

ACOS-4 性能改善 エキスパートシステムの開発

麻生川 稔^{*} , 白水 明^{**} , 伊久美 功一^{**} , 久保 秀士^{*}

^{*}日本電気(株) C&Cシステム研究所
^{**}日本電気(株) 基本ソフトウェア開発本部

ACOS-4を使用しているときに発生する性能問題について診断を行いその診断結果に基づき具体的な改善方法を示すエキスパートシステムプロトタイプをEXCORE(エキスパートシステムの構築ツール)を用いて作成した。性能分析に必要なデータは半自動的に取り込むことが可能なので、性能問題の専門家でなくても使用することができるようになっている。専門家の知識を、性能問題の原因がどの部分にあるか見つける初期診断と、初期診断の結果にしたがって分析をする診断部と、診断の結果にしたがって対策を示す対策部の3つにグルーピングした。各部分をさらに階層別に分けた。この様に構造化することにより、知識の表現が容易になった。

A PERFORMANCE IMPROVEMENT EXPERT SYSTEM FOR ACOS-4 OPERATING SYSTEM

Minoru Asogawa^{*} , Akira Shiramizu^{**} , Koichi Ikumi^{**} , Hidehito Kubo^{*}

^{*}C&C Systems Reserach Laboratories, NEC Corporation
^{**}Basic Software Development Division, NEC Corporation

^{*}1-1, Miyazaki 4-chome, Miyamae, Kawasaki, Kanagawa 213 Japan
^{**}10, Nisshin-cho 1-chome, Fuchu, Tokyo 183 Japan

We've constructed a prototype performance improvement expert system for ACOS-4 operating system, using EXCORE (expert system constructing tool) developed by NEC. In this system, causes of unfavorable performance are examined, and definite concrete plans to remove these causes are given. Because this system can acquire needed data semi-automatically, it can be used without difficulty by system engineer not familiar with performance analysis. Expert's knowledge is divided into following three groups.

1. Rules in the consulting group list up all suspected causes of unfavorable performance by consulting with system engineer.
2. Rules in the diagnosis group determine suspected causes and determine the fundamental causes.
3. Rules in the counter plan group propose counter plan for each fundamental causes.

In each group, knowledge is divided into layers. By using these structured expression, expert's knowledge is expressed without difficulties.

1. はじめに

本論文では、稼働中のコンピュータシステムの性能分析に着目する。コンピュータシステムの運用に於て重要なことは、ハードウェアを有効に利用しながら、業務のサービス（スループットとレスポンス）を保証することである。業務（業務量または業務の種類）を変更するとサービスを保証できなくなることが多い。性能分析は、その保証できなくなる真の原因を調査するために重要である¹⁾。

従来の性能分析作業は、コンピュータ利用者（ユーザー）側の運用専門家とメーカー側のシステム専門家が協力して、性能上のボトルネックを解決する具体策を見つける努力を行ってきた。しかし、最近のコンピュータシステムのより一層の複雑化・多様化は、これらの専門家達でさえも、システム稼働状況を伝えるデータ群の中から何が異常かを検出したり、何がボトルネックかを解析したり、解決の具体的対策を見つけたりという作業を非常に負担のかかるものにしてしまった。このことを解決するためには、専門家達が持っている広範囲な専門的知識と、知識を系統的に組み合わせ判断するといった能力が必要である²⁾。

この状況を改善するために、コンピュータに対し、性能分析と対応に必要な広範囲な知識を与えることにより、性能分析を行っている専門家の業務の大部分を代行させるという意図で開発しているのがACOS-4性能改善エキスパートシステムである。

ACOS-4性能改善エキスパートシステムは、エキスパートシステム構築用ツールとしてEXCOREを採用している。EXCOREは日本電気（株）が昭和60年10月に発表したエキスパートシステム構築ツールで以下のような特徴を持っている³⁾。

- ①豊富な知識表現（フレーム型、述語型、ルール型、メタルール記述）。
- ②柔軟な推論による高度なシステムの実現（前向き推論、後向き推論、曖昧さを取り入れた推論、を組み合わせた推論）。
- ③オブジェクト指向と知識ベース管理による高い拡張性。
- ④既存ソフトウェアとの融合ができる（データベース、FORTRANのプログラムとの融合）。

性能改善エキスパートシステムでは、以下の理由からEXCOREを採用した。

①実際に存在する計算機の各装置を取り扱うためにオブジェクト指向の知識表現が必要であり、診断・対策を行う上で前向き推論と後向き推論をすることが必要である⁴⁾。

②ACOS-4のパフォーマンス測定ツールを使用するため、構築ツールとして同じACOS-4上で動作する事が望ましい。

ACOS-4性能改善エキスパートシステムの特徴は、以下の通りである。

- ①パフォーマンス測定ツールからデータを半自動的に取り込むことができる。
- ②対話形式でユーザーから入力された性能改善の方針を考慮した、具体的な対策を示すことができる。
- ③性能問題の専門家でない人（SE）でも、対話処理形式で簡単に実施できる。

2. コンピュータシステムの性能問題とは

コンピュータシステムの性能問題は、ユーザーの立場より、システムの性能低下、特定のワークの性能低下の現象として捉えられる。システムの性能としては、オンラインデータベースシステム（VIS）のレスポンス、TSSのレスポンス、バッチジョブのスループット、これらを含めたシステム全般を考えることができる。また、特定のワークとしては、VISTランザクション、TSSコマンド、バッチジョブを考えることができる。現象としては、特定のジョブあるいはランザクションの実際の処理時間が予定された（見積られた）処理時間より長くかかるということである。

ジョブあるいはランザクションの最小限必要な処理時間は、まさにそのジョブやランザクションの処理に必要なCPUや外部記憶装置の使用時間によって規定される。実際には、多くのジョブあるいはランザクションがCPUや主記憶装置を共有して実行される。こうした状況のもとではいたるところでCPUや主記憶装置そして外部記憶装置等の資源に対する競合が発生し、この時発生する待ち時間がそのままジョブやランザクションの処理時間に反映され（加算され）先に定義したコンピュータシステムの性能問題として表れてくる。この資源に対する競合によるコンピュータシステムの性能問題の発生の原因としては以下の3つが挙げられる。

- ①資源が絶対的に不足している。
- ②資源量としては充分であるが資源の割当方法が良くない、またはワークロードの与え方が良くない。
- ③資源量としては充分であり、資源を使用する方法も管理されているが、個々のジョブやトランザクションが必要以上に（無駄に）資源を使用するように設計されている。

一見したところ①が原因のように見えても分析していくと真の原因は②あるいは③にあるという場合も多い。例えば、総ての磁気ディスクの使用率が高いときは、磁気ディスク装置の台数が不足しているように見える。しかし、主記憶装置の割当が適当でなくて、仮想記憶システムに於てページの入出力が多く磁気ディスク装置の使用率を上げていることが真の性能劣化の原因である場合もある。この場合、メモリーの割当を調整することにより、磁気ディスク装置の使用率を下げる事ができる。

性能を解析するためには、被診断計算機の資源使用状況を知る事が必要不可欠である。ACOS-4では、運用管理の機能としてシステムパフォーマンス測定ツール(SMF)が供給されている。SMFでは、システムの運用管理データを統合管理し、稼働統計情報等の種々の情報を収集している。例えば、システム全体の情報としてCPU時間、入出力装置の利用状況などを収集し、ジョブあるいはステップ情報として、CPU時間、使用装置、入出力回数などを収集している⁵⁾⁶⁾。本エキスパートシステムでは、SMFの情報を半自動的に取り込む機能を持っているので、性能問題の専門家でない人でも使用することができる。

ACOS-4性能改善エキスパートシステムは、SMFの情報を半自動的に取り込み、解析を行い真の原因を割出し、利用者によるその解決方法についてのアドバイスを呈示するエキスパートシステムである。

3. 試作した性能改善システム

3.1 診断・対策対象領域

このシステムで診断・対策の対象とする領域は、サブシステムに固有の性能問題(スループット分析、レスポンス分析:オンラインデータベースシステム(VIS), TSS, バッチジョブ)と資源の使用状況(CPU, 仮想記憶システム, 外部記憶装置)とに分けられる。前者の診断・対策をするためには、後者の診断・対策をすることが必

要であることが多い。

3.2 システム構成

このシステムは、①初期間診部、②性能問題診断部(診断部)、③性能問題対策部(対策部)、④データ部、の4つの部分より構成されている(図1参照)。

診断・対策の流れは、以下の通りである。

- ①初期間診部が起動されて、性能劣化の原因であると疑われる部分を、総てリストアップする。
- ②各々の疑われている性能劣化の原因について、それを診断することができる診断部が起動される。
- ③診断部が診断を行い性能劣化の原因があると診断された場合、具体的な性能劣化の原因の箇所を指摘する。
- ④診断部が診断を行った結果、他の性能劣化の原因の疑いが発生した場合、その性能劣化の原因の疑いについて他の診断部が起動される。
- ⑤対策部では、指摘された性能劣化の原因に対して具体的な対策方法を示す。
- ⑥データ部は、初期間診部、診断部、対策部より推論実行中に随時アクセスされ必要なデータを供給する。SMFのデータ等はあらかじめ取り込まれている。

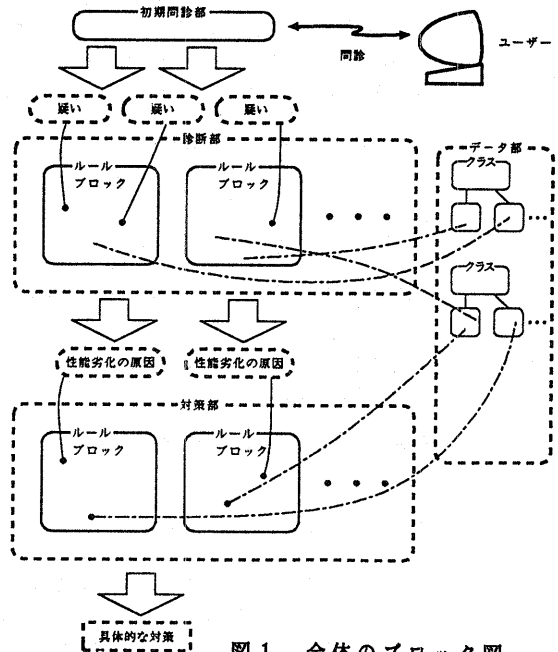


図1 全体のブロック図

初期問診部、診断部、対策部は、それぞれいくつかのルールブロックから成り立っている。各ルールブロックは、複数のルールから成り立っており、IF・・・THEN・・・の形のプログラクシヨンルール形式で表現されている。データ部は、フレーム形式と述語形式で表現されていて、保存するデータの種別に応じて形式を使い分けている。フレーム形式に於ては、EXCOREのオブジェクト指向を用い高記述性を生かしている。

3.3 知識の表現法

試作した性能改善エキスパートシステムでは、知識の表現にEXCOREのオブジェクト指向性を有効に用いている。具体的には、①フレーム表現、②関係表現、③述語表現、④ルール表現を用いている。以下に各表現の使い分けについて説明する。

①フレーム表現

被診断計算機を構成する装置（e.g.磁気ディスク処理装置、磁気ディスク装置）の機能の表現と、SMFより得られた性能データの表現と、SMFデータを表現するクラス、をスーパークラスに持つクラスのインスタンスとして表現されている。性能の規格を表現するクラス、SMFデータを表現するクラス、をスーパークラスに持つクラスのインスタンスとして表現されている。性能の規格を表現するクラスでは、性能の規格の値と、それらを処理する手法（メソッド）が定義されている。SMFデータを表現するクラスでは、SMFデータの値と、その値を処理する

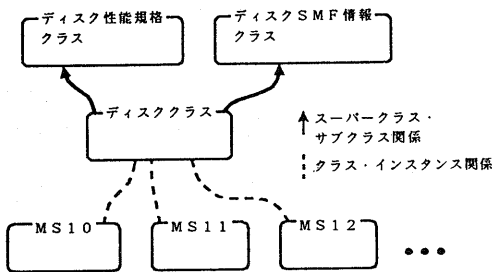


図2 知識のフレーム表現

メソッドが定義されている。このクラスを親に持つクラスは、これらのメソッドを継承する。そのクラスのインスタンスも、これらのメソッドを継承するので、そのインスタンスにメッセージを送信することによって性能規格、SMFデータの処理を行うことができる。この様にフレーム表現では、オブジェクト指向を用いている。

②関係表現

関係表現では、フレーム表現でインスタンスとして表現されている計算機を構成する各要素間の接続状況を示す。たとえば、磁気ディスク装置と磁気ディスク処理装置は、配下関係にあると表現する（図3参照）。

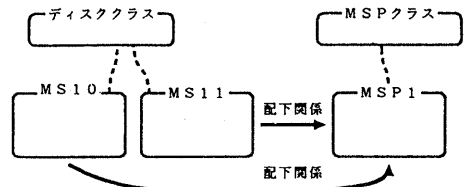


図3 知識の関係表現

③述語表現

述語表現は、推論の中間結果を表現するのに用いられる。疑い、調査結果、診断結果、性能劣化の原因、対策候補リスト、対策候補、対策等の述語名を用いている。推論中は、述語はワーキングステージと呼ばれる領域に保存されて、ルールから参照されたり、登録されたりする。例えば、調査の結果としてMS10という磁気ディスクのBUSY率が40%であるという事は、
(: 調査結果 MS10 ディスクのBUSY率 40%)
の様に表現される。

④ルール表現

ルール表現の例をルール1に示す。診断と対策の手続き的な知識は、ルールで表現されている。ルールは、プログラクシヨンルール形式で表されている。例のルールの条件部では、

| | |
|---|-----|
| IF (: 疑い ?ディスク ディスクボトルネック) | 条件部 |
| THEN (&CREATE-PREDICATE ; 述語の作成関数 (: 調査結果 ?ディスク ディスクのBUSY率 (...ディスクのインスタンスのBUSY率を計算するメソッドを実行関数...)) | 実行部 |
| ルール 1 | |

述語名が“疑い”というワーキングステージ上の述語に対して、マッチする述語が捜される。その結果、マッチするものが存在する場合、“?ディスク”という述語変数には、述語中で対応する値が代入され、ルール1の実行部が実行される。実行部では、述語を作成するLISP関数を実行している。一般に実行部では、LISPの各関数を実行することができる。

3.4 機能概要

3.4.1 初期問診部

初期問診部は、1つのルールブロックで構成されている。推論は、SEが把握し得るデータをもとにして対話形式で診断を進めてゆき、その結果性能劣化の原因がどの部分にあるか見当をつける(疑いの抽出)。

この様な大局的な診断では、性能問題をどの様にして切り分けるかが重要な問題である。後で起動される診断部では初期問診部で抽出された疑いに対して更に詳しい解析を行うので、初期問診部では性能問題がどの部分にあるかを漏れなくリストアップすることが必要である。従って、初期問診部では、“疑わしきものは、総て疑う”という方針を取ることが重要である。また、SEが把握し得るデータは、SEの能力と経験に左右されるので、初期問診部で得られるデータの量は不定である。推論に必要な値がSEによって把握されていないこともありうる。

この様な場合は、前向き推論よりも後向き推論が適している。この様な特徴を持つ初期問診部では、“深い”知識よりも“直感的”な知識を多用している。“直感的”なルールを用いた場合は、実行されるルールの数が“深い”ルールを用いた場合に比べて少ないために実行時間が短縮される。問題の複雑さがルールに反映しないので管理が容易である。反面、“直感的”なルールは、専門家の幅広い経験を反映しなくてはならない。この様な機能を持つ初期問診部を“個々の”診断部に先駆けて実行することによって、性能問題の対象を絞り込むことができ、システム全体の実行時間を短縮することができる。この初期問診部で扱う疑いの数は、10個程度である。

3.4.2 性能問題診断部

診断部は前向き推論で行われており、初期問診部によって作成された性能劣化の原因の疑いについて診断する機能を持つルールブロックで構成されている。各ルールブロックは初期問診部で作られた性能劣化の原因の疑いをもとにして起動される。疑いについてあるルールブロックで対応できない場合は、他のルールブロックを起動し対応する。診断部では、あらかじめ取り込まれたSMFのデータをもとにして解析を進める。診断は図4に示すように行われる。ルールブロック中のルールは3つの種類に分類できる。診断についての専門家の知識を3つの種類に分類することによって、専門家の知識を表現しやすくしている。以下、例を示しながら各ルールについて説明する。

①性能問題の疑いに対して必要な調査を行い、その結果を登録するルール(ルール2参照)

性能問題の疑いは、述語形式で表現されている。この種のルールでは、性能問題の疑いを明らかにするためにどの様な調査が必要であるかという専門家の知識を表現している。

実際のルールでは、ルールの実行部に必要な調査を実行しその結果を述語形式で登録する。調査は、ルール実行部中のLISP関数を実行することによって得られる。ほとんどの場合、対応するオブジェクトにメッセージを送信しその返却値を値とした述語を生成する。調査結果は、判断をまじえない値(数値等)である。

②調査結果から新しい疑いの作成、診断結果の登録をするルール(ルール3参照)

この種のルールでは、数個以内の調査結果に判断を加え、それを診断結果として登録する。この判断は、専門家の限定された場面に適用される知識に基づいて行う。従って、この知

```

IF (: 疑い ?ディスク ディスクボトルネック)
THEN (&CREATE-PREDICATE
      (: 調査結果 ?ディスク ディスクのBUSY率
        (...ディスクのBUSY率を計算する関数...)))

```

ルール2

```

IF (: 調査結果 ?ディスク 平均シーク距離 ?平均シーク距離)
(: 調査結果 ?ディスク トラック数 ?トラック数)
(&> ?平均シーク距離 (&÷ ?トラック数 3))
THEN (&CREATE-PREDICATE
      (: 診断結果 ?ディスク シーク距離 大きすぎる))

```

ルール3

識により得られた診断結果は、性能劣化の原因を直接指し示しているとは限らない。実際には他の部分が原因であるのだが、その影響が診断結果として表れることがあるからである。これらの診断結果をもとにして、総合的に性能劣化の原因が判断される。

この時に、判断の基準となるものは、あらかじめルールの中に埋め込まれている。判断基準をルールに埋め込まずに、外部から変数として与える方法がある。この場合には複数の判断基準を組み合わせて一つの判断基準とすることが必要である時には、判断基準の組合せが増えるために管理が複雑になる。

また、調査結果により、新たに別の疑いを登録するルールもある。これは、ある疑いと調査結果をもとにして、他の部分が疑わしいと判断することができるときである。ここで作成された疑いは、そのルールブロック内で対応している場合と、していない場合がある。している場合は、その疑いに対応するルールが起動される。していない場合は、疑いに対応しているルールを含む他のルールブロックが起動される。

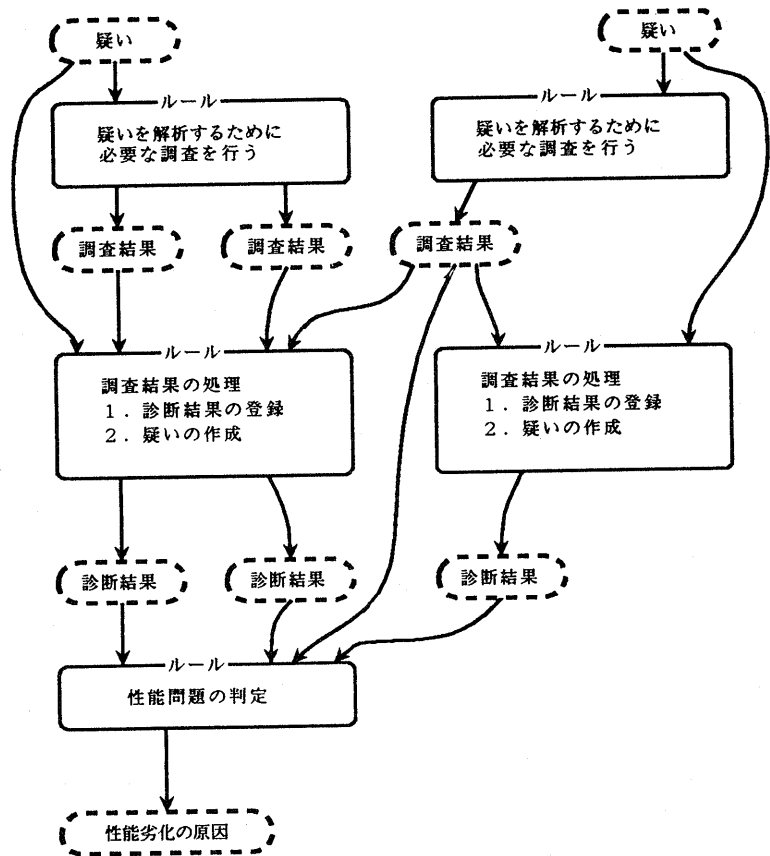


図4 診断部の内部構造

③性能劣化の原因を判定するルール（ルール4参照）

この種のルールは、いくつかの診断結果を総合的に判断して、具体的に何が性能劣化の原因であるかを指摘するルールである。このルール実行の結果、性能劣化の原因についての述語が作成される。この述語は、対策部で対策を行うためのものとなる。このルールでは、診断結果より性能劣化の原因かどうかを総合的に判定する専門家の知識が表現されている。

```

IF (: 診断結果 ?ディスク シーク距離 大きすぎる)
(: 診断結果 ?ディスク BUSY率 正常)
THEN (&CREATE-PREDICATE
(: ボトルネック ?ディスク シーク距離大きすぎる))
    ルール4
  
```

3.4.3 性能問題対策部

対策部は前向き推論で行われており、診断部で作成された性能劣化の原因について、具体的な対策を示す機能を持つルールブロックで構成されている。各ルールブロックは、対策部で作られた性能劣化の原因をもとにして起動される。対策は図5に示すように行われる。ルールブロック中のルールは3つの種類に分類できる。対策についての専門家の知識を3つの種類に分類することによって、専門家の知識を表現しやすくしている。以下、各ルールについて説明する。

①ボトルネックに応じて、総ての対策候補のリストを登録するルール（ルール5参照）

この種のルールでは、診断部で示された具体的なボトルネックに対して、ボトルネックを解消できる可能性のある総ての対策を、対策候補のリストとして登録する。具体的には、対策候補をリストとして含む述語を作成する。あらかじめ対策候補は、実行しやすい順に並べられている。この並びには、専門家の知識が表現されている。

②対策候補のリストのうちから、ユーザーの方針にしたがって実行すべき対策候補を選択するルール

この種のルールは、対策候補のリストから1つの対策候補を選択するルールである。この時判断の基準として、あらかじめ登録された、または対話的に得られたユーザーの方針にしたがって、1つの対策候補を選択する。2回目にこのルールが起動された場合は、次の対策候補を選択する場合、ユーザーの方針を取り入れる場合と取り入れない場合がある。

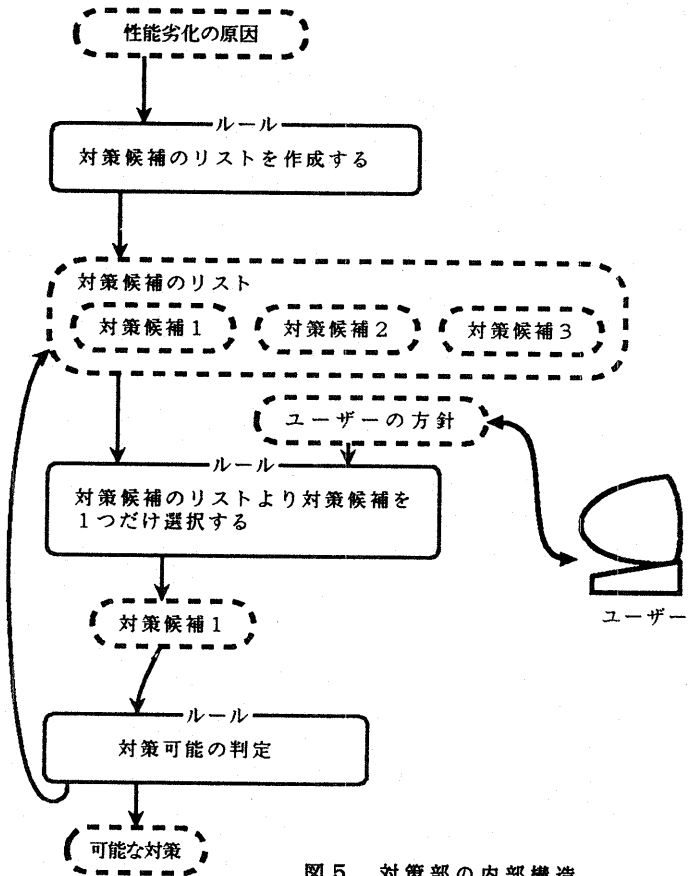


図5 対策部の内部構造

③選択された対策候補が可能であるかを判定するルール

選択された対策候補は、計算機の状態により、実行困難であることがある。その判定には、診断部で作られた、調査結果または診断結果等の情報をもとにして判断し、その結果を述語として登録する。選択された対策候補が実行不可能である場合、対策候補リスト上のその対策候補は実行不可能であるというフラグがつけられ、対策候補のリストから新たに別の対策候補が選択される。

```

IF ( : ボトルネック ?MSP BUSY率高すぎる))
THEN (&CREATE-PREDICATE
      (: 対策候補リスト ?MSP BUSY率高すぎる
        (ディスクの新規購入 配下ディスク移動 ファイルの移動)))
      ルール5
  
```

4. 使用例

プロトタイプの使用例として、磁気ディスク処理装置 (MSP) が性能劣化の原因となっているときの診断部と対策部の推論の一部を示す。MSP のオーバーロードを避けるために、そのMSP配下の磁気ディスク装置を他のMSPの配下に移動することを試みている。

5. 結び

本システムでは、一般に中間仮説と表現されている部分を、その種類に応じて述語で表現することにより、ルールの管理を容易にしている。プロトタイプ作成に要した時間は、約10人月程度である。この様な診断システムは、知識を変更することによって容易に他のシステムに適応可能である。プロトタイプのルール数は約350ルール、フレームのクラス数は約55である。

これからの拡張方向としては、診断部、対策部のルールブロックの数を増やし綿密な推論を行うことがある。また、ユーザーに実際に使用してもらいその評価をフィードバックすることにより、より実地的なシステムにしていきたいと考えている。

計算機のMSPシステム全体がボトルネックであると疑われています
システム全体のMSP (MSP5 MSP4 MSP3 MSP2 MSP1) がボトルネックであると疑います

MSP (MSP1) がボトルネックであると疑われています
ボトルネックかどうかを調べるためにMSP (MSP1) のBUSY率を調査します

MSP (MSP1) がボトルネックであると疑われていて、BUSY率が30%以上です
MSP (MSP1) がBKSTとSWP関係のIOが多すぎる疑いがありますさらに、MSPオーバーロードである疑いがあります

MSP (MSP1) がボトルネックであると疑われていて、BUSY率が30%以上です
MSP (MSP1) がBKSTとSWP関係のIOが多すぎる疑いがありますさらに、MSPオーバーロードである疑いがあります

MSP (MSP1) がボトルネックであると疑われています
ボトルネックかどうかを調べるためにMSP (MSP1) のBUSY率を調査します
MSP (MSP1) にBKSTとSWP関係のIOが多すぎる疑いがあります。
VMM関係のルールブロックを呼び出します。

MSP (MSP1) にBKSTとSWP関係のIOが多すぎる事が無く、MSPのオーバーロードが性能劣化の原因であると診断されました。
MSPの対策ルールブロックを呼び出します。

MSP (MSP1) システムがオーバーロードであると診断されています
その対策としてMSPの配下のDISKを移動することによってMSPの負荷を下げます

MSP (MSP1) がオーバーロードであると診断されています。配下のDISKを移動することによってMSPの負荷を下げる事が対策の目標です。
MSPのBUSY率を下げる為にDISKを移動することを目的とします。その為の初期調査を行います。

システム全体のBUSY率にまだ余裕があり、DISKを分散してMSP (MSP1) のBUSY率を下げる事が分かりました。
DISKを分散することによってMSP (MSP1) のBUSY率を下げることに可能性があることが分かりました。

MSP (MSP1) 配下のDISKを分散することに可能性が有ります
移動できるDISKの候補のリストを作成します

MSP (MSP1) の配下で移動できるDISKの候補が有ります
MS04を移動した時の状態を調査します

MSP (MSP1) の配下のDISK (MS04) を移動させることを試みています。
DISKをMSP配下より外したときのBUSY率を調査します

DISK (MS04) をMSP (MSP1) より外したときのBUSY率が30%以下です。
このDISKを移動した時、MSPのBUSY率の改善はできる可能性が有ります。移動する先のMSPをリストアップする。

移動する先のMSPがリストアップされている。
MSP5を移動する先のMSPとして調査します。

DISKを移動させたときのMSP (MSP2) のBUSY率について調査します。

DISK (MS04) をMSP (MSP1) からMSP (MSP2) へ移動させた時のBUSY率が30%以上です。
DISK (MS04) をMSP (MSP2) へ移動できない事が分かりました。

DISK (MS04) をMSP (MSP1) からMSP (MSP3) へ移動させることを試みています。
DISKを移動させたときのMSP (MSP3) のBUSY率について調査します。

DISK (MS04) をMSP (MSP1) からMSP (MSP3) へ移動させた時のBUSY率が30%以上です。

図6 使用例の一部

<<参考文献>>

[1] 中村: 「コンピュータ・コスト」泥縄式管理」脱出の勧め, 日経コンピュータ, no.145, pp.71-82, 1987年4月。

[2] 本位田, 内平, 粕谷, 井原: MENDELによる計算機OS性能診断エキスパート・システム, 電気学会, システム・制御研究会, SC-86-15, (1986)

[3] 「EXCORE説明書」, AQA 10-3, 日本電気株式会社, (1987)

[4] 飯塚, 辻, 山中: 「コンピュータのシステム構成設計を支援するエキスパート・システム」, 日経エレクトロニクス, no.415, pp.163-183, 1987年2月。

[5] 「PERFORMS解説書」, DDP 71-1, 日本電気株式会社, (1985)

[6] 「運用管理説明書」, DDE 71-2, 日本電気株式会社, (1986)