

UNIX 4.2, 4.3 BSD 版のトラブルシューティング用 エキスパートシステム

藤 卷 みどり*
渡 辺 善 道*
今 宮 淳 美*
坂 本 忠 明*

(昭和63年8月31日受理)

Trouble Shooting Expert System in UNIX 4.2, 4.3 BSD Machines

by Midori FUJIMAKI*, Yoshimitsu WATANABE*,
Atsumi IMAMIYA* and Tadaaki SAKAMOTO*

Abstract

We are now developing a knowledge-based user interface and system.

This paper describes the design and the mechanism for trouble shooting expert system in UNIX 4.2, 4.3 BSD machines. We used the translator writing system SOUGEN and C-program for expert system.

概 要

近年の OS の水準として高位にある UNIX。そのバークレイ (BSD) 版は、当科で稼動しているスーパーミニコンピュータの VAX-11/785, VAX-11/730, ワークステーションである SUN3, Micro VAX I, II, NEWS, さらに図形画像処理システム室での VAX-11/750 で動作している。また、これらの計算機は、ソフトウェア、ハードウェア資源の共有等の高度利用法として、LAN (Local Area Network) で結合されている。

この共通な OS の環境上、おのおの管理運用 (維持管理体制)¹⁾ する上で、障害状況の把握や回避、復旧等おのおのの計算機に共通した要素をもっている。そのため、障害に関する情報を維持管理支援スタッフに共通に提供する点での、当エキスパートシステムを試作した。しかし、障害の捕らえ方や、条件の定義による機能形態、状況の提供方法等によりエキスパートシステムの構成、機能が異なる。本試作では、知識データベースに記録されていない情報を新出情報とし、すでに

登録されている情報を既出情報と分け、新出情報での不足情報の予測機能と、既出情報の検索を考察したエキスパートシステム化を図った。

1. はじめに

当システムの試作には、1986年4月より納入された VAX-11/785, Micro VAX I, II, およびすでに他 OS で稼動している VAX-11/730, 図形画像処理システム室の VAX-11/750 のおのおのを、当時入手した UNIX 4.2 BSD 版で動作²⁾させ、最新バージョン 4.3 BSD 版に移行した時点での様々な障害を整理することから始める。整理内容は、障害発見および修復経緯、障害における計算機の状況、その他付随する情報 (操作条件や操作性) である。つぎに、これらの情報を有効に利用するために、如何に活用するかの点で、障害の早期発見と対策、障害内容の把握の2点での条件を集約した。

2. 計算機環境

先にも示した当科および図形画像処理システムの VAX ネットワークは、図-1 に示されるイーサネット (TCPIP) で結合されている。さらに、このネットワ

* 計算機科学科, Computer Science Department.

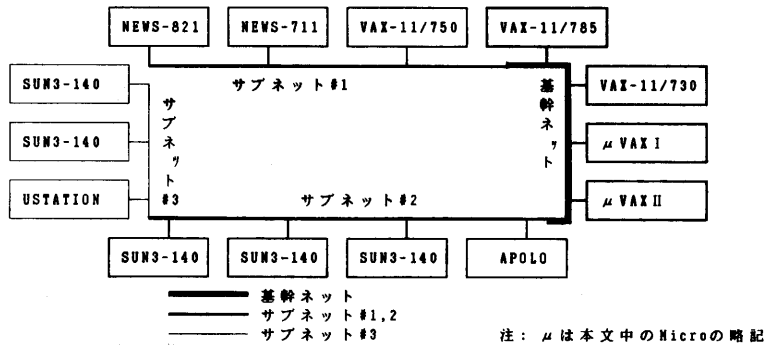


図-1 計算機科学科のコンピュータネットワーク構成 (イーサネット)

ークを有機的に機能させるため、OS を UNIX のパークレイ版を基に構成されている。これらのネットワークは、大容量、高速度処理計算機 VAX-11/785 を包含している基幹ネットワークと、それ以外の計算機からなるサブネットワークで構成されている。

当エキスパートシステムは、この基幹ネットワーク上の VAX シリーズの計算機を対象とした。各サブネットワーク上の計算機環境は、それを保持、運用している研究室が主に管理する点で、トラブル情報の整理された報告例が極めて少ないために除外した。

3. 障害内容の整理

障害内容を整理する上で 2 段階のレベルで分類することを念頭においた。第 1 段階は致命的な障害レベルで、回復作業には、即時的に行うべき障害群である。第 2 段階は警告レベルで、授業等に影響しない時間帯に修復すべき障害群である。なお、利用者の操作ミス、外部からの進入による障害²⁾は、ハードウェアが介入する点以外は除外した。

3.1 整理項目 1

障害に関するメッセージと障害との対応、他に波及する内容、これらを修復手順とその修復段階による一時的修復、恒久的修復の視点からの整理を行う。この障害に関するメッセージは、特にシステムコンソール上に表現されたエラー情報である (UNIX は、障害が発生した場合のエラー情報は、システムコンソールに打ち出されるか、メッセージファイルに書き出される)。このエラー情報は、ソフトウェアエラーと、ハードウェアエラーの 2 種がある。しかし多くの場合、ハードウェアエラーが生じた場合、ソフトウェアエラーを伴い、逆にソフトウェアエラーが生じた場合も遅かれ早かれハードウェアエラーが生じる。ただし、片

方のエラー情報のみで他が表示されない場合も多少だがある。さらに何も表示されない場合もある。こういった点でのエラー情報と障害との対応がとれず (完全な対応が取れないので条件が組み込めない)、エキスパートシステム上の知識データベースに登録できない条件がある。これには、技術変更時の情報や junet 上での関連データにより予測できる可能性をシステムに組み入れている。

エラー情報で注意しなければならないのは、提示されたエラー情報が必ずしも正しいデータではない場合もあることで、2 次的なエラーが原因となる。これは、カーネルが呼び出したツールの内部で他のツールを使用している時に生じたエラーメッセージである。

波及する内容については、LAN でつながっている計算機システム構成上、他の計算機にも波及する可能性と、障害が起きた計算機内部の資源に影響を及ぼす可能性についての整理項目である。他に通常ビールスと呼ばれる、一度動作したら計算機上の資源を全て食いつぶすまで止まらない等といったソフトウェアの進入等の破壊する事のみを目的とするソフトウェアについては、現状では考慮していない。

修復手順については、その機能を使わせないようにするか、エミュレート機能を用いるかしてしばらくの間通常運転に踏み切る「一時的修復手順」と、装置、インタフェースの交換により恒久的に使用できるようにする「恒久的修復手順」とに分けて整理する。

3.2 整理項目 2

他の整理項目として、早期発見、システムコンソールおよびメッセージファイルに記録されない障害を発見する場合、障害発生の予測等を行うための整理内容である。これには、定期的に行った (得た) 情報と不定期的に行うものも 2 種類がある。定期的に行った結

果得る情報については、計算機上のメッセージファイルの点検によるエラー情報以外の情報と、junet上の情報、それにシステム立ち上げ時等のシステムコンソール上の情報である。また、不定期的に行った結果得る情報については、メーカーからの技術変更に伴う情報がある。これは、ハードウェア的に何を変えたのか、それはなぜか、他の動作している状況はどうなのかについて整理する。また、ソフトウェア的にその変更箇所を使用しているものは何なのか整理する。さらに、UNIXに関する変更として、カーネルに対する変更には、カーネルの位置と内容、機能、利用しているハードウェアの内容と利用状況の整理、ツールの変更には、機能と利用ハードウェアの記述、当該ツールと他との共合の有無である。

3.3 整理項目3

以上の整理内容³⁾より、有効利用を基礎にしたエキスパートシステム化を第1目標としたため、情報提供を如何にするか、もしくはすべきかを第2目標とする。この第2目標を達成するためには、エキスパートシステムへの情報提供内容と、エキスパートシステムからの情報提供内容の2点より整理を行う³⁾。

4. システム構成と機能

エキスパートシステムは3種類あり、それを構築するためのツールをC言語で記述してある「想玄」⁴⁾および、C言語による補足機能のプログラム化を用いて、知識ベース部、推論部、ユーザインタフェース部を構成する。各エキスパートシステムの情報は、内部保持情報および利用者への提供情報、応答情報、おのおの漢字混じりの日本語を対象にした。なお、当システムは、PC9801上に実現する。これは、日本語情報の登録や、利用者からの入出力が熟語レベルでの機能性の高いフロントエンドプロセッサATOK5を利用することができる点と、図情報を登録するために、ワープロ機能が使える利点を持っている。反面、記憶量、検索速度、パターン照合等の量的、速度的、精度的要素は、ミニコン、ワークステーション類には劣るが、当システムでの要求範囲内を満たしている点、ターゲットマシンに選定した。

機能として、2種のエキスパートシステムおのおのに対し、ユーザインタフェース部、判定部、ルール部、推論部、知識ベース部から構成されている。ユーザインタフェースは、利用者の要求に沿うよう入出力を行う機能である。また、入力情報の一部を輪切りにし、その一部のパターン照合をするグループ化処理が判定部の機能で、IF、THENルールにより知識を獲得、分

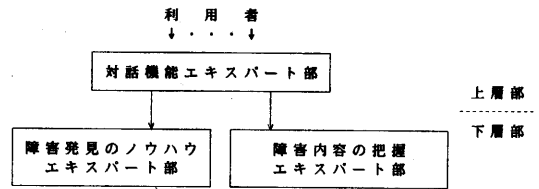


図-2 システム概念図

析、利用するのがルール部の機能である。さらにこれらを有効に利用するための機能が推論、および知識ベース部である。

3種のエキスパートシステムの構成は、図-2に示す2階層構造により、より利用し易く結合される。

4.1 構成

システム全体は、図-2の3種のエキスパートシステムにより有機的に結合されている。しかし、対話機能エキスパート部は、利用者からの要求が何であるかを知り、必要とするエキスパート部を呼び出す「分配機能」と、HELP的機能を持った「解説機能」、それに利用者情報から利用者へ提供すべき情報を分けて表示する「利用者表現機能」を有するが、当システムではまだ完成に至っていない。これは、分配機能には判断部として、この機能の最下層に1ルールを用いることで実現することができるが、最終の到達目標である、自然言語による利用者の意図を把握し、この最下層のルールに到達させるかの機能設計が不完全な点と、利用者情報をどの様に定義するかの詳細設計が未決定な点からである。

他のエキスパートシステムは、上層の対話機能エキスパートシステムを起動せずとも動作するよう知識ベース部、推論部、ユーザインタフェース部を上層から分離させた。この上層部と下層部の独立から、上層の対話機能エキスパートの完成を見て動作チェックを行う総合性を図る開発段階は、次の段階におくことができる。

下層2種のエキスパートシステムは、共通な情報を保持する部分が多大にある³⁾。そこで限られた資源での有効利用は、共有できるものは極力共有化を図る点である。この共有化については、ルールベースのエキスパートシステムである当システムのルール部のみを分離し、他を共通に持つことで実現する。つまり、2つのエキスパートシステムのルール部が異なり、他の知識データベースには同じ内容が格納されている。

4.2 機能

(1) ユーザインタフェースとルール部

利用者の要求をエキスパートシステムに繋げ、対話

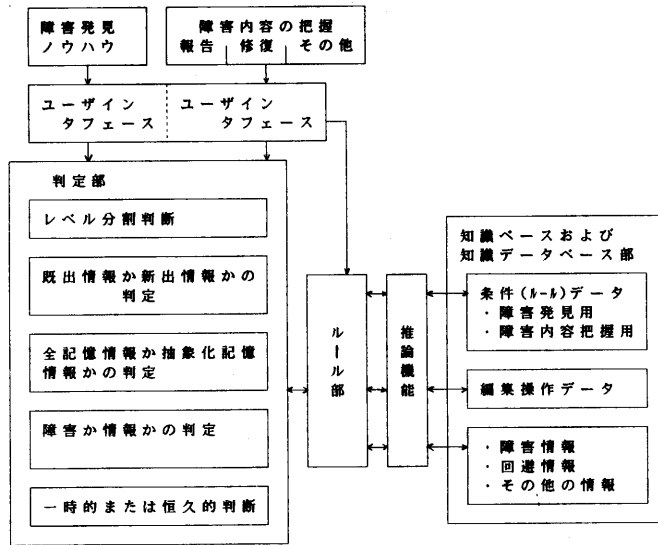


図-3 システム機能図

しながら目的を達する機能である。つまり各エキスパートシステムの必要とする情報を利用者から吸収する機能と、利用者に提供する機能、連続した情報を分割する機能⁵⁾の3種がユーザインタフェースである。

ルールには主ルールとサブルールがある。主ルールの内容は、障害の発見のノウハウには、メッセージファイル若しくはシステムコンソールに打ち出されたルールと、VAXの技術変更にもなるルール(ハードウェア的、ソフトウェア的影響点のルール)、UNIXに関する変更のルール(カーネルの変更、ツールの変更ルール)の3グループを用意している。サブルールに関しては、主ルールから到達した目標に対し、変更等の入力用ルールを用意している。なお、情報の修正について、現情報をひとまとめに、その情報と追加情報の論理演算の結果を新しい情報と置き換えることができる。

① 「障害発見のノウハウ」エキスパートシステム(主ルール数は19個)

障害発見のノウハウは、障害を見つけ出す目的と、障害があった場合、その情報を与えることで障害箇所を当システムが整理し、障害箇所の予測をする目的を持つ。ここでは、システムコンソール上に出力された障害情報を、出力されたそのままのメッセージで与えることによりシステムが整理、格納し、サブルールを適用して分かりやすい情報提供を行う例題を掲げた。

```

<主ルール>
IF   メッセージファイル   または   システムコンソール上に注意すべき情報がある

```

```

THEN
    メッセージを入力せよ。

```

```

情報入力機能
→ Jun 9 17:46:32 yuimage vmunix: udao: hard error Jun 9 17:46:32 yuimage vmunix: udao: hard error, disk transfer error, unit 0, grp Ox 0, hdr Ox 17921, event 0110

```

```

分割機能
日付: Jun 9 17:46:32
システム名: yuimage
エラー装置名: udao
エラー内容: hard error
           : disk transfer error
エラー位置: unit0 grp Ox0 hdr Ox17921
エラー符号: event 0110

```

```

<サブルール>
IF   エラー内容   または   位置   または   符号
      について修正すべき情報がある
THEN
    致命的エラーの発生: →

```

エラー内容：ハードウェア障害：→
 : ディスク装置変換時の障害：→
 エラー位置：ヘッド0のグループ番地=0x0,
 ハードウェア番地=0x17921
 エラー符号：主記憶上の書き込み時の障害
 : → +パリティエラー

② 「障害内容の把握」エキスパートシステム

このシステムは、報告機能と修復方法の提示、その他として障害情報の追加修正および付帯情報の記録機能を持つ。報告機能は、今までに生じた障害の記録を時系列を追って報告する機能で、情報検索機能が主体である。修復方法については、障害情報を頼りに障害箇所の発見と、対処方法の提示、作業時間の予測を行う。障害情報の追加については、障害(エラー)条件と対処方法の対応を取るための編集機能を持つ。この例題として修復方法の提示を行う。

注：通常 IF, THEN ルールは、以下のように記述されないが、ここでは分かりやすさを基調とする点、このように記述した。

<主ルール> (ディスク上のバッドブロックにおいて)

```
IF 致命的エラーに含まれている障害であり かつ 一時的修復
THEN
  以下のスケジュールで修復せよ
  IF しばらく動作させる必要がある
  THEN
    IF 当該プログラムが、障害対象部に影響しない
    THEN
      障害対象部のパーティションのみ UMAUNT し、行いたいプログラムを動作
      ・ /etc/mstab のファイル内容からパーティション情報を見る。
      ・ 障害パーティション内でプログラムが影響するか
      ・ プログラムの位置を参照せよ
    IF 障害パーティション内のプログラムまたは ファイルを利用しない
    THEN
      /etc 上の unaunt 機能を動作させ、umaunt する。
      プログラム実行後、早急に恒久的修復作業に入りなさい。
```

(2) 判定部

判定部には、警告レベルと致命的レベルの判断を行う機能がある。これは、障害が同一エラー情報の一部が発見されたからといって、その前後の状況により警告レベルにも、致命的レベルにもなり得る場合が多い。先の②の例題に示したディスク障害に対しても、同一パーティション内に影響のあるプログラムであったなら、これは致命的障害である。そういった状況下の判断を予測するため、グループ番地の内容と、ハードウェア番地の内容から、対応するプログラムの位置関係を把握し、判断する機能である。

さらに判定部には、記憶する情報の最適化を判断するために行う機能をも要求する。つまり、記憶していく情報は、共通な情報が多数存在し、すべてに対しそのまま記憶したのでは、記憶容量が爆発的に増えてしまう。そこで、すでに記憶された内容ならネットワーク情報を持たせることにより記憶の節約を行う機能で、この記憶情報が既に記憶されたものなのかの判定部と、その判定された内容全てに対し記憶すべき情報か、それとも部分的に抽出若しくは抽象化し、大ざっぱな情報として保持する。これより、すでに記録された既出情報であるか、それともまったく新しい新出情報であるかの判定機能により、整理して利用者に提供する機能となる。これには、抽象化した情報としての編集情報を持つことにより実現する。この編集データを自動的に生成するのは、現システムでは不可能である。そこで、当初自然言語データを記述させ、次に分類項目に沿い記入させる方法を取った。この結果、以下に例示した期待する編集データが求められなかった。その例を示す。

<MTの障害情報>

: → MT を読み込み中、3箇所データが壊れてファイルに書き出された。

<利用者の登録情報>	<期待する編集データ>
装置名：MT	装置名：MT, ディスク
動作：→読み込み	動作：読み込み, 書き出し, 逐次
: →書き出し	: →書き出し
結果：→3箇所データ	結果：15570/3ブロックごとにデータ破壊
が破壊	破壊
MTの大きさ：15570ブ	他の情報：エラーメッセージの有無
ロック	無

他の情報：→
これを避けるため、障害情報の文章を日本語処理フ

ロントエンドプロセッサ ATOK5 に向け、文節、熟語ごとに分け、利用者の介在による編集データ作成機能をもつ。これより利用者は、システム内で分類されている情報一つ一つに対応、照合させ、当該障害データに埋め込むことにより、日本語として連続的に記録させ、内部で分類される。この例では、以下となる。

〈編集データ作成機能〉

MT を 読み込み 中 3箇所 データが 壊れて
ファイルに 書き 出された

装置名：→MT, ディスク (ファイル)

動作：→読み込み, 書き出し：(逐時的か一括か)：→
逐次

結果：→3箇所データ破壊：(大きさは)：→15570ブ
ロック

他の情報：→メッセージはでない：(システムコンソ
ール上には)：→何も記述されない

〈編集データ〉

装置名：MT, ディスク

動作：読み込み, 書き出し, 逐次

結果：15570/3 ブロックごとにデータ破壊

他の情報：エラー情報なし

障害か情報かの判定は、技術変更や UNIX に関する変更により登録された情報が障害情報として記録され、エラー情報と同位で対象とされるのを防ぐために行う。逆に、障害が起こる可能性があるために記録する情報という前提上、障害情報として意味を成す情報として扱う。当システムでは、「創玄」上の確信度⁴⁾を利用し、これを可能としている。

(3) 推論部

全ての推論には「後向き推論」を利用している。これは、求めるべき最終ゴールを事象変数（つまり確信度）を利用してその候補となった各ゴールの評価を行うのが当エキスパートシステムである。この事象変数の決定には、ルール番号の小さい順に候補とし、ルールの条件が成立するかの判定を行う。この結果対象となるルールの場合、確信度計算に入り、次のルールを候補として順次計算する。ここで注意しなければならない点は、自己参照の場合で、これを許している。例えば、

IF ディスクが故障している

かつ ディスク障害メッセージがでている

THEN ディスクが故障している

・・・(付帯情報)・・・

が許される。

これは、多くのエキスパートシステムでは、結論を先に提示し、様々な情報を与えた場合、先に示した結論以外の結論に到達する⁶⁾。つまり、矛盾を生じる場合がある。ここでは、結論が先に利用者から与えられても、戸惑うことなくその結論に行き着くことができ、矛盾が生じることをおさえている。

5. おわりに

当システムは、試作段階の關係上多くの設計仕様の変更を余儀なくされた。その結果、未完成部分も多々あるが、部分的には完成度の高いものである。反面、全体をひとまとめにすると機能的には、あまり良いものとは言えない。この点について、エキスパートシステムを構築するツールである創玄の記憶容量不足より、結論部（到達ゴール）が爆発的に増大し有機的な結合（共有化）が望まれる点である。

また、システム全体が情報を入力するルールベースに主眼が行き、報告等の利用面でのエキスパート化として弱点がある。これには、この側面でのルール数を増すことにより今後均衡の取れたシステムになるよう思われる。

参考文献

- 1) 藤巻, 他: UNIX 4.2 BSD 版システムジェネレーションと管理運用のノウハウ, 計算機管理室報告集, 1986. 11
- 2) Alan Filipski: UNIX の防衛能力, 日経 BYTE, 1986. 6
- 3) 藤巻: 障害情報の整理 1, 2, 計算機管理室報告集, 1988. 4
- 4) A.I. SoFT 社: 創玄操作マニュアル, A.I. SOFT, 1986. 6
- 5) 住田, 他: 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・システム研究会報告, 88-OS-38, 38-6, 1988. 2
- 6) 榎本 肇編: ソフトウェア工学ハンドブック, オーム社, pp. 223~258, 1986. 6
- 7) 宇津宮, 清水: 分散型個人用計算機環境のための UNIX の自然な拡張の試み, 情報処理学会論文誌, Vol. 27, No. 12, pp. 1209~1217, 1986. 11