]

                                                    command please  : g

   **   エンジン冷却系診断システム


 if [エンジンが過熱している]
   **  エンジン自身か，冷却系に異常がある
   if [エンジン自身には原因がない]
   **  純水系か，熱交換器か，海水系に異常がある
   then [冷却系が異常]
     if [純水冷却系の圧力が低い ]
       **  圧力計 p160 を調べて下さい．１５kg／cm² 以下だと低いです
     then [純水系の冷却水が不足  ]
     **   純水系に異常がある
       if [純水ポンプが動いていない ]
         **  ポンプに触れてみると，良く判かります
       then [純水ポンプの停止 ]
       else [純水ポンプ作動のもとでの冷却水不足]
       **   ポンプ自身か，純水系閉塞か，冷却水補給不足が原因である
         if [純水ポンプ吸込側の圧力が低い ]
           **  圧力計 p161 を調べて下さい．２kg／cm² 以下だと低いです
         then [純水ポンプ吸込弁が閉じている ]
         endif
         if [純水ポンプ吐出側の圧力が高い ]

                 ●
                 ●

         endif
       endif
     endif

     if [純水冷却系の圧力は正常 ] <-------------------
       **  圧力計 p160 を調べて下さい．正常値は１５kg／cm² 以上です
     then [純水系の冷却水は十分である ]
     **  冷却水が，誤って熱しられているか，十分に冷却されていない
       if [環境弁が閉いている]
         **  インジケータランプ 123 が赤ランプだと閉いています
       then [環境弁の閉め忘れ]
       else
         if [熱交換器から出てきた純水の温度が高い ]
           **  温度計 T15 を調べて下さい．１２０度以上だと高いです
         then [熱交換器が十分働いていない]
         **  熱交換器か海水系に原因がある
           if [熱交換器から出てきた海水の温度が低い ]
             **  温度計 T16 を調べて下さい．１００度以下だと低いです
           then [熱交換器が汚れていて交換率が低い]
           endif
           if [熱交換器から出てきた海水の温度が高い ]
             **  温度計 T16 を調べて下さい．１８０度以上だと高いです
           then [海水系の冷却水が不足  ]
           **  海水系に原因がある


Fact Number 10 *** colc
[ 純水冷却系の圧力は正常  ]

                                                    command please  : t

State Number 9  *** colc
[ 純水系の冷却水は十分である  ] が推論されました

Fact Number 11 *** colc
[ 環境弁が閉いている ]

                                                    command please  : t

State Number 10 *** colc
[ 環境弁の閉め忘れ ] が推論されました

<<<  診断結果  >>>  colc

 ** 環境弁の閉め忘れ **

                                                    command please  : b
```

この命題に関するデータを示す。

命題を偽とみなし，”ｆ”を入力する。

推論状況を構造化プログラミング型で表現された知識上に示す。

真の命題は，黄色，偽の命題は，紫色，現在位置は，背景白色で示す。
２，８キーによって，スクロールダウン，スクロールアップする。

この命題が背景白色である。

ＥＳＣキーによって，通常モードに戻る。

診断結果が出たが入力ミスをしたので後退する。

```
Fact Number 11 *** colc
[ 循環弁が閉いている ]

                                                command please  : f

Fact Number 12 *** colc
[ 熱交換器から出てきた純水の温度が高い  ]

                                                command please  : t

State Number 11 *** colc
[ 熱交換器が十分働いていない ] が推論されました

Fact Number 13 *** colc
[ 熱交換器から出てきた海水の温度が低い  ]

                                                command please  : t

State Number 12 *** colc
[ 熱交換器が汚れていて交換率が低い ] が推論されました

<<<  診断結果  >>>  colc

 ** 熱交換器が汚れていて交換率が低い **

                                                command please  : c

Fact Number 14 *** colc
[ 熱交換器から出てきた海水の温度が高い  ]

                                                command please  : t

State Number 13 *** colc
[ 海水系の冷却水が不足  ] が推論されました

Fact Number 15 *** colc
[ 海水制御弁が閉じている  ]

                                                command please  : t

State Number 14 *** colc
[ 海水制御弁の故障  ] が推論されました

<<<  診断結果  >>>  colc

 ** 海水制御弁の故障  **

                                                command please  : q

                colc **  診断終了  **

8>


8>vt250

                vt250 ** 診断開始 **

**  ＶＴ２５０Ｆ  故障診断

**  始動時，走行中の速度，回転不調  の３種類の診断を行う

Fact Number 0 *** vt250
[ 始動時または走行中に異常がある ]

                                                command please  : t

Fact Number 1 *** vt250
[ 始動不能または始動困難である ]

                                                command please  : g


**  ＶＴ２５０Ｆ  故障診断

**  始動時，走行中の速度，回転不調  の３種類の診断を行う

if [始動時または走行中に異常がある]

    **  始動不能または始動困難    （ルーチン名  ＶＴ１）

    if [始動不能または始動困難である]
    then [燃料系または電装系の異常の可能性が強い]
        do vt1
    endif

    **  速度が出ない，力が出ない    （ルーチン名  ＶＴ２）

    if [速度が出ないまたは力が出ない]
    then [すべての系統に異常の可能性がある]
        do vt2
    endif
```

診断結果が表示される。

推論を続行して，更に探索を進める。

診断結果が表示される。

推論を終了する。

自動二輪車の診断を開始する。

推論状況を構造化プログラミング型で表現された知識上に示す。

この命題が背景白色である。

```
    **  回転不調      (ルーチン名  VT3, VT4)

    if [回転がおかしい]
     if [低速およびアイドリング時に回転不調]
     then [プラグやキャブレタに問題がある]
          do vt3
     endif
     if [高速時に回転不調]
     then [エンジン系の異常の可能性が強い]
          do vt4
     endif
    endif

endif
```

ＥＳＣキーによって，通常モードに戻る。

```
Fact Number 1 *** vt250
[ 始動不能または始動困難である ]
                                                command please  : t

State Number 0  *** vt250
[ 燃料系または電装系の異常の可能性が強い ] が推論されました

<<<  診断結果  >>>  vt250

 ** 燃料系または電装系の異常の可能性が強い **
                                                command please  : m
```

このルーチン(VT250)の診断結果が表示される。
更に詳細な診断ルーチン(VT1)に入る。

```
              vt1 ** 診断開始 **

**  ＶＴ２５０Ｆ  故障診断

**  始動不能または始動困難

Fact Number 0 *** vt1
[ スタータモータが回らない ]
                                                command please  : f

Fact Number 1 *** vt1
[ クランクが回らない ]
                                                command please  : f

Fact Number 2 *** vt1
[ キャブレータまでガソリンが十分行っていない ]
                                                command please  : t

State Number 2  *** vt1
[ フューエルチューブまたはフィルタのつまり，ストレーナの過度な汚れ，フューエルタ
ンク内にガソリンが入っていない，フロートバルブのつまり，フューエルタンクキャップ
の孔のつまり，  ] が推論されました

<<<  診断結果  >>>  vt1

 ** フューエルチューブまたはフィルタのつまり，ストレーナの過度な汚れ，フューエル
タンク内にガソリンが入っていない，フロートバルブのつまり，フューエルタンクキャッ
プの孔のつまり，  **
                                                command please  : q

              vt1 **  診断終了  **

Fact Number 2 *** vt250
[ 速度が出ないまたは力が出ない ]
                                                command please  : q

              vt250 **  診断終了  **

B>
```

ＶＴ１による診断結果が表示される。
ＶＴ１を終了する。

ＶＴ２５０上で次の探索を自動的に始める。
ＶＴ２５０を終了する。

7章　今後の展望

今後の展望として，構造化プログラミング型知識表現を更に充実させるために次のようなことが考えられる。

1）ＩＦ－ＴＨＥＮ－ＥＬＳＥ－ＥＮＤＩＦ以外にもＷＨＩＬＥ文のような制御構造を検討する。
2）モジュール化された診断サブルーチンを１つのファイルとしてではなく関数として取り扱えるようにし，１つのファイルに複数の関数を用意し，リンクできるようにする。
3）コメント文にも一定の書式を設定し，データベースシステムにおけるフィールドのような役割を持たせる。つまり，事実の確認方法，推論の現在位置，プラントの復旧手段，ルールの正当性といったメタ知識をプログラムのコメント文として構造化プログラミング型知識表現に取り込み，必要な時に必要なデータを検索できるようにする。
4）ヒューマンインタフェース的に向上させるためにプログラムの書式に柔軟性を持たせる。

8章　結言

本論文では，構造化プログラミング型知識表現によってエキスパートシステムを作成することを提案した。また，構造化プログラミング型知識表現に基づいて故障診断用エキスパートシステムを構築するツールを作成し，実際に船舶エンジン冷却系と自動二輪車に対する故障診断に適用してその有用性を確かめた。

最後に本研究に当たって温かい御指導と，適切かつ有用な助言をいただいた井上紘一教授，熊本博光助手と研究室諸氏に心から感謝の意を表します。

参考文献

[1] F.H.Roth,D.A.Waterman,D.B.Lenat ed: "Building Expert System" Addison-Wesley (1983)
（ＡＩＵＥＯ訳 "エキスパートシステム" 産業図書 (1985)）

[2] A.Barr,E.A.Feigenbaum ed:
"The Handbook of Artificial Intelligence Vol.1-3"
Pitman Books Ltd. (1981)
（田中，淵 監訳：人工知能ハンドブック，全３巻，
共立出版(1984)）

[3] 白井，辻井："人工知能"，岩波書店，情報科学－２２ (1983)

[4] Allen Newell,Herbert A.Simon: "Human Problem Solving"
Prentice Hall (1972)

[5] M.Minsky: "A Framework for Representing Knowledge"
In the Psychology of Computer Vision (Ed.P.H.Winston),
Mcgraw-Hill (1971)

[6] 小林："知識の表現と利用"，計測と制御，
Vol.24, No.2 PP.155-164 No.3 PP.242-250 (1985)

[7] 小林："エキスパートシステム"，計測と制御，Vol.24,
No.8 PP.730-738, No.9 PP.833-840 (1985)

[8] 熊本，鈴木，井上，池西："関係データベース操作言語によるプラント異常診断用エキスパートシステムの作成"，
計測自動制御学会論文集，Vol.21, NO.8 , PP.842-848 (1985)

[9] 奥村，熊本，井上："データベース検索可能なパソコン上のプラント異常診断用エキスパートシステム"，
計測自動制御学会第１回ヒューマン・インタフェース・シンポジウム論文集，PP.317-322 (1985)

[10] 鈴木："パーソナルコンピュータＤＢＭＳによるＡＩプロダクションシステムの作成と異常診断"，京都大学工学部精密工学科修士論文 (1984)

[11] "ホンダＶＴ２５０Ｆサービスマニュアル"，
本田技研工業株式会社 (1982)

[12] 竹内："Ｓｍａｌｌｔａｌｋ"，ｂｉｔ，Vol.15, No.12 PP.26-34
Vol.15, No.13 PP.54-63 (1983)，Vol.16, No.1 PP.34-44 (1984)

[13] B.W.Kernichan,D.M.Ritchie （石田訳）：
"プログラミング言語 C"，共立出版 (1981)

[14] Ｃ言語研究会："Ｃ サンプル＆ツール集"，技術評論社 (1984)

[15] "ＰＥＮＧＵＩＮ Ｃ86 Ｃ ＣＯＭＰＩＬＥＲ"，
Penguin Software Inc. (1985)

[16] 土："Ｅｄｉｔｏｒ Ｅの楽しい使い方" (1986)

[17] "日本語ｄＢＡＳＥⅡ User's Manual "，Ashton-Tate (1981)

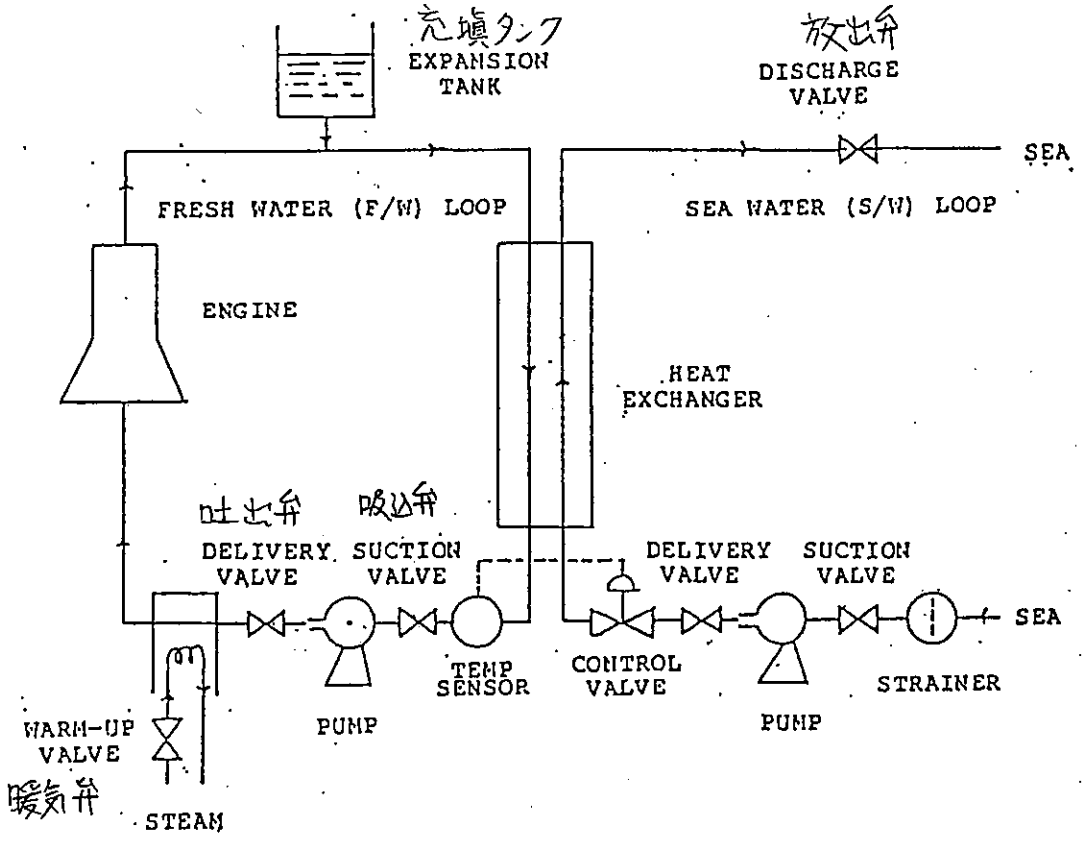
[18] 植村："データベースシステムの基礎"，オーム社 (1979)

[19] "ＭＳ－ＤＯＳ 2.0 ユーザーズマニュアル"，
日本電気株式会社 (1983)

[20] "標準ＭＳ－ＤＯＳハンドブック"，アスキー出版局 (1984)

[21] "Optimizing C86 User's Manual"，
Computer Innovations Inc. (1984)

付録．1　船舶エンジン冷却系ダイアグラム



付録．２　構造化プログラムで表現された自動二輪車の診断知識

```
** ＶＴ２５０Ｆ　故障診断

** 始動時，走行中の速度，回転不調　の３種類の診断を行う

if [始動時または走行中に異常がある]
    ** 始動不能または始動困難　（ルーチン名　ＶＴ１）
    if [始動不能または始動困難である]
    then [燃料系または電装系の異常の可能性が強い]
        do VT1
    endif
    ** 速度が出ない．力が出ない　（ルーチン名　ＶＴ２）
    if [速度が出ないまたは力が出ない]
    then [すべての系統に異常の可能性がある]
        do VT2
    endif
    ** 回転不調　（ルーチン名　ＶＴ３，ＶＴ４）
    if [回転がおかしい]
        if [低速およびアイドリング時に回転不調]
        then [プラグやキャブレタに問題がある]
            do VT3
        endif
        if [高速時に回転不調]
        then [エンジン系の異常の可能性が強い]
            do VT4
        endif
    endif
endif
```

** ＶＴ２５０Ｆ　故障診断

** 始動不能または始動困難

```
if [スータモータが回らない]
then [バッテリ電圧が不十分．スイッチが導通しない．ブラシの摩耗，ヒューズが切れている]
else
 if [クランクが回らない]
   ** クランクの回る音がします．プラグを抜いて確認することができます
 then [スタータクラッチまたはスタータ部のギアの破損]
 else
  if [キャブレータまでガソリンが十分行っていない]
    ** ドレンボルトをゆるめて，キャブレタまでガソリンが十分行っているか調べて下さい
  then [フューエルチューブまたはフィルタのつまり，ストレーナの過度な汚れ．フューエルタンク内にガソリンが入っていない．フロートバルブのつまり．フューエルタンクキャップの孔のつまり．]
  else
   if [プラグを外して２次コードキャップにつけて，スタータモータを回しても火が弱いか飛ばない]
   then [プラグ，パルスジェネレータ，スパークユニット不良． ハイテンションコードの断線または短絡．イグニションコイルの断線または短絡．メインスイッチ不良]
   else
    if [キルスイッチをＯＦＦにしてスタータモータを回したとき圧縮圧力が弱い]
    then [タッペットの隙間がない．バルブとシートの当りが悪い．シリンダとピストンリングの摩耗が大きい．シリンダガスケット部のガス漏れ．バルブの焼き付き．バルブタイミングが狂っている]
    else
     if [始動時に爆発の兆候があるが始動しない]
       ** 指導要領に従って始動してみて下さい
     then [バイスタータを使用し過ぎている．キャブレタのパイロットスクリュが閉じ過ぎている．マニホールドから空気を吸っている．点火時期が大きく狂っている．（スパークユニット不良）]
     else
      if [プラグが濡れている]
        ** プラグをもう一度外してみて下さい
      then [キャブレタがオバーフローしている．エアクリーナの汚れ．バイスタータを使用し過ぎ，ガソリンを吸い込みすぎた．スロットルバルブの開き過ぎ]
      endif
     endif
    endif
   endif
  endif
 endif
endif
```

```
**  ＶＴ２５０Ｆ　故障診断

**  速度が出ない．力が出ない

  if［前後輪を上げ，手で車輪が軽く回らない］
  then［ブレーキが効いたままである．ホイールベアリングが傷んでいる．ベアリンググ
       リースが切れている．ドライブチェインが張り過ぎている］
  else
   if［タイヤの空気が抜けている］
     **  タイヤを押してみてください．タイヤゲージで調べて下さい
   then［タイヤがパンクしている．タイヤのバルブが悪い］
   else
    if［チェンジをローからセカンドに入れながら急加速したときに，クラッチをはして
        もしばらく回転が下がらない］
    then［油圧クラッチの液漏れ，クラッチディスクの摩耗，クラッチディスクプレート
         の歪み，クラッチスプリングがへたっている］
    else
     if ［エンジンをふかしても回転が十分上がらない］
       **  エンジンを軽くふかして下さい
     then［ バイスタータが使用されたままになっている．エアクリーナがつまっている
          ．ガソリンの吐出量不足．フューエルタンクキャップの孔のつまり．マフラ
          のつまり］
     else
      if［点火時期が正しくない］
        **  点火時期を調べて下さい．タイミングライトを用いて下さい
      then［スパークユニット不良，パルスジェネレータ不良］
      else
       if［タペットの隙間が正しくない］
       then ［タペットの調整が悪い．バルブシートが減っている（突き上げ）］
       else
        if［スタータを回しても圧縮がない］
          **  コンプレッションゲージを用いて下さい
        then[バルブシートの当りが悪い．シリンダピストンリングの摩耗．ガスケット
             部からガスが漏れる．バルブタイミングが狂っている］
        else
         if［エアクリーナ，キャブレタがつまっている］
         then［掃除が悪い］
         else
          if［プラグが汚れていたり，変色している］
            **  プラグを外してみて下さい
          then［掃除が悪い．プラグの熱価が合わない］
          else
           if［オイルの量が正しくない］
           then［オイルが多過ぎて力が出ない．少な過ぎる．汚れ過ぎている］
           else
            if［潤滑状態が良くない］
              **  シリンダヘッドカバーを外してみて下さい
            then［オイル通路のつまり．オイルオリフィスのつまり］
            else
             if［エンジンがオーバーヒートする］
             then［エンジンオイルの不足．オイルポンプ不良．オイル通路のつまり．
                  燃焼室にカーボンが堆積している．ガソリン質が悪い．クラッチが
                  すべっている．混合気が薄い］
             else
              if［加速または高速連続運てん中にノッキングする］
              then［ピストンとシリンダの摩耗．燃焼室にカーボンが堆積している．ガ
                   ソリン質が悪い．混合気が薄い．点火時期が早すぎる（スパークユ
                   ニット不良）］
              endif
             endif
            endif
           endif
          endif
         endif
        endif
       endif
      endif
     endif
    endif
   endif
  endif
endif
```

```
** ＶＴ２５０Ｆ　故障診断

** 回転不調　（主として低速およびアイドリング）


if [点火時期，タペットの隙間が正しくない]
then [タペットが正規に調整されていない．スパークユニット不良]
else
 if [キャブレタのパイロットスクリュの調整が不良]
 then [ガスが薄い．ガスが濃い]
 else
  if [二次空気を吸っている]
  then [インシュレータのバンドの締め付け不足．インシュレータの破損．インシュレータの取り付け不良]
  else
   if [スタータモータを回したとき火花が悪い，または時々出ない]
     ** プラグを外し，二次コードキャップに付けてスタータモータを回して火花の状況をみる
    then [プラグの不良．スパークユニット不良．パルスジェネレータ不良．イグニッションコイル不良．ハイテンションコードの断線または短絡．メインスイッチが悪い]
   endif
  endif
 endif
endif


** ＶＴ２５０Ｆ　故障診断

** 回転不調　（高速）


if [点火時期，タペットの隙間が正しくない]
then [タペットが正規に調整されていない．スパークユニット不良．パルスジェネレータ不良]
else
 if [フューエルチューブをガソリンタンクのところで外すと，ガソリンの出方が悪い]
 then [タンクのガソリンの量が少ない．フューエルチューブがつまっている．タンクキャップの孔がつまっている．フューエルフィルタまたはコックがつまっている．フューエルストレーナの過度な汚れ]
 else
  if [ジェットがつまっている]
    ** キャブレタを外してジェットのつまりを見る
  then [ジェットの掃除不足]
  else
   if [バルブタイミングが狂っている]
   then [カムスプロケットの合わせ位置があっていない]
   else
    if [バルブスプリングに折れまたはへたりがある]
    then [スプリングが不良である]
    endif
   endif
  endif
 endif
endif
```

付録．３　Ｃ言語に変換された船舶エンジン冷却系診断プログラム

```
#include <fdiag.h>
char *exname = "colc" ;

diag()
{

   int cm;

   while ((cm = ffact(0)) != 2 ){
    if (cm) {
     while ((cm = ffact(1)) != 2 ){
      if (cm) {
       fstat(0) ;
       while ((cm = ffact(2)) != 2 ){
        if (cm) {
         fstat(1) ;
         while ((cm = ffact(3)) != 2 ){
          if (cm) {
           fstat(2) ;
           term();
           if (cof) break;
          }
          else {
           fstat(3);
           while ((cm = ffact(4)) != 2 ){
            if (cm) {
             fstat(4) ;
             term();
             if (baf) break;
            }
            while ((cm = ffact(5)) != 2 ){
             if (cm) {
              fstat(5) ;
              term();
              if (baf) break;
             }
             while ((cm = ffact(6)) != 2 ){
              if (cm) {
               fstat(6) ;
               term();
               if (baf) break;
              }
              while ((cm = ffact(7)) != 2 ){
               if (cm) {
                fstat(7) ;
                term();
                if (baf) break;
               }
               while ((cm = ffact(8)) != 2 ){
                if (cm) {
                 while ((cm = ffact(9)) != 2 ){
                  if (cm) {
                   fstat(8) ;
                   term();
                   if (cof) break;
                  }
                  if (cof) break;
                 }
                 if (cof) break;
                }
                if (cof) break;
               }
               if (cof) break;
              }
              if (cof) break;
             }
             if (cof) break;
            }
            if (cof) break;
           }
           if (cof) break;
          }
          if (cof) break;
         }
        }
       }
      }
     }
    }
```

```
while ((cm = ffact(10)) != 2 ){
 if (cm) {
  fstat(9) ;
  while ((cm = ffact(11)) != 2 ){
   if (cm) {
    fstat(10) ;
    term();
    if (cof) break;
   }
   else {
    while ((cm = ffact(12)) != 2 ){
     if (cm) {
      fstat(11) ;
      while ((cm = ffact(13)) != 2 ){
       if (cm) {
        fstat(12) ;
        term();
        if (baf) break;
       }
       while ((cm = ffact(14)) != 2 ){
        if (cm) {
         fstat(13) ;
         while ((cm = ffact(15)) != 2 ){
          if (cm) {
           fstat(14) ;
           term();
           if (baf) break;
          }
          while ((cm = ffact(16)) != 2 ){
           if (cm) {
            fstat(15) ;
            term();
            if (cof) break;
           }
           else {
            fstat(16);
            while ((cm = ffact(17)) != 2 ){
             if (cm) {
              fstat(17) ;
              term();
              if (baf) break;
             }
             while ((cm = ffact(18)) != 2 ){
              if (cm) {
               fstat(18) ;
               term();
               if (baf) break;
              }
              while ((cm = ffact(19)) != 2 ){
               if (cm) {
                fstat(19) ;
                term();
                if (baf) break;
               }
               while ((cm = ffact(20)) != 2 ){
                if (cm) {
                 fstat(20) ;
                 term();
                 if (baf) break;
                }
                while ((cm = ffact(21)) != 2 ){
                 if (cm) {
                  while ((cm = ffact(22)) != 2 ){
                   if (cm) {
                    fstat(21) ;
                    term();
```

```
                    if (cof) break;
                   }
                   if (cof) break;
                  }
                  if (cof) break;
                 }
                 if (cof) break;
                }
                if (cof) break;
               }
               if (cof) break;
              }
              if (cof) break;
             }
             if (cof) break;
            }
            if (cof) break;
           }
           if (cof) break;
          }
          if (cof) break;
         }
         if (cof) break;
        }
        if (cof) break;
       }
       if (cof) break;
      }
      if (cof) break;
     }
     if (cof) break;
    }
    else {
     fstat(22);
     term();
     if (cof) break;
    }
    if (cof) break;
   }
   if (cof) break;
  }
  if (cof) break;
 }

}
```

```
/*    Construct Diagnosis System                version 2.0                    */

#include <fdiag.h>

int _MXFMEM = 0x900;

struct rowg {long off;int col;} row[512];
struct statg *stat ;
struct factg *fact ;

int maxline;
int rowc[200];
int pfnum,psnum ;
int penum = 0;
int cof = 0;
int mof = 0;
int baf = 0;
int fatnum;

main ()
{

    if (!(stat = malloc(60*sizeof(struct statg)))){
       printf("¥n¥terror¥n");
       return;
    }
    if (!(fact = malloc(60*sizeof(struct factg)))){
       printf("¥n¥terror¥n");
       return;
    }

    datf(); /*  Fact & State data file are converted to Struct Array  */
    mess(); /*  Messages    */
    diag(); /*  Diagnosis  routine     */

    printf("%c[34m¥n¥n¥t¥t%s **  診断終了  **  ¥n",ESC,exname);
    printf("%c[m",ESC);

}


datf()
{

     FILE *fp;
     char pat[256];
     int i ;
     char fat[15],sat[15];

     strcat(strcpy(fat,exname),".fat");
     strcat(strcpy(sat,exname),".sat");

     for (i = 0;i < 60; fact[i++].truth = 2) ;

     fp = fopen(fat,"r");
     i = 0;
     while(fgets(pat,256,fp)){
        strcpy(fact[i].fprop,delen(pat));
        fgets(pat,256,fp);
        fact[i++].truth = atoi(pat);
     }
     fclose(fp);
     fatnum = i;

     fp = fopen(sat,"r");
     i = 0;
     while(fgets(pat,256,fp)){
        strcpy(stat[i++].sprop,delen(pat));
     }
     fclose(fp);

}
```

```
mess()
{
    FILE *fp;
    char tap[256],dat[15];
    int j = 0;

    printf("%c[34m¥n¥t¥t%s ** 診断開始 **  ¥n¥n",ESC,exname);
    printf("%c[m",ESC);
    strcat(strcpy(dat,exname),".dat");
    fp = fopen(dat,"r");
    while (fgets(tap,256,fp)){
       j++;
       if (j > 5) break;
       if (search(tap,"**") || blank(tap)) printf("%c[33m%s",ESC,tap);
    }
    printf("%c[m",ESC);
    fclose(fp);
}

/*  Following functions are used in diag().  */

ffact(i)
int i;
{
     char ans;
     char pat[256];

     pfnum = i;
     cof = mof = 0;

     strcpy(pat,"[ ");
     strcat(pat,fact[i].fprop);
     strcat(pat," ]");
     for (;;) {
         if (baf){
              fact[i].truth = 2; baf = 0 ;
         }
         switch(fact[i].truth){
         case 0:
              printf("%c[35m¥nFact Number %d¥n",ESC,i);
              printf("%c[m",ESC);
              printf("%s is fault.¥n",pat);
              if ( i == penum ) {
                 fact[i].truth = 2;
                 printf("%c[31m¥n !!! 診断できません !!!¥n",ESC);
                 printf("%c[m",ESC);
              }
              penum = i;
              return(0);
         case 1:
              printf("%c[33m¥nFact Number %d¥n",ESC,i);
              printf("%c[m",ESC);
              printf("%s is true.¥n",pat);
              return(1);
         case 2:
              printf("%c[36m¥nFact Number %d *** %s¥n",ESC,i,exname);
              printf("%c[m",ESC);
              printf("%s¥n",pat);
              printf("%c[32m¥n¥t¥t¥t¥t¥t¥t   command please  : ",ESC);
              printf("%c[m",ESC);
              for (;;){
                  ans = inkey();
                  putchar(ans);
                  switch (tolower(ans)){
                  case 'b':
                        baf = 1;
                        return(2);
                  case 'c' :
                        cof = 1;
                        return(0);
                  case 'd':
                        system("dbase");
                        break;
                  case 'e':
                        system("e *.fat");
                        break;
                  case 'f' :
                        fact[i].truth = 0;
                        return(0);
```

```
                    case 'g' :
                          disp(i);
                          makerow();
                          list();
                          break;
                    case 'h' :
                          fhelp();
                          break;
                    case 'i':
                          dbas(fact[i].fprop);
                          break;
                    case 'q' :
                          printf("%c[34m¥n¥n¥t¥t%s **  診断終了  **  ¥n",ESC,exname);
                          printf("%c[m",ESC);
                          exit();
                    case 's':
                          savef();
                          break;
                    case 't':
                          fact[i].truth = 1;
                          return(i);
                    default  :
                          printf("¥b",ans);
                          ans = '@';
                    }
                    if (ans != '@') break;
               }
          }
     }
}


fstat(i)
int i;
{
    char pat[256];

    psnum = i;
    printf("%c[33m¥nState Number %d¥t*** %s ¥n",ESC,i,exname);
    printf("%c[m",ESC);
    printf("[ %s ] が推論されました¥n",stat[i].sprop);
}

delen(s)
char s[ ];
{
    *index(s,'¥n') = '¥0';
    return(s);
}

term()
{
    char ans;
    char pat[256];

    for (;;){
        printf("%c[35m¥n<<<  診断結果  >>>  %s¥n¥n",ESC,exname);
        printf("%c[4lm ** %s ** ¥n",ESC,stat[psnum].sprop);
        printf("%c[m",ESC);
        printf("%c[32m¥n¥t¥t¥t¥t¥t¥t   command please  : ",ESC);
        printf("%c[m",ESC);
        for (;;){
            ans = inkey();
            putchar(ans);
            switch (tolower(ans)){
            case 'b':
                  fact[pfnum].truth = 2;
                  baf = 1;
                  return;
            case 'c':
                  cof = 1;
                  return;
```

```
                case 'd':
                      system("dbase");
                      break;
                case 'e':
                      system("e *.fat");
                      break;
                case 'g' :
                      displ();
                      makerow();
                      list();
                      break;
                case 'h' :
                      shelp();
                      break;
                case 'm':
                      mOf = 1;
                      return;
                case 'q' :
                      printf("%c[34m¥n¥n¥t¥t%s **  診断終了  **  ¥n",ESC,exname);
                      printf("%c[m",ESC);
                      exit();
                case 's':
                      savef();
                      break;
                default  :
                      printf("¥b",ans);
                      ans = '@';
                )
                if (ans != '@') break;
            }
        }
    }

displ()
{
    FILE *fp;
    char tap[256],doc[15];
    char pat[4],pac[5];
    int i,j;

    strcat(strcpy(doc,exname),".doc");
    fp = fopen(doc,"r");
    for (i = 0;i < 200; rowc[i++] = 37 ) ;

    for (i = 0;i < 60;++i){
        switch (fact[i].truth){
        case 0:
            itoa(i,pat);
            strcat(pat,")");
            strcpy(pac,"(");
            strcat(pac,pat);
            j = 0;
            rewind(fp);
            while (fgets(tap,256,fp)){
                ++j;
                if (search(tap,pac)){
                    rowc[j] = 35;
                    break;
                }
            }
            break;
        case 1:
            itoa(i,pat);
            strcat(pat,")");
            strcpy(pac,"(");
            strcat(pac,pat);
            j = 0;
```

```
            rewind(fp);
            while (fgets(tap,256,fp)){
                ++j;
                if (search(tap,pac)){
                    rowc[j] = 33;
                    break;
                }
            }
            break;
        }
    }
    rewind(fp);

    itoa(psnum,pat);
    strcat(pat,"]");
    strcpy(pac,"[");
    strcat(pac,pat);

    j = 0;
    while (fgets(tap,256,fp)){
        ++j;
        if (search(tap,pac))
            rowc[j] = 41;
        else if (search(tap,"**"))
            rowc[j] = 32;
    }
    fclose(fp);
}

disp(n)
int n;
{
    FILE *fp;
    char tap[256],doc[15];
    char pat[4],pac[5];
    int i,j;

    strcat(strcpy(doc,exname),".doc");
    fp = fopen(doc,"r");
    for (i = 0;i < 200; rowc[i++] = 37 ) ;

    itoa(n,pat);
    strcat(pat,")");
    strcpy(pac,"(");
    strcat(pac,pat);

    j = 0;
    while (fgets(tap,256,fp)){
        ++j;
        if (search(tap,pac))
            rowc[j] = 47;
        else if (search(tap,"**"))
            rowc[j] = 32;
    }
    rewind(fp);

    for (i = 0;i < 60;++i){
        switch (fact[i].truth){
        case 0:
            itoa(i,pat);
            strcat(pat,")");
            strcpy(pac,"(");
            strcat(pac,pat);
            j = 0;
            rewind(fp);
            while (fgets(tap,256,fp)){
```

```
                    ++j;
                    if (search(tap,pac)){
                        rowc[j] = 35;
                        break;
                    }
                }
                break;
            case 1:
                itoa(1,pat);
                strcat(pat,")");
                strcpy(pac,"(");
                strcat(pac,pat);
                j = 0;
                rewind(fp);
                while (fgets(tap,256,fp)){
                    ++j;
                    if (search(tap,pac)){
                        rowc[j] = 33;
                        break;
                    }
                }
                break;
            }
        }
        fclose(fp);
}

search(prt,p)
char *prt;
char *p;
{
     for(;;){
       if (!(prt = index(prt,*p))){
          return(0);
       }
       if (!(strncmp(prt++,p,strlen(p)))){
           return(--prt);
       }
     }
}

makerow()
{
    FILE    *fp;
    long    offset,k;
    int     line=1,n;
    char    buf[256];

    detab();
    fp = fopen("c:dm3.txt","r");
    maxline=1;
    for (;;){
        offset=fseek(fp,0L,1);
        if (!fgets(buf,256,fp)) break;
        n=strlen(buf);
        for (k=0 ; k<n ; k+=80){
            row[maxline].off=offset+k;
            row[maxline].col=rowc[line];
            maxline++;
        }
        line++;
    }
    fclose(fp);
}

list()
{
    int line=1,y=0;
    FILE *fp;
    char cc;

    fp = fopen("c:dm3.txt","r");
    cls();
    do {print(fp,line++);++y;} while (y<YMAX && y<maxline);
    line=1;
    locate(0,YMAX);
```

```
    while ((cc=inkey())!=ESC){
        if (cc=='8' && line>1){
            scdown();
            print(fp,--line);
        }
        else if (cc=='2' && line<maxline-YMAX){
            scup();
            locate(0,YMAX-1);
            print(fp,line+YMAX);
            ++line;
        }
        locate(0,YMAX);
    }
    fclose(fp);
    printf("%c[m",ESC);
}

print(fp,line)
FILE *fp;
int line;
{
    char buf[81];

    fseek(fp,row[line].off,0);
    fgets(buf,81,fp);
    printf("%c[%dm%s",ESC,row[line].col,buf);
}

detab()
{
     FILE *fp,*fpw;
     char c;
     char dat[15];
     int colum = 0;

     strcat(strcpy(dat,exname),".dat");
     fp = fopen(dat,"r");
     fpw = fopen("c:dm3.txt","w");
     while ((c = getc(fp)) != EOF){
        if (c != '¥t'){
            putc(c,fpw);
            colum++;
            if ( c == '¥n')
               colum = 0;
        }
        else
            do putc(' ',fpw) ; while (++colum % 8);
     }
     fclose(fpw);
     fclose(fp);
}

savef()
{
    FILE *fpw;
    int i;
    char fat[15];

    strcat(strcpy(fat,exname),".fat");
    fpw = fopen(fat,"w");
    for (i = 0; i < fatnum ;++i)
        fprintf(fpw,"%s¥n%d¥n",fact[i].fprop,fact[i].truth);
    fclose(fpw);

}
```

```
blank(tap)
char tap[256];
{
    int i = 0;
    if (tap[i++] == '¥n') return(1);
    while (tap[i] == ' ' || tap[i] == '¥t'){
        if (tap[++i] == '¥n') return(1);
    }
    return(0);

}

#include <fdiag.h>


dbas(pat)
char pat[256];
{
    FILE *fp,*fpw ;
    char hpat[100],idx[16],idxn[16];
    char c;

    strcat(strcpy(idx,exname),"x");
    strcat(strcpy(idxn,exname),"x.ndx");
    if (!(fp = fopen(idxn,"r"))){
        printf("%c[5;35m ¥t¥tファイルが準備されていません  ¥n",ESC);
        return;
    }
    fclose(fp);
    strncpy(hpat,pat,100);

    fpw = fopen("b:dida.prg","w");
    fprintf(fpw,"use %s index %s¥n",exname,idx);
    fprintf(fpw,"find %s¥n",hpat);
    fp = fopen("a:sample.txt","r");
    while ((c = getc(fp)) != EOF)
        putc(c,fpw);
    fclose(fp);
    fclose(fpw);

    system("dbase b:dida");
    printf("%c[3A",ESC);
    printf("%c[K",ESC);

}


#include <fdiag.h>

fhelp()
{
    printf("¥n¥n");
    printf("%c[35m¥tb ‥ 前の質問に戻ります．戻らなければ再入力して下さい¥n",ESC);
    printf("%c[35m¥tc ‥ 次のブロックの推論を行います．ブロックが無ければ終了します¥n",ESC);
    printf("%c[31m¥td ‥ ｄＢＡＳＥⅡのモードに入ります¥n",ESC);
    printf("%c[31m¥te ‥ エディトモードに入り，クィックテープル等の編集ができます¥n",ESC);
    printf("%c[33m¥tf ‥ 命題を偽とみなします¥n",ESC);
    printf("%c[35m¥tg ‥ 推論の経過および方向を表示します．上下　２－８，ＥＳＣで終了します¥n",ESC);
    printf("%c[35m¥th ‥ コマンドの説明を行います¥n",ESC);
    printf("%c[35m¥ti ‥ 確認方法などのデータを提供します¥n",ESC);
    printf("%c[35m¥tq ‥ 現在推論中のルーチンを終了します¥n",ESC);
    printf("%c[35m¥ts ‥ 現在までの応答をセーブします¥n",ESC);
    printf("%c[33m¥tt ‥ 命題を真とみなします¥n",ESC);

}

shelp()
{
    printf("¥n¥n");
    printf("%c[37m¥tb ‥ 前の質問に戻ります．戻らなければ再入力して下さい¥n",ESC);
    printf("%c[37m¥tc ‥ 推論を続行します¥n",ESC);
    printf("%c[31m¥td ‥ ｄＢＡＳＥⅡのモードに入ります¥n",ESC);
    printf("%c[31m¥te ‥ エディトモードに入り，クィックテープル等の編集ができます¥n",ESC);
    printf("%c[37m¥tg ‥ 推論の経過および方向を表示します．上下　２－８，ＥＳＣで終了します¥n",ESC);
    printf("%c[37m¥th ‥ コマンドの説明を行います¥n",ESC);
    printf("%c[33m¥tm ‥ 以下に推論ルーチンがあればそのルーチンへ移ります¥n",ESC);
    printf("%c[37m¥tq ‥ 現在推論中のルーチンを終了します¥n",ESC);
    printf("%c[37m¥ts ‥ 現在までの応答をセーブします¥n",ESC);

}
```

付録．４　診断システム構築ツール・プログラムリスト

trans.bat

```
echo off
cls
echo off
:loop
    if exist %1.dat goto compile
    echo 5;41m   %1.dat が見つかりません.
    goto end
:compile
    ECHO 47m データファイルからＣ言語のプログラムを作成し，コンパイルします.
    a:cdiag %1
    cc %1 a:fdiag a:dbas a:help
    goto end
:end
ECHO m
echo off
```

cdiag.c

```
/*     diagnosis creation      from dat file to diagnosis program     */

# include <cdiag.h>

struct statg *stat ;
struct factg *fact ;
struct key *data;

int ftc,stc;

main (argc, argv)
int argc;
char *argv[ ];
{
     FILE *fp ;
     char org[11],dat[15],fat[15],sat[15],doc[15],ccc[15];

     strcat(strcpy(dat,argv[1]),".dat");

     if ( !(fp = fopen(dat,"r"))){
        printf("¥n¥terror0¥n");
        return;
     }
     if (!(stat = malloc(60*sizeof(struct statg)))){
        printf("¥n¥terror1¥n");
        return;
     }
     if (!(fact = malloc(60*sizeof(struct factg)))){
        printf("¥n¥terror2¥n");
        return;
     }
     if (!(data = malloc(100*sizeof(struct key)))){
        printf("¥n¥terror3¥n");
        return;
     }

     strcpy(org,argv[1]);
     strcat(strcpy(doc,argv[1]),".doc");
     strcat(strcpy(fat,argv[1]),".fat");
     strcat(strcpy(sat,argv[1]),".sat");
     strcat(strcpy(ccc,argv[1]),".c");

     printf("¥n%c[3lm 診断に必要なファイルを作成します ",ESC);

     /*  #exp9.c                                                  */

     grep(dat);          /* from dat file to c:dml.doc  */
     crep(doc,fat,sat);  /* from c:dml.doc to doc file  */
```

```
    printf("%c[13D",ESC);
    printf("%c[K",ESC);
    printf("%c[36m  %s, %s , %s が作成されました¥n",ESC,fat,sat,doc);
    printf("¥n%c[32m C言語による診断システム     %s を作成します",ESC,ccc);

    arep(doc);          /* from doc file to c:dm2.doc */
    mldp(ccc,org);      /* create diag() */

    printf("%c[%dD",ESC,strlen(ccc)+13);
    printf("%c[33m%s が作成されました¥n",ESC,ccc);

    printf("%c[m¥n¥n",ESC);
}



/*     diagnosis creation     a group of functions for CDIAG.c */

# include <cdiag.h>

grep(dat)
char *dat;

{
    FILE *fp,*fpw;
    char pat[256],buf[10],que[3];
    int i = 0;
    int j = 0;
    char c;

    fp = fopen(dat,"r");
    fpw = fopen("dm0.doc","w");
    que[0] = getc(fp);
    que[1] = getc(fp);
    while (que[0] != EOF){
       if (strncmp(que,"  ",2)){
           putc(que[0],fpw);
           que[0] = que[1];
           que[1] = getc(fp);
       }
       else {
           fputs("  ",fpw);
           que[0] = getc(fp);
           que[1] = getc(fp);
       }
    }
    fclose(fp);
    fclose(fpw);

    fp = fopen("dm0.doc","r");
    while ((c = getc(fp)) != EOF){
       if (iskanji(c)){
           c = getc(fp);
           continue;
       }
       if (c == '['){
          while ( (c = getc(fp)) != ']') {
             if (c == ' '){
                pat[i++] = c;
                while ((c = getc(fp)) == ' ' || c == '¥t');
             }
             else if ( c == '¥t')
                while ((c = getc(fp)) == ' ' || c == '¥t');
             else if (iskanji(c)){
                pat[i++] = c;
                pat[i++] = (c = getc(fp));
                continue ;
             }
             if (c == ']') break;
             pat[i++] = c;
          }
          pat[i] = '¥0';  i = 0;
          strcpy(data[j++].prop, pat);
       }
    }
    rewind(fp);
```

```
        fpw = fopen("dml.doc","w");
        putc(' ',fpw);
        while ((c = getc(fp)) != EOF){
          if (iskanji(c)) {
              putc(c,fpw);
              putc((c = getc(fp)),fpw);
          }
          else if (c == '[' ){
             itoa(++l,buf);
             putc('[',fpw);
             fputs(buf,fpw);
             while ((c = getc(fp)) != ']')
               if (iskanji(c))
                  c = getc(fp);
             putc(']',fpw);
          }
          else {
             putc(c,fpw);
          }
        }
        fclose(fp);
        fclose(fpw);

        fpw = fopen("dml0.doc","w");
        fp = fopen("dml.doc","r");
        while ((c = getc(fp)) != EOF) {
           if (iskanji(c)){
              putc(c,fpw);
              putc((c = getc(fp)),fpw);
           }
           else {
              putc(tolower(c),fpw);
              if (c == '¥n' || c == '¥t')
                  putc(' ',fpw);
           }
        }
        fclose(fp);
        fclose(fpw);
}

crep(doc,fat,sat)
char *doc,*fat,*sat;
{
        FILE *fp,*fpw ;
        int k = 1;      /* counter of 'indent' */
        int m = 0;      /* counter of 'data'   */
        int i;
        char tap[256];
        char *pat;

        ftc = 0;    /* counter of 'fact'   */
        stc = 0;    /* counter of 'state'  */

        fp = fopen("dml0.doc","r");
        fpw = fopen(doc,"w");

        while (fgets(tap,256,fp)){
            if (search(tap,"**"))
                fprintf(fpw,"  ** comment ¥n");
            else if (search(tap," if")) {
                indent2(++k ,fpw);
                fprintf(fpw," if (%d)¥n",regif(data[m++].prop));
            }
            else if (search(tap,"then")){
                indent2(k,fpw);
                fprintf(fpw," then [%d]¥n",regis(data[m++].prop));
            }
            else if (pat = (search(tap,"do"))){
                indent2(k+1,fpw);
                fprintf(fpw,"%s",pat);
            }
            else if (search(tap,"else")){
                indent2(k,fpw);
                fprintf(fpw," else");
                if (search(tap,"[")){
                    fprintf(fpw," [%d]",regis(data[m++].prop));
                }
                fprintf(fpw,"¥n");
            }
```

```
            else if (search(tap,"endif")){
                indent2(k,fpw);--k;
                fprintf(fpw," endif¥n");
            }
            else
                fprintf(fpw,"¥n");
        }
        fclose(fp);
        fclose(fpw);


        /*   construct fact data file  */

        fpw = fopen(fat,"w");
        for(i = 0; i < ftc ;i++){
            fprintf(fpw,"%s¥n",fact[i].fprop);
            fprintf(fpw,"%s¥n","2");
        }
        fclose(fpw);

        /*   construct state data file  */

        fpw = fopen(sat,"w");
        for(i = 0; i < stc ;i++){
            fprintf(fpw,"%s¥n",stat[i].sprop);
        }
        fclose(fpw);
}


arep(doc)
char *doc;
{
        FILE *fp,*fpw ;
        int i;
        char tap[256],taq[256];
        int dep[64];

        fp = fopen(doc,"r");
        fpw = fopen("dm2.doc","w");

        for (i = 0 ;i < 64 ;++i) dep[i] = 0 ;

        *taq = 0;
        while (fgets(tap,256,fp)){
            if (search(taq," endif")){
                if (search(tap," if")){
                    ++(dep[depth(tap)]);
                    strcpy(taq," *endif¥n");
                }
                else if (search(tap," endif")||search(tap," else")){
                        for (i = 0;i < dep[depth(taq)];++i)
                            fprintf(fpw,"%s¥n"," #endif");
                        dep[depth(taq)] = 0;
                }
            }
            if (search(tap,"**")||blank(tap))
                continue;
            fprintf(fpw,"%s",taq);
            strcpy(taq,tap);
        }
        for (i = 0;i < dep[0];++i) fprintf(fpw,"%s¥n"," #endif");
        dep[0] = 0;

        fprintf(fpw,"%s",taq);
        fclose(fp);
        fclose(fpw);

}
```

```
mldp(ccc,org)
char *ccc,*org;
{
     FILE *fp, *fpw;
     char tap[256],taq[256];
     int k = 1 ;

     fpw = fopen(ccc,"w");
     fp = fopen("b:dm2.doc","r");

     fprintf(fpw,"#include <fdiag.h>¥n");
     fprintf(fpw,"char *exname = ¥"%s¥" ;¥n¥n",org);
     fprintf(fpw,"diag()¥n{¥n¥n");
     fprintf(fpw,"    int cm;¥n¥n");

     *taq = 0;
     while (fgets(tap,256,fp)){
         if (search(tap," ** "))
             continue;
         if (search(tap," if ")){
             indent(k,fpw);
             fprintf(fpw,"while ((cm = ffact(%d)) != 2 ){¥n",atoif(tap));
             indent(++k,fpw);
             fprintf(fpw,"if (cm) {¥n"); ++k;
             strcpy(taq,tap);
             continue;
         }
         if (search(tap," then")){
             indent(k,fpw);
             fprintf(fpw,"fstat(%d) ;¥n",atois(tap));
             strcpy(taq,tap);
             continue;
         }
         if (search(tap," else")){
             if (search(taq,"then")){
                indent(k,fpw);
                fprintf(fpw,"term();¥n");
             }
             else if (search(taq,"do")){
                 indent(k,fpw);
                 fprintf(fpw,"term();¥n");
                 indent(k,fpw);
                 fprintf(fpw,"if (mof) system(¥"%s¥");¥n",essen(taq));
             }
             indent(k,fpw);
             fprintf(fpw,"if (cof) break;¥n");
             indent(--k,fpw);
             fprintf(fpw,"}¥n");
             indent(k,fpw);
             fprintf(fpw,"else {¥n"); ++k;
             if (search(tap,"[")){
                 indent(k,fpw);
                 fprintf(fpw,"fstat(%d);¥n",atois(tap));
             }
             strcpy(taq,tap);
             continue;
         }
         if (search(tap," endif")){
             if (search(taq,"then") || search(taq,"else")){
                 indent(k,fpw);
                 fprintf(fpw,"term();¥n");
             }
             else if (search(taq,"do")){
                 indent(k,fpw);
                 fprintf(fpw,"term();¥n");
                 indent(k,fpw);
                 fprintf(fpw,"if (mof) system(¥"%s¥");¥n",essen(taq));
             }
             indent(k,fpw);
             fprintf(fpw,"if (cof) break;¥n");
             indent(--k,fpw);
             fprintf(fpw,"}¥n");
             indent(k,fpw);
             fprintf(fpw,"if (cof) break;¥n");
             indent(--k,fpw);
             fprintf(fpw,"}¥n");
             strcpy(taq,tap);
             continue;
```

```
            }
            if (search(tap,"#endif")){
                if (search(taq,"then")||search(taq,"else")){
                    indent(k,fpw);
                    fprintf(fpw,"term();¥n");
                    indent(k,fpw);
                    fprintf(fpw,"if (baf) break;¥n");
                }
                else if (search(taq,"do")){
                    indent(k,fpw);
                    fprintf(fpw,"term();¥n");
                    indent(k,fpw);
                    fprintf(fpw,"if (mof) system(¥"%s¥");¥n",essen(taq));
                }
                indent(--k,fpw);
                fprintf(fpw,")¥n");
                strcpy(taq,tap);
                continue;
            }
            if (search(tap,"#endif")){
                if (search(taq,"then") || search(taq,"else")){
                    indent(k,fpw);
                    fprintf(fpw,"term();¥n");
                    indent(k,fpw);
                    fprintf(fpw,"if (baf) break;¥n");
                }
                else if (search(taq,"do")){
                    indent(k,fpw);
                    fprintf(fpw,"term();¥n");
                    indent(k,fpw);
                    fprintf(fpw,"if (mof) system(¥"%s¥");¥n",essen(taq));
                }
                indent(k,fpw);
                fprintf(fpw,"if (cof) break;¥n");
                indent(--k,fpw);
                fprintf(fpw,")¥n");
            }
            strcpy(taq,tap);
        }
        fputs("¥n)¥n",fpw);
        fclose(fp);
        fclose(fpw);
}

search(prt,p)
char *prt;
char *p;
{
        for(;;){
          if (!(prt = index(prt,*p))){
             return(0);
          }
          if (!(strncmp(prt++,p,strlen(p)))){
              return(--prt);
          }
        }
}

indent(k,fpw)
int k;
FILE *fpw;
{
    fprintf(fpw,"  ");
    while ( k-- ) fprintf(fpw," ");

indent2(k,fpw)
int k;
FILE *fpw;
{
    fprintf(fpw,"  ");
    while ( k-- ) fprintf(fpw,"  ");

}
```

```
regif(pat)
char *pat;
{
    strcpy(fact[ftc].fprop,pat);
    return(ftc++);
}

regis(pat)
char *pat;
{
    strcpy(stat[stc].sprop,pat);
    return(stc++);
}


atoif(tap)
char *tap;
{
  tap = index(tap,'(');
  return(atoi(++tap));
}

atois(tap)
char *tap;
{
  tap = index(tap,'[');
  return(atoi(++tap));
}

depth(tap)
char tap[256];
{
    int i,j;
    for (i = 0; tap[i] == ' ' ;++i) ()
    j = (i-7)/2 ;
    return(j);
}

essen(pat)
char *pat;
{
    pat = search(pat,"do ")+2;
    while ( *++pat == ' ');
    *index(pat,'¥n') = 0;
    return(pat);

}

blank(tap)
char tap[256];
{
    int i = 0;
    if (tap[i++] == '¥n') return(1);
    while (tap[i] == ' ' || tap[i] == '¥t'){
        if (tap[++i] == '¥n') return(1);
    }
    return(0);
```

cdiag.h

```
#include <stdio.h>
#include <ctype.h>

struct statg { char sprop[256] ;}  ;
struct factg { char fprop[256] ;}  ;
struct key { char prop[256];} ;

extern struct statg *stat;
extern struct factg *fact;
extern struct key *data;

extern int ftc,stc;

#define ESC      27
```