

ケーススタディ - ACOS-4 性能解析エキスパート システム (EXPERFORM) -

麻生川 稔* , 白水 明** , 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 性能解析エキスパートシステム (EXPERFORM) の製品版を完成し、実際のユーザのシステム稼動状況データを解析したので、これについて報告する。EXPERFORM はコンピュータシステムの性能問題全般をカバーし、複合原因による問題も解析可能である。エキスパートシステムとしても、ルールブロックの実行制御、対象計算機の記述などにおいて、新しい手法を採用している。適用した結果によると、EXPERFORM は約3年間の経験を有するシステムエンジニアと同程度の解析能力を有すると評価され、実用に耐える十分な機能と性能を持つと判断された。

A Computer System Performance Analysis Expert System (*EXPERFORM*)

- A Case study -

Minoru ASOGAWA*, Akira SHIRAMIZU** and Hidehito KUBO*

* C&C Systems Research Laboratories, NEC Corp.

4-1-1, Miyazaki, Miyamae-ku, Kawasaki, Kanagawa 213 Japan

** Basic Software Development Division, NEC Corp.

1-10, Nisshincho, Fuchu, Tokyo 183 Japan

We have constructed an expert system called *EXPERFORM*, which can diagnose and propose counterplans for performance problems on NEC's ACOS-4 series. The performance problem is a case in which computer system cannot attain maximum performance due to insufficiency of hardware resources, inappropriate allocation of resources, and unsuitable job scheduling strategy.

Even though *EXPERFORM* is constructed with several rule-blocks, it can diagnose ranging over several rule block, by employing meta-rule block. A new mechanism on representation of diagnosed computer system is also employed.

EXPERFORM has been released and is evaluated in several users sites. And its performance is estimated as that of a three-year-experienced expert.

1. はじめに

近年、コンピュータシステムの大規模化・複雑化ともなっており、コンピュータシステムを効率的に運用することは非常に重要な課題となっている。効率的な運用とは、ハードウェア資源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。このために、業務量、レスポンスタイム、リソース使用状況などの膨大な稼動状況データを収集し評価することが必要である。しかし、コンピュータシステムにより一層の複雑化・多様化によりこの作業が非常に負担のかかるものになってきた〔中村〕〔本位田〕。

我々はこの状況を解決するために、性能解析の専門家達が持っている広範囲な解析知識をルールベース化したものと、知識（ルール）を系統的に組み合わせ判断するルールエンジンをを用いたエキスパートシステム化が有効であることを示し、ACOS-4性能解析エキスパートシステムのプロトタイプについて報告した〔麻生川a, 麻生川b〕。

本稿では、先のプロトタイプを元にして改良と拡張を加え、ACOS-4性能解析エキスパートシステム（EXPERFORM）をACOS-4ソフトウェアプロダクトとして第1版を完成したので、このシステムの概要と実際のケースに適用した結果について報告する。

EXPERFORM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稼動状況データを自動的に取り込むことができる。
- ②対話形式でユーザ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性能解析が実施できる。

EXPERFORMはエキスパートシステム構築用ツールであるEXCOREを用いて作成されている。EXCOREは、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）、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）、オブジェクト指向と知識ベース管理による高い拡張性、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）などの特徴を持っている〔日本電気a〕。

EXPERFORMの解析できる範囲は、コンピュータシステム側に起因する性能問題一般である。具体的には、サブ・システムについては、オ

ンラインシステム（VIS）、TSS、バッチシステム、個々のリソースとしては、CPU、メモリ、ディスク、MSP、ファイルを対象として扱っている。

EXPERFORMでは、これらの各々の解析に必要な専門家の知識を、診断される計算機に付随した知識と、それ以外の解析の知識とに分類して記述しているため、異なる機種に対する解析ができるように変更することも可能である。

計算機をオブジェクト指向で表現し、計算機に付随した知識は、オブジェクトのファセットとメソッドとして記述した。計算機に付随しない解析知識は解析する項目によっていくつかの領域に分割し、各々をプロダクションルールで記述し、ルールブロックとした。

性能解析を効果的に行うために専門家は、特定の項目に注目して解析しつつ、状況に応じて注目する項目を動的に変更する。このような動的な制御の知識は、専用のルールブロックにルールブロックの実行順序を制御するルールとして記述した。解析知識の各ルールブロックは他のルールブロックと協調して推論が可能なように、推論状態を総てのルールブロックで共通な解析状態表現形式で記述した。

実際のシステム稼動状況データをEXPERFORMで解析したところ、約3年間の経験を有するシステムエンジニアと同程度の解析能力を有すると評価され、実用に耐え得る十分な機能と性能を持つと判断された。

2. 性能解析エキスパートシステムの使用手順

EXPERFORMはすべてソフトウェアで構築されているので、エキスパートシステム構築ツールが実行可能な計算機で実行させることができる。このことによって、被診断計算機で性能を解析することも可能である。また、解析に必要な稼動状況データとシステム環境定義を移送することによって、他の計算機で解析することも可能である。

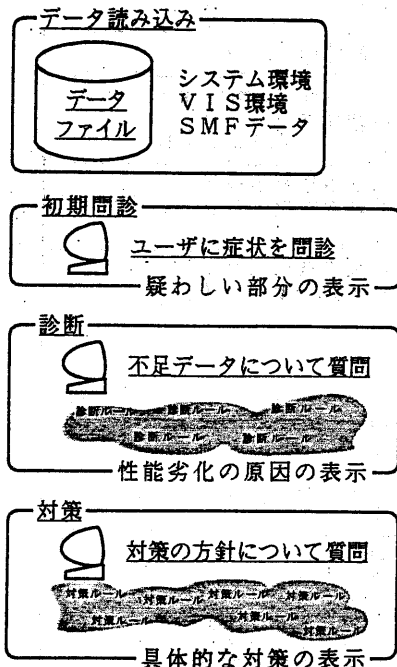
EXPERFORMの使用手順を図1に示す。

①稼動状況データの読み込み

パフォーマンス測定ツールで測定した計算機システムの稼動状況データと、システムの環境定義をEXPERFORMに設定する。

②初期問診

対象システムの性能に対して感じていることや、特に解析したい項目をEXPERFORMに対話形式で入力する。初期問診の結果として、性能劣化の原因と疑われる部分が表示される。



第1図 使用手順

診断の結果、性能劣化の原因である項目が表示される。

④対策

対策のおおまかな方針について入力する（運用を変更してもよい、もしくは運用を変更せずにリソース等の追加で解決する等）。また、対策についての方針について細かい要求を、対話形式で入力する。性能劣化の原因に対しての対策案が表示される。

3. 性能解析エキスパートシステムの構成

EXPERFORMは、コンピュータシステム側に起因する総ての性能問題に対して、実用に耐え得る機能を提供することを目的としている。コンピュータシステムの性能問題は、ユーザの立場より、システムの性能低下、特定のワークの性能低下の現象がある。これらの総ての性能問題について対処するための専門家の知識は膨大な量となり、例えばルール数ではプロトタイプで350であったが、EXPERFORMでは約1000となっている。またEXPERFORMの全体の規模として約26k行となった。

この様に知識が膨大となってくると、全体をいくつかの領域に分割してモジュールとし、各モジュール単位で管理する必要がある[住田]。

しかしながら、往々にして性能問題の原因は一つの領域にあるのではなくて、システム全体のバ

③診断

パフォーマンス測定ツールで測定されたデータを元にして診断が行われるが、測定できなかった項目についてのEXPERFORMの質問に対して対話形式で答える。ヘルプ機能があるので分からない質問に対しては、説明を求めることができる。

ランスが取れていないこと起因する。従って、実際の計算機システムを解析する時は、各領域の知識を同時に用いる必要がある。また、サブシステム(TSS等)のサービスが低下している時、リソース不足が原因である場合が少なくない。この場合では、サブシステムの性能解析とリソースの性能解析を同時に、行って初めて性能解析ができる[日本電気][IBM]。

従って、この様にモジュールとして分割された知識を必要に応じて同時に扱えるような知識表現が重要となる。また、分割した各モジュールを、いつ・どの様な場面で使用するかを表現することが必要となる(詳しくは、4. 解析知識の表現方法)。

コンピュータシステムの性能問題は、ユーザの立場より、システムの性能低下、特定のワークの性能低下の現象がある[日本電気c]。システムのパフォーマンスとしては、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、また特定のワークとしては、特定の種類の、VISトランザクション、TSSコマンド、バッチジョブがある。

これらの性能問題の原因は、

- ・リソースの不足、
- ・サブシステム等のパラメータが不適切、
- ・負荷の与え方が不適切

のいずれかである。負荷の与え方が不適切である問題については、サブシステム固有の問題と、リソース及びシステム全体のバランスの問題とに分けられる。

従って、性能解析の知識は、解析対象とするサブシステムの性能解析と、リソースの性能解析に分割することができる。さらにサブシステムとしては、オンラインデータベースシステム(VIS)、TSS、バッチシステムがあるので、これらのサブシステム単位に性能解析の知識を分割できる。(表1を参照)

これらのサブシステムでは、性能問題はサービス(スループット、レスポンスタイム)の低下として表れる。この現象の原因は、サブシステムのパラメータが計算機を取り巻く状況に対して不適切であることと、サブシステムに割り当てられているリソースが不足していることがある。従って、サブシステムの診断からリソースの診断を行うことができる様に必要がある。

計算機のリソースとしては、CPU、メモリ、外部記憶装置がある。表1中でメモリの項目は、仮想記憶管理システムの解析を含んでいる。外部記憶装置の項目は、ファイルのディスク装置間で

	項目	性能上注目すべき現象	性能問題の原因
サブ・システム	VIS (オンラインシステム)	VIS全体の応答が遅い 特定のトランザクションが遅い	VISが利用するリソースの割当量の不足 不適切なVISの動作環境の定義 アプリケーション・プログラムの動作
	TSS	エディタの応答が遅い LMの実行が遅い	全体のリソースの割当量が不足 不適切なTSSの動作環境(含むシステムパラメータ)
	バッチシステム	バッチジョブ全体の終了が遅い 特定のバッチジョブの終了が遅い バッチジョブの実行開始が待たされる	全体のリソースの割当量が不足 システムパラメータの定義が不適切
リソース	CPU	使用率(課金時間、非課金時間)	CPUへの過負荷(不適当な運用) メモリの絶対量不足 I/Oネック 特定のジョブによる占有 CPUの性能不足
	メモリ(含む仮想記憶システム)	多いミッシングページの発生回数	不適切なドメインバランス 特定のジョブによる占有 メモリの絶対量不足
	外部記憶装置	ディスクの使用率 待ち行列長 I/O回数とその種別 シーク距離 一回のI/O時間	不適切なファイルの物理的な配置 不適切なファイルの論理的な配置 ディスクのI/Oが過負荷 ディスクの絶対量不足
	ディスク処理装置	使用率の偏り 高い使用率	ディスクの配置の偏り MSPが全体的に過負荷
	ファイル	多いI/O回数	不適切なファイルの配置

表1 性能解析知識の一覧

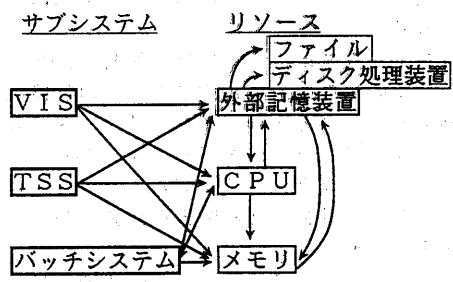


表2 呼び出し関係

の分散状況とディスク処理装置の解析を含んでいる。リソースに關係する性能問題としては、資源量の不足と不適当な割当てが原因で起こる過負荷がある。また、リソースの使用状況から、サブシステムのパラメータの設定が不適切であることを解析する必要がある。

以上のことを考慮して、各領域の知識の呼び出し關係は表2の様にした。この様に知識を構築す

ることによって、EXPERFORMではシステムのさまざまな領域に渡る複合された原因による性能問題も解析可能である。

4. 解析知識の表現方法

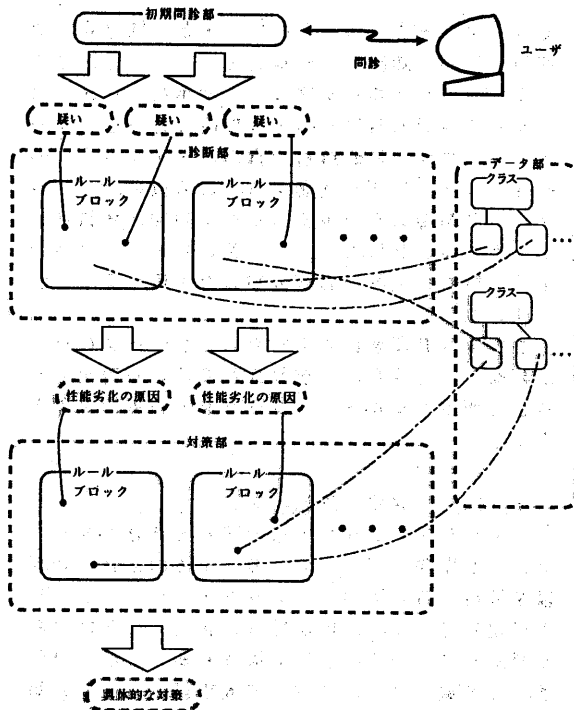
本章では、前回報告したプロトタイプ of 知識表現方法と、プロトタイプでは欠けていたためEXPERFORM追加した知識表現方法について述べる。

4.1 プロトタイプでの知識表現方法

プロトタイプは図2に示すように、①初期問診部、②性能問題診断部(診断部)、③性能問題対策部(対策部)、④稼動状況データ部、の4つの部分より構成した。

解析は以下の様に行われる。

- ①初期問診部が起動されて、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、その項目についての診断をすることができる診断部が起動される。



第2図 全体の構成

- 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 稼動状況データ部は、初期問診部、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稼動状況データとシステム環境定義データを供給する。これらのデータはパフォーマンス測定ツール等で収集され、あらかじめ取り込まれている。

性能解析の知識は、診断される計算機に対する知識、性能解析をする専門家の知識、解析中の状態に分けて表現した。

① 被診断計算機の表現

被診断計算機を構成する装置 (e.g. ディスク処理装置、ディスク装置) はオブジェクトとして表現することができる [飯塚]。例えば、ディスク装置は、性能規格を表現するクラス、稼動状況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、性能の規格値と、それら进行处理する手法 (メソッド) が定義されている。稼動状況データを表現するクラスでは、稼動状況データ値と、その値进行处理するメソッドが定義されている。このクラスをスーパークラスに持つクラスは、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、稼動状況データの処理を行うことができる。この様にオブジェクト指向の表現を用いることによって、被診断計算機は、オブジェクトの集まりとして表現できる。

② 性能解析の知識

性能解析の知識は、大きく分けて初期問診部、診断部、対策部に分けた。各部分は、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 IF~THEN 形式のプロダクションルールで表現されている。

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が把握し得る稼動状況データをもとにして対話形式で診断を進めてゆき、最終的に性能劣化の原因がどの部分にあるか見当をつける。

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を目的としているので、“疑わしきものは、総て疑う”という方針を取っている。初期問診部で扱う疑いの数は、10個程度である。

性能問題診断部

診断部は前向き推論で行っており、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している。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。診断部では、あらかじめ取り込まれた稼動状況データをもとにして解析を進める。ルールブロック中のルールは3つの種類に分類しルール化した。これによって、専門家の診断の知識が表現しやすくなった。

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
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
3. 性能劣化の原因を判定するルール

性能問題対策部

対策部は前向き推論で行っており、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している。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。ルールブロック中のルールは3つの種類に分類しルール化した。

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
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
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

③ 推論中の状態の表現。

ルールの推論中の中間結果を表現するのに述語表現を用いた。述語名としては、“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”の述語名を用いた。これらの述語名は、総てのルールブロックに共通である。解析中は、述語はワーキングメモリ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。これらの述語名は、総てのルールブロックで共通であるので、他のルールブロックで推論された結果を、述語を通して参照することが可能である。このようにして、複数のルールブロックで診断が可能となる。

4.2 EXPERFORMに追加された知識表現方法

今回報告するEXPERFORMは基本的にはプロトタイプの構成方法、知識表現方法を引き継いでいるが、さらに性能解析の知識を増強した。プロトタイプでは、ルールブロックをいつ・どの様な場面で用いるかに関する知識と、対象計算機に対する知識に関する記述能力が足りないことが分かった。ここでは、それらについて増強した知識表現法について述べる。

4.2.1 ルールブロックの実行制御

プロトタイプでは、ルール数は350であったが、本製品ではルール数は約1000となっている。この様に知識が膨大となってくると、前項でも述べたように、領域に分けてモジュール化する必要がある。モジュール化した知識を必要とされる時に使用するよう、ルールブロックの実行順序を制御することが必要となった。

EXPERFORMでは、性能解析の各領域の知識をルール化し、一まとめにしてルールブロック(性能解析知識のルールブロック)としている。

それらのルールブロックをいつ・どの様な場面で使用するかの知識をルール化して、一まとめにルールブロック(ルールブロック制御のルールブロック)を各領域毎に一つずつ作成した。このルールブロックは、性能解析知識のルールブロックの動作のタイミングを制御する。この様に、ルールブロック制御のルールブロックを知識の各領域に割り当てることによって、EXPERFORMの全体の解析の流れの制御の知識を表現することが可能となった。

また実行を高速化するために、ルールブロックが起動された場合に、起動されることが確実であると判断できるルールブロックを予め実行するようにした。

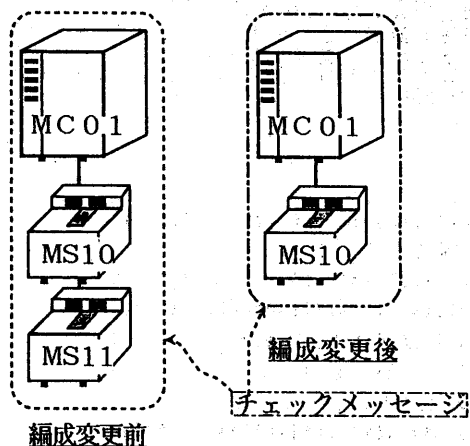
4.2.2 対象計算機の記述

EXPERFORMでは診断する計算機をオブジェクト指向で表現している。具体的には、計算機を構成する装置のそれぞれをオブジェクトとして表記している。解析を行うときに、計算機の構成要素をグループに分けて扱う必要がある。例えば、計算機システムにおいて仮想記憶システム関係のファイルを含むディスク装置と含まないディスク装置とでは、扱い方が異なる。従って、仮想記憶システム関係のファイルを含むディスク装置を表すオブジェクトと、ファイルを含まないオブジェクトでは継承するスロット・メソッドは異なる。この違いを、オブジェクトの一つの変数で管理する方法と、複数のオブジェクトを一つのオブジェクトにマッピングする方法とがある。EXPERFORMでは、以下の理由から後者の方法を採用した。

計算機の性能問題を解析する場合は、計算機の構成要素のグループ全般についてスロット・メソッドを考えることが多い。例えば、仮想記憶システム関係のファイルを含むディスク装置の平均I/O回数、平均I/O時間の計算などである。しかしながら、これらのスロット・メソッドはオブジェクトの集合であるグループについて考えられることであり、グループの各構成要素に対して考えられることではない。従って、このグループを一つのオブジェクトとして取り扱う方が都合がよい。

計算機の性能問題を解析した後は、対策案が計算機の再構成であることが少なくない。これらの対策案を採用することによって、計算機の状態が業務のサービスを保証しているかをチェックする必要がある。このときに、対策案を採用する前と、採用した後の状態を同じ様に扱うことができれば、そのチェックは容易にできる。計算機の構成をグループで表現している場合は、グループそのもの

がオブジェクトに対応しているため、取扱が容易である。(図3参照)



ディスク処理装置とディスク装置の1グループが1つのオブジェクトとして表現されているので、構成を変更したときのチェックは構成要素に関係なくオブジェクトで扱える。

第3図 グループのオブジェクト表現

5. 適用例

実際のユーザのシステム稼動状況データをEXPERFORMで解析した結果の具体的な一例の評価を以下に説明する。

○評価環境

- ACOS-630モデル10 (中型汎用計算機)
- ・マルチプロセッサ構成 (数MIPS)
- ・主記憶 48MB
- ・OS ACOS-4/MVP XE R3.1

○利用者が問題にしている症状

- ・TSSエディタの応答が遅い
- ・バッチジョブのスケジュール時にメモリ待ちの状態になる。

○性能問題の専門家の解析結果

- ・メモリ多重度の不足 (真の原因)
 - メモリ多重度を増やす
- ・各ドメインのサイズのバランスが悪い
 - バランスを調整
- ・特定のディスクにI/Oが集中している。
 - ファイルを分散

○性能問題の専門家が診断に必要なとした時間

(データ収集と解析)

- ・約14時間

○EXPERFORMの診断結果

- ・メモリ多重度不足 (真の原因)

→メモリ多重度を増やすように提案

- EXPERFORMが診断に要した時間
 - ・約20分

このケースでは、EXPERFORMは決定的な真の原因を見つけたために、その他の多少プライオリティが低い原因については言及していない。

別のケースでは、性能問題の専門家は主に定性的な判断 (症状に対する経験的判断) に基づき解析したのに対し、EXPERFORMは定性的な判断に加え、定量的な判断 (システムのさまざまなパラメータ数値の妥当性に対する判断) に基づいた解析も行った。

ある見方をすれば、このケースでの解析に於いては、性能問題の専門家に対し、EXPERFORMは常に定量的な判断も行うことから専門家の補助的な役割も果たすことができるということが分かった。

これらの結果から、EXPERFORMは実際に耐える十分な機能と性能を持つと判断できる。我々の専門家は、EXPERFORMは3年間程度の経験を持つシステムエンジニアとほぼ同程度の解析能力を示したと評価している。

6. 結び

EXPERFORMでは、項目毎の性能問題の解析が可能である。さらに、知識をいつ・どの様な場面で使用するかの戦略を記述しているので、複合原因による性能問題も解析可能である。

実際のシステム稼動状況データをEXPERFORMで解析したところ、約3年間の経験を有するシステムエンジニアと同程度の解析能力を有すると評価された。また、対話形式でユーザ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、性能問題の専門家でない人でも対話処理形式で簡単に性能解析が実施できることから、EXPERFORMは実際に耐える十分な機能と性能を持つと判断できる。

<<参考文献>>

[中村] 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)

[本位田] 本位田, 内平, 粕谷, 井原: MEND Eシによる計算機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)

[麻生川a] 麻生川, 白水, 伊久美, 久保: ACOS-4性能改善エキスパートシステムの開発, 情報処理学会, オペレーティング・システム研究会, 87-05-35-2, (1987)

[麻生川b] 麻生川, 久保: ACOS-4性能改善
 エキスパートシステムの構成, 情報処理学会第3
 5回全国大会講演論文集, 1L-5, (1987)
 [日本電気a] 「EXCORE説明書」, AQA 10-
 3, 日本電気株式会社, (1987)
 [住田] 住田, 村井, 大庭, 加藤, 中谷: 性能監
 視エキスパートシステム: EXPECT, 情報処
 理学会, オペレーティング・システム研究会, 88
 -05-38-6, (1988)
 [日本電気b] 「PERFORMS解説書」, DDP
 71-1, 日本電気株式会社, (1985)

[IBM] IBM: OS/VS2 MVS Per
 formance Notebook, GC28-0886
 -2, (1986)
 [日本電気c] 「運用管理説明書」, DDE 71-2, 日
 本電気株式会社, (1986)
 [飯塚] 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.

付録. 解析結果の例

```

*****
*          バッチジョブの実行についての診断をしています。          *
*****
*          バッチジョブの起動時に「DX J」及び「DJ」の画面で「W *
*          AIT M」になってしまうとのことですが、そのバッチジョブ *
*          が実行されているドメイン名をBATDの様に入力してください。 *
*****
>>BATD
*****
*          性能劣化の原因を以下に表示します。          *
*****
*          診断の結果、バッチジョブの起動時に「DX J」及び「DJ *
*          」の画面で「WAIT M」になってしまうことの原因はその *
*          バッチジョブが実行されるドメインのメモリ多重度に抑えられ *
*          たためであることが分かりました。 *
*****
*****
*          対策を以下に表示します。          *
*****
*          バッチジョブの起動時に「DX J」及び「DJ」の画面で「 *
*          WAIT M」になってしまうことの原因はその性能劣化の原因はそのバ *
*          ッチジョブが実行されるドメインのメモリ多重度に抑えられた *
*          ためであることが分かりました。 *
*          この対策として、システムパラメータの辛DRMPARM文の *
*          BATDの主記憶多重度長期枠最大値(MPLX)を最低でも30以 *
*          上に指定し直してください。 *
*****

```