

## 深い知識に基づくドメイン特化型シェルの構築

田岡 直樹 山口 高平 溝口 理一郎 角所 収

大阪大学産業科学研究所

従来のエキスパートシステムは、専門家が持つタスクを直接的に処理する経験的知識（以後、浅い知識あるいは単にルールと呼ぶ）に基づいて構築されていたため、『予期していない（ルールがない）状況には対応不能』、『ルールの本質に関する理解が全くない』などの欠点を持ち、診断能力及び説明機能に限界があった。

しかし人間の専門家は、ドメインに関する基本的な原理・知識（深い知識）を持っているため、経験的知識の意味を理解している。また、未経験の問題に直面した場合でも、深い知識に立ち戻ることにより問題を解決し、その結果を経験則として蓄えることができる。すなわち、深い知識から浅い知識を生成する能力があると言える。したがって、エキスパートシステムに深い知識を持たせ、利用するとにより、上記の問題は解決できるものと考えられる。

本稿では、まず故障診断における深い知識を①部品の機能や構造を記述した対象モデル、②装置の振舞いを解析するための物理原理、③装置の状態を人間の故障及び異常徴候の概念にマッピングする解釈知識、④推論を制御するための制御知識の4種類に分類し、これらの深い知識から浅い知識を段階的に生成する機構を提案し、そのルール生成機構のインプリメント及び故障診断への応用について言及する。さらに、故障診断が困難な状況として、①対象モデルに明示的に記述されておらず、通常は無視されている要因(implicit-structure)の影響がある場合、②故障によって対象が変化し(mal-structure)、もとの対象モデルが診断に利用できない場合を考え、深い知識に基づいてその状況の診断を支援する機構について検討する。

## MACHINERY DOMAIN-SPECIFIC TOOL BASED ON DEEP KNOWLEDGE

Naoki Taoka, Takahira Yamaguchi, Riichiro Mizoguchi and Osamu Kakusho  
ISIR, Osaka Univ., 8-1, Mihogaoka, Ibaraki-shi, Osaka, 567, Japan

The following problems are caused in expert systems, which are constructed by the shallow knowledge (heuristics) approach.

1. There is no way to cope with a situation beyond heuristics.
2. There is no deep understanding in the domain.

It is suggested that deep knowledge approach can resolve these problems. However, there is little idea about what kinds of deep knowledge an expert system should have and about the details (implementations) of deep reasoning, which generates rules from various kinds of deep knowledge.

So this paper in detail discusses an idea about both deep knowledge and deep reasoning mechanism in "machinery domain". And the implementation is presented using such a knowledge-representation language that has amalgamated object-oriented concept on logic programming paradigm (Prolog). Furthermore the following topic is discussed. What it is caused by, the situation that human experts are apt to fails to diagnose? And it is shown that implicit-structure and mal-structure are key idea for solving the situation.

## 1. はじめに

従来のエキスパートシステムは、専門家が持つタスクを直接的に処理する経験的知識（以後、浅い知識あるいは単にルールと呼ぶ）に基づいて構築されていたため、

- ① 予期していない（ルールがない）状況には対応不能
- ② ルールの本質に関する理解が全くない

などの欠点を持ち、診断能力及び説明機能に限界があった。

しかし人間の専門家は、ドメインに関する基本的な原理・知識（深い知識）を持っているため、経験的知識の意味を理解している。また、未経験の問題に直面した場合でも、深い知識に立ち戻ることにより問題を解決し、その結果を経験則として蓄えることができる。すなわち、深い知識から浅い知識を生成する能力があると言える。したがって、エキスパートシステムに深い知識を持たせ、利用するとにより、上記の問題は解決できるものと考えられる。

筆者らは、異常の観点から機械の故障診断をタスクとした深い知識に基づくエキスパートシステム（あるいはシェル）の研究を続けており[1, 2]、文献[3]では深い知識を、対象モデル（Device World）、物理原理（Physical World）、解釈知識（Interpretation World）、制御知識（Control world）の4種類に整理し、深い知識から浅い知識（診断ルール）を生成する機構の構成について述べた。

本稿では、この4種類の深い知識のうち制御知識の内容および利用法の変更点、さらにオブジェクトの概念が記述できる論理型言語[4]によるインプリメントを中心にルール生成機構について述べると共に、専門家にとって診断が困難な問題としてimplicit-structure, mal-structureを例にあげ、深い知識に基づく診断支援について検討する。

## 2. 深い知識の整理

本説では、ルール生成機構で用いる4種類の深い知識をオブジェクトの概念が記述できるPROLOGによるインプリメントに関連づけて述べる。

### 2.1 対象モデル（DW）

診断の対象となる装置のモデルを対象モデルと呼び、これは以下の4種類の情報から構成される。

- ① 構造：部品間の接続関係や部品の配置
- ② 機能：部品を単独で考えた場合の一般的機能ではなく、設計者が意図した部品の役割
- ③ 動作環境：副作用的な機能、及び他の部品から受ける影響
- ④ 属性：各部品が持つ特徴的な物理パラメータ

ルール生成機構では、深い知識を利用する推論方式として定性的推論における制約伝播(propagation)[5, 6]が主として用いられるために、対象モデルは装置のふるまいを推論するために必要な物理パラメータを中心に記述される。実際には、図1の様に各部品の記述にポートの

概念を採用し、部品間の影響を示す機能(role)及び動作環境(environment)は物流（物理パラメータ）がポートを通してやり取りされる形式で、また構造(structure)は部品の持つポートの接続関係で記述することによって表現する。さらに、部品の属性(attribute)は物理パラメータをリスト形式で記述することにより表現する。

このうち構造は、装置が持つ物理パラメータ間の制約条件として深い推論に利用され、さらに機能、動作環境及び属性は、制約伝播の過程で制約条件となる物理式を物理原理から選出するために利用される。

```
1: instance(cylinder, components, role,
2:         [[in(1), temp1, cooling_water],
3:         [out(1), temp2, cooling_water],
4:         [out(2), q(-367), 'Q']]).
5:
6: instance(cylinder, components, ext_structure,
7:         [[in(1), water_pump, out(1)],
8:         [in(2), kongouki, out(1)],
9:         [in(3), piston, out(1)],
10:        [in(4), piston, out(2)],
11:        [in(5), engine_oil, out(1)],
12:        [in(6), piston, out(3)],
13:        [in(7), piston, out(4)],
14:        [out(1), thermostat, in(1)],
15:        [out(2), cooling_water, in(1)]]).
16: instance(cylinder, components, int_structure,
17:         [[in(3), in(2)], out(2)]).
18: instance(cylinder, components, environment,
19:         [[in(2), q(1), 'Q'],
20:         [in(3), f_q(2), 'Q'],
21:         [in(4), f, 'Q'],
22:         [in(5), myuu, 'Q'],
23:         [in(6), n, 'Q'],
24:         [in(7), s, 'Q']]).
25: instance(cylinder, components, attribute,
26:         [metal, f_q(-367)]).
27: instance(cylinder, components, parameter,
28:         [q(1), f_q(2), temp1, temp2, f, myuu, n, s]).
29: instance(cylinder, components, observability, 0).
30: instance(cylinder, components, durability, 9).
```

図1 対象モデル（DW）

### 2.2 物理原理（PW）

装置のふるまいを解明する様々な物理式を物理原理と呼ぶ。装置の状態解析において制約条件として用いられる物理式は、その物理式の適用時に値が与えられるパラメータ（原因となる変数）を右辺、その式を計算することによって値が求まるべきパラメータ（結果を表す変数）を左辺という形式で記述されている。

インプリメントでは、図2に示すようにマン・マシン・インタフェースとしての外部表現(External-expression)、対象モデルへの適用に必要な条件となる物理パラメータのリスト(Condition)、右辺にあるパラメータ（原因となる変数）のリスト(RHS)、左辺にあるパラメータ（結果を表す変数）のリスト(LHS)、値を伝播するときに用いられる両辺のパラメータ間の関係(Relation)から構成される。

```

1: instance(pw1,pw,external_form,'[]').
2: instance(pw1,pw,condition,[temp2,temp1,c,flow,q(1)]).
3: instance(pw1,pw,lhs,[temp2]).
4: instance(pw1,pw,rhs,[temp1,c,flow,q(1)]).
5: instance(pw1,pw,relation,
6:         [same(temp2,q(1)),
7:         inv(temp2,c),
8:         inv(temp2,flow),
9:         same(temp2,temp1)]).

```

図2 物理原理 (PW)

### 2.3 解釈知識 (IW)

ルール生成機構において装置内部の物理状態は、対象モデル内の物理パラメータの値によって表されるため、人間の持つ故障や異常徴候に関する概念との間にギャップが生じる。これを取り除くのが解釈知識であり、以下の2種類の知識がある。

①装置の状態を故障として解釈するための知識

②装置の状態を異常徴候として解釈するための知識

①に含まれる知識の中には、「どの様な壊れ方をしているのか?」というレベルの診断を可能にするために必要な知識が含まれる。

図3に示すように解釈知識は、対象モデルの知識をアクセスして初めて解釈が成立する知識と、直接状態を解釈できる知識に分かれる。

```

1: instance(iwh1,iwh,compoCondition,stuff).
2: instance(iwh1,iwh,parameterCondition,[c,minus]).
3: instance(iwh1,iwh,hypothesis,[impurity is mixed]).
4: instance(iwh1,iwh,setNo,set1).

```

図3 解釈知識 (IW)

### 2.4 制御知識 (CW)

人間が故障診断をする場合には、故障の可能性の高い部分から故障原因の探索を始める。このような探索を実現するには、故障の可能性に関する知識が必要であり、ルール生成機構でこの役割を果たすのが制御知識である。制御知識には、

①操作容易性: 物理パラメータの値の変化のしやすさ

②耐久性: 部品の耐久性 (寿命)

③観測容易性: 徴候の観測しやすさ

の3種類の情報がある。このうち、操作容易性(control-lability)は、状態解析において物理原理を制約条件として制約伝播を行うときに、値が変化しにくいパラメータに対して、値の伝播を制限するための知識である。この知識によって、可能性の低い故障仮説及び異常徴候の生成が抑制される。また、耐久性 (durability) は、深い前向き推論において耐久性の高い部品に対して解釈知識の適用を制限し、発生頻度の低い故障仮説の生成を抑制するための知識である。最後に、観測容易性 (observability) は、深い後ろ向き推論において、観測が困難な物理パラメータに対して解釈知識の適用を制限し、観測困難な異常徴候の生成を抑制するための知識である。

## 3. 診断ルールの段階的自動生成機構

前節で述べた4種類の深い知識から、診断ルールを段階的に自動生成するプロセスを図4に示す。ルールの生成過程は、制御知識の利用法を変更したことにより2段階で行われ、故障診断の前に行われる第1段階では発生頻度の高い故障を処理するルールを生成し、故障診断中に行われる第2段階では発生頻度の低い故障を処理するルールを生成する。

各段階は、与えられた異常徴候から故障仮説を生成する深い前向き推論及び生成された故障仮説が引き起こす異常徴候を導出する深い後ろ向き推論から成立する。このルール生成機構は、オブジェクトの概念が記述できるPROLOGでインプリメント中であり、深い前向き推論機構がほぼ完成している。

以下、第1段階を例にあげて、この2種類の深い推論について詳しく述べた後、第2段階のルール生成について述べ、さらにルール生成機構のインプリメントに言及する。

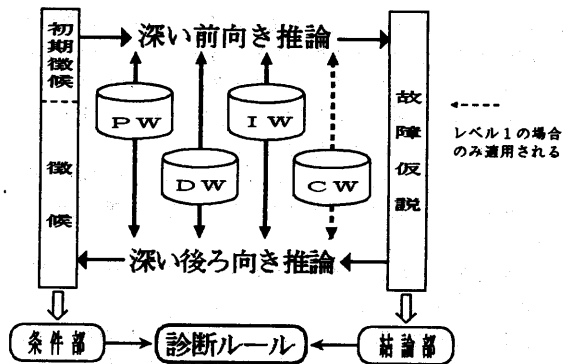


図4 診断ルール生成プロセス

### 3.1 深い前向き推論

深い前向き推論は、初期状態の生成、制約伝播による状態解析、故障仮説の生成という3つのプロセスから成立する。まず、初期状態の生成は、与えられた異常徴候を対象モデルの機能に関する情報により、物理パラメータに変換することによって行われる。

次に行われる制約伝播プロセスは、通常定性的推論場合と異なり、対象モデルにパラメータ間の制約条件が記述されていないため、物理原理からの制約条件の選出とその条件に基づいた物理パラメータの定性値の伝播の繰り返しになる。

定性値の伝播は、物理式の左辺の変数から右辺の変数に伝播され、故障原因が1箇所であると仮定して、右辺の変数のうち異常値 (+ または -) をとる変数は1つだけとし、残りの変数は基準値 (即ち 0) をとるように

値を決定する。第1段階のルール生成においては、この制約伝播のプロセスで制御知識が適用され、物理式の右辺にあっても変化しにくい物理パラメータには、異常な値を割り当てられず、値の伝播は第2段階までサスペンドされる。

最後に、新しく生成された状態に対して故障仮説生成用の解釈知識を適用し、適用可能ならばその状態に対して故障仮説を生成する。ただし、この過程も制御知識によって制限され、部品の耐久性が高ければ、解釈知識の適用が可能でも故障仮説は生成されず、第2段階までサスペンドされる。制約伝播は、解釈知識の適用の成否に関わらず続行される。

以上のプロセスを、新しい状態が生成できなくなるまで繰り返すことにより、与えられた異常徴候を引き起こし得る発生頻度の高い全ての故障仮説が生成される。したがって、深い前向き推論が終了した時点で、『もし、与えられた異常徴候ならば、生成された故障仮説』という形式の不完全な診断ルールが生成されたことになる。

### 3.2 深い後ろ向き推論

深い後ろ向き推論は、深い前向き推論で生成された故障仮説からの初期状態の生成、制約伝播による状態解析、解釈知識の適用による異常徴候の生成という3つのプロセスから成立する。まず、初期状態は、故障仮説を立てられた部品の機能と動作環境の否定を、物理パラメータに変換することにより生成される。

次に、制約伝播プロセスは深い前向き推論と同様、物理原理からの制約条件の選出とそれに基づく制約伝播の繰り返しになる。

定性値の伝播は、原因から結果を求めるために、物理式の右辺から左辺に伝播されるだけであり、右辺の他の変数には値は伝播されない。このとき、深い前向き推論と同様に制御知識の操作容易性が適用され、値の伝播が第2段階までサスペンドされる。

最後に、新しく生成された状態に対して徴候生成用の解釈知識を適用し、適用可能ならば異常徴候を生成する。ただし、この過程も制御知識によって制限され、新しい状態に解釈知識が適用可能でも、観測容易性が低ければ、解釈知識の適用は第2段階までサスペンドされる。制約伝播は、解釈知識の適用の成否に関わらず続行される。

以上のプロセスを、新しい状態が生成できなくなるまで繰り返すことにより、各々の故障仮説に関連するすべての異常徴候が生成される。最後に、この異常徴候と故障仮説を結合することにより、『もし、生成された異常徴候群なら、対応する故障仮説』という形式の完全な診断ルールが生成される。

### 3.3 第2段階のルール生成

前節までに述べた第1段階のルール生成では、制御知識を適用することによって発生頻度が低い故障状況処理する診断ルールの生成を抑制していた。これに対して第2段階では、第1段階で生成された診断ルールだけでは診断が不可能な場合に、第1段階でサスペンドされていた状態からの深い推論を行う。この第2段階のルール生成は第1段階で生成されたルールに基づく診断中に行われ、問題となっている徴候に関連した診断ルールのみを生成する。

診断ルールの生成法は、基本的に第1段階と同じであり、現在問題となっている異常徴候を入力として、その徴候と関連したサスペンド中の状態から深い推論を行うことにより、診断ルールを生成する。ただし、第1段階とは異なり、観測容易性による無意味な異常徴候の生成の抑制を除いては、深い推論の過程で制御知識の適用は行わない。この結果、耐久性の高い部品の故障に言及したルール、及び変動しにくいパラメータが変動した場合のルール、すなわち発生頻度の低い故障を処理するルールを生成したことになる。

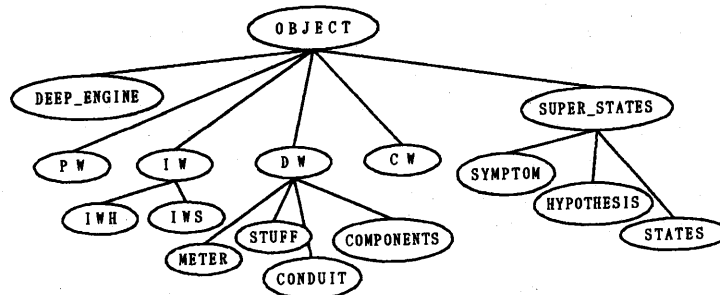


図5 ルール生成機構のクラス階層

### 3.4 段階的ルール生成機構の実現

オブジェクトの概念が記述できるPROLOGでインプリメント中のルール生成機構のクラス階層を、図5に示す。クラスDeep-Engineは、深い推論全体を管理しており、どのクラスに、どのようなメッセージを送れば、深い前向き推論及び深い後ろ向き推論が実現できるかを把握している。

状態を管理するためのクラスとして、Symptom, Hypothesis, Statesの3つがある。クラスSymptomは異常徴候を管理し、入力された異常徴候がインスタンスとして生成される。クラスHypothesisは故障仮説を管理するクラスであり、深い前向き推論で生成された故障仮説は、全てこのクラスのインスタンスになる。また、クラスStatesは深い推論中に生成される状態(ノード)を管理するクラスであり、推論中に生成された状態は、全てこのクラスのインスタンスになる。

知識に関するクラスとしてDW, PW, IW, CWの4つがあり、それぞれ具体的な対象モデル、物理原理、解釈知識、制御知識をインスタンスとして持ち、各知識の利用のされ方を把握している。

以下、深い前向き推論のプロセスについて説明する。まず、Deep-Engineにメッセージを送って知識生成を開始すると、深い前向き推論が起動される。

- ① Deep-Engineは、Symptomにメッセージを送り、ユーザに異常徴候の入力を促す。入力された異常徴候は、Symptomのインスタンスとして蓄えられ、DWを参照して物理パラメータの異常値に変換される。変換された結果はStatesに送られ、そのインスタンスとして初期状態が生成される。
- ② PWにメッセージが送られ、生成された状態の中で、新しく異常値が割り当てられたパラメータに注目して、制約条件として適切な物理式が選択される。選択された物理式を利用して、値の伝播が行われる。このときCWが参照される。
- ③ PWで制約伝播ができない場合は、DWにメッセージを送り接続関係などを制約条件として利用した制約伝播を行う。
- ④ 値の伝播によって、新しく生成されたパラメータの値をStatesに送ることにより、新しい状態をStatesのインスタンスとして生成する。
- ⑤ IWにメッセージが送られて、故障仮説の生成が試みられる。ここでも、CWが参照される。故障仮説が生成された後、新しい状態が生成できなくなるまで②に戻って制約伝播が繰り返される。

### 4. 段階的ルール生成機構における診断

前節で述べたように、第2段階のルール生成は第1段階のルールに基づく診断の失敗をトリガーとしており、最初の診断の後に実行される。すなわち、第1段階のルール生成後は、

①第1段階で生成されたルールによる診断

②第2段階のルール生成

③第2段階で生成されたルールも利用した診断

というプロセスを踏む。このうち、②については前節で述べたので、本節では診断の部分である①の第1段階で生成された発生頻度の高い故障を処理する診断ルール(レベル1のルール)による診断方法について述べた後に、③の第2段階で生成された発生頻度の低い故障を処理する診断ルール(レベル2のルール)に基づく診断について述べる。

#### 4.1 レベル1のルールに基づく診断

ルール生成機構では2段階のルール生成を行うが、これらは常に

(仮定1)故障が1箇所だけで発生している

(single-faultの仮定)

という仮定に基づいて行われる。したがって、ルール生成機構によって生成されたルールに基づく故障診断は、まず単一故障が発生していると仮定して行われる。診断は、従来のプロダクションシステムに基づくエキスパートシステムと同じく、入力された異常徴候と条件部が完全に一致するルールを適用することにより行われる。

このようなルールが存在しない場合に、既存のルールを利用して診断を行う様子を図6に示す。すなわち、このような場合には、仮定1が成り立たないと考え、

(仮定2)独立な故障が複数発生している

(独立なmultiple-faultの仮定)

と仮定して診断を続ける。ただし、生成されたルールの性質上、独立でない故障、すなわち複数の故障が相互作用によって新しい異常徴候を発生する場合は対象から除外する。この仮定によって、既存の診断ルールが利用できるようになる。

まずルールベースから、条件部が現在発生している異常徴候S1~S5の部分集合になっているルールR1~R3, R5を取り出す。次に、取り出されたルールを利用して、問題となっている全ての異常徴候を説明できる故障仮説の組合せG1, G2を作り、診断結果とする。この組合せが複数できる場合は、

(仮定3)1度に多くの故障が発生する可能性は少ない

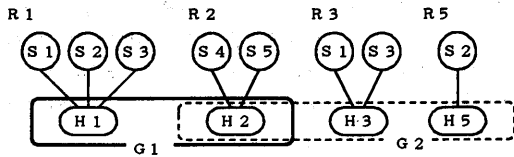
と仮定することにより、組合せに必要なルールの数が少ない組、すなわち考慮する故障仮説が少ない組G1を診断結果として出力する。

以上のように、レベル1のルールによって、単一故障及び独立な故障が複数発生している場合の診断が可能になる。しかし、レベル1のルールだけで診断が常に成功するわけではなく、失敗した場合には、第2段階のルール生成が起こる。

与えられた徴候



選ばれたルール



選ばれた故障仮説の組

図6 レベル1の診断プロセス

#### 4.2 レベル2のルールに基づく診断

レベル1のルールによる診断の失敗をトリガーとして、第2段階のルール生成が行われる。このレベル2のルールもレベル1のルールの場合と同様に、仮定1に基づいて生成されるため、レベル2のルールに基づく故障診断もレベル1と基本的に同じである。

まず仮定1に基づいて、新しく生成されたレベル2のルールから、入力された異常徴候と条件部が完全に一致するルールを適用することにより診断が行われる。

このようなルールがレベル2のルールにない場合の診断の様子を図7に示す。この場合は再び仮定1が成り立たないと考え、仮定2に基づいて診断を行う。まず、レベル2のルールベースからルールの条件部が、現在発生している異常徴候の部分集合になっているルールR25、R27を取り出す。

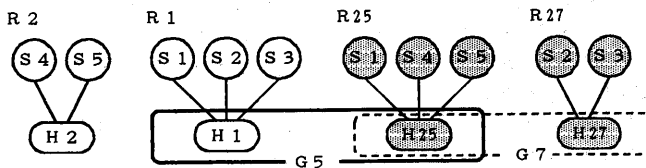
次に、ルールの条件部が上記の条件を満たすレベル1のルールR1、R2をあわせて、全ての異常徴候を説明できる故障仮説の組合せG5、G7を作り、診断結果とする。ただし、レベル1での失敗を繰り返さないために、診断結果となる故障仮説の組合せの中には、少なくとも1つはレベル2のルールによる故障仮説を含むようにし、R1とR2の組合せは考えない。

与えられた徴候



レベル1のルール: R1  
レベル2のルール: R25, R27

選ばれたルール



選ばれた故障仮説の組

図7 レベル2の診断プロセス

この組合せが複数できる場合は、やはり仮定3に基づき、生成される故障仮説が少ない組合せを診断結果とする。また、故障仮説の数が同じで診断結果が下せない場合には、レベル2のルールによって生成された故障仮説がもっとも少ないG5を診断結果とする。

以上述べてきたように、発生頻度の高い故障を処理する浅い知識(レベル1のルール)で診断ができない場合には、深い知識を利用して発生頻度の低い故障を処理する浅い知識(レベル2のルール)を生成し診断を行うことにより、深い推論と浅い推論を融合した診断が実現できる。

#### 5. 診断が困難な場合の支援

前節で述べたような、ルール生成機構を従来のエキスパートシステムに組み込むことにより、発生頻度の高い故障状況だけでなく、発生頻度の低い故障状況に対してもある程度対応が可能になる。また従来のエキスパートシステムと異なり、新製品や巨大プラントのように故障診断に関する経験則が十分に蓄積されていない分野でも故障診断が可能になる。しかしながら、依然としてルール生成機構のフレームワークでも取り扱えない問題が存在する。

本節では、現在のフレームワークで対応できず、専門家が本当に支援を必要とする診断が困難な場合として、implicit-structure, mal-structureの2つの場合を考え、深い知識に基づく診断の支援について検討する。

##### 5.1 診断が困難な故障

通常、人間の専門家は経験的知識を使って故障診断を行っている。この診断の過程において、

『故障と診断した部品が実は誤動作であり、故障してはいない。』

という状況に直面する場合がある。ここで誤動作とは、動作中は何等かの影響で異常な動きをしているが、停止させてチェックしてみるとその影響が消えているために、故障していない場合のことである。この場合、その部品を誤動作させている原因を探索しなければならない。また、

『診断は正しかったが、同じ部品が度々壊れる。』  
というような状況もあり、この場合はその部品を壊す原因を解明し、対策を施す必要がある。本節では、この様な状況を『診断が困難な場合』と考える。

診断が困難な状況に陥る原因としては、人間自体に問題がある場合と、診断対象に問題がある場合が考えられるが、前者は考えず、後者についてのみ考察する。

対象に問題がある場合としては、

- ①無視できるはずの影響が無視できない  
(implicit-structure)
- ②故障により対象が変化してしまっている  
(mal-structure)

などが考えられる。これらの場合は、計算機だけではなく専門家にとっても診断が困難である。しかし、通常の故障診断に関する知識が全く役に立たないわけではなく、何らかの方向付ができると考えられる。

以下、このように診断が困難で、専門家が本当に支援を必要とする場合に、深い知識によってどのような診断支援が可能であるかを考察する。

### 5.2 無視できる影響を考慮した診断

診断が困難な場合の1つとして、通常無視できる要因の影響が考えられる。この影響を処理するために、まずルール生成機構が持つ深い知識において通常無視されている要因を考える。

ルール生成機構が持つ4種類の深い知識のうち、解釈知識と、制御知識には、通常無視されている情報というものはない。しかし、物理原理においては、近似式で通常無視している項、対象モデルにおいては、通常記述されない暗黙的な構造(接続関係)、副作用的な機能がこの要因として考えられる。通常記述されない暗黙的な接続関係として、空間を通した影響が考えられる。また、無視されている副作用的な機能は、ここでは考慮の対象から外す。

以上の結果、ルール生成機構が持つ深い知識で通常無視されている影響として、

#### ①近似物理式に現れない項の影響

②熱、光、電磁波などが空間等を通して与える影響があることが分かった。このうち①は、物理原理に近似を含まない物理式を記述すれば、現在のフレームワークで処理できる。これに対して②は空間、すなわち対象モデルに構造として明示的に記述されていない接続関係(implicit-structure)を通して受けるため、従来のフレームワークでは処理できない。以下、空間を通した影響として、外部からの影響(使用環境による影響)は考えず、装置内部からの影響のみを対象として、故障診断の支援について検討する。

診断が困難な場合においても、既に生成された診断ルールが全く役に立たないわけではなく、度々壊れる部品あるいは誤動作している可能性のある部品を発見するために利用できる。この診断の様子を図8に示す。

まず、レベル1, 2の診断結果を利用して、度々壊れたり誤動作している部品H1, H2をリストアップする。次に、空間的影響を及ぼす部品、すなわち熱や電磁波の発生源となる部品C1~C5が、部品H1, H2にどれだけ影響を与えるかを図9に示すような評価式によって評価する。この結果、度々壊れる部品、あるいは誤動作している部品H1, H2に許容範囲以上の影響を及ぼしている部品C2, C3, C5を求め、全ての故障状況を説明できる故障原因の組(C2, C5), (C3)を作り、診断結果とする。

この組合せが複数できる場合は、そこに含まれる故障原因の数が最小となる組合せ(C3)を故障原因とする。また、原因となる影響の発生源の数が同じ場合は、影響の大きいものを含む方を優先する。

このような診断を行うためには、深い知識の記述を拡張する必要がある。まず対象モデルには、どの部品が影響の発生源となり得るか、その部品と誤動作をしている部品の間の距離、その部品間に存在する他の部品の寸法・材質などに関する情報を記述しなければならない。また、物理原理にその影響を評価するための評価式が必要となってくる。さらに、評価に必要な物理定数、例えば熱の場合であれば、材質による熱伝達率や熱伝導係数などが必要となる。

### 5.3 構造の変化を考慮した診断

故障診断が困難な場合のもう1つの例としては、故障によって装置の構造が変化し(mal-structure)、対象モデルが対象そのものを表現するものではなく、診断に利用できないケースが考えられる。

このような場合の処理は、基本的には空間的影響を考慮する場合と同じである。まず、既存の診断ルールを用いることにより、誤動作あるいは度々壊れると考えられる部品あるいは部品群をリストアップする。次に、

(仮定4) mal-structureは、隣接部品間にしか生じない

という仮定を設け、隣接部品からの影響を考慮する。しかし、mal-structureの場合はimplicit-structureの場合と異なり、この影響を評価する尺度になるものがない。したがって、誤動作あるいは度々壊れる原因となった(解釈知識の適用を受けた)物理パラメータに関連がある隣接部品をmal-structureの原因だと考え、全ての故障状況を説明できる組合せを作り、診断する。この組が複数できる場合は、やはり故障原因の数が少ない組合せを診断結果とする。

この診断は、誤動作している部品に隣接している部品の中から、誤動作の原因になりそうな部品を取り出すため、診断結果の信頼性が乏しく、強力な手法ではないが、診断の糸口がつかめないような状況では、専門家に対する支援になるものと考えられる。

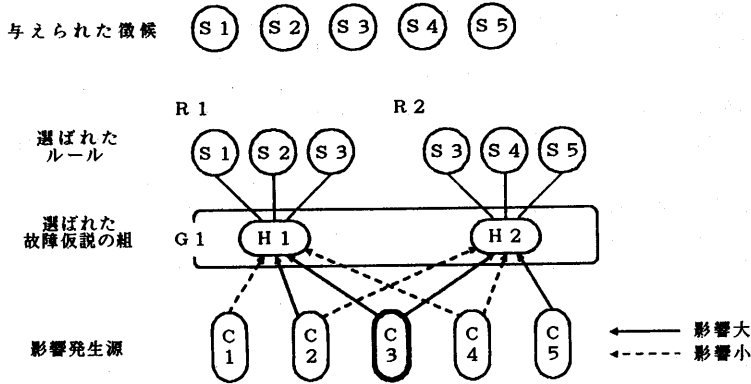


図8 診断が困難な場合の処理

$$f = \frac{T}{\sum_i \frac{1}{h_i} + \sum_j \frac{\delta_j}{\lambda_j}}$$

T: 高温部の温度     $h_i$ : 熱伝達係数  
 $\delta_j$ : 物質の厚さ     $\lambda_j$ : 熱伝導係数

図9 熱の影響の評価式

## 6. おわりに

本稿では、機械の故障診断にタスクを限定し、4種類の深い知識、すなわち対象モデル、物理原理、解釈知識、制御知識から浅い知識である故障診断ルールを生成するルール生成機構を実現すると共に、本フレームワークを人間が本来支援を必要としている診断が困難な場合、すなわちimplicit-structure及びmal-structureまで取り扱えるように拡張を検討した。

後者は、文献[7]で言及している故障メカニズムを解明する故障物理 (Failure Physics) と関連が深く、今後検討を加えて行きたいと考えている。

## 参考文献

- [1] 山口, 溝口, 杉原, 小高, 野村, 角所: "エキスパートシステムにおける高級な説明機能" 情報処理学会第30年全国大会, pp. 1551-1552 (1985-3)
- [2] 山口, 溝口, 小高, 野村, 豊田, 角所: "深い知識を利用した知識コンパイラの構成" 電子通信学会, 人工知能と知識処理研究会資料, AI86-7 (1986-4)
- [3] 小高, 野村, 田岡, 山口, 溝口, 角所: "知識コンパイラの構成とその応用" 情報処理学会, 知識工学と人工知能研究会, 48-2, (1986-9)
- [4] 小高, 溝口, 山口, 川口, 野村, 角所: "オブジェクト指向の概念を導入したPrologとその支援環境" 情報処理学会第31年全国大会, pp. 955-956 (1985-9)
- [5] de Kleer, J. and Brown, J. S.: "Qualitative Physics Based on Confluences", Artificial Intelligence, 24, 1, pp. 7-83, (1984)
- [6] Kuipers, B. J.: "Qualitative Simulation", AI 29, pp. 289-338, (1986)
- [7] 越川, 植草, 村田: "実務にすぐ役立つ信頼性技術" 日刊工業新聞社, (1982)