

# FA分野における知識処理システム構築用ソフトウェア“EUREKA”

## Software for Building Knowledge Based Systems in Factory Automation “EUREKA”

熟練技術者のもつプロセス運転・診断ノウハウを入力することにより、人間と同水準の高度な判断処理を計算機に代行させることを可能とする知識処理システム構築用ソフトウェア“EUREKA”を開発した。本ソフトウェアは、(1)プロセスの機器状態、機能に関する知識、状況に応じて何をなすべきかという操業判断知識などを自然な形で与えることができ、更にこれらの知識を多様に利用できる(前向き、後向き推論、あいまい性処理)こと、(2)リアルタイム環境下での使用に耐えられるよう推論処理の高速化を図っていることに特徴がある。制御用計算機HIDIC Vシリーズ、エンジニアリングワークステーションESシリーズに搭載して、運転、診断のための知識処理システムを実現し、その効果性を確認している。

船橋 誠 壽\* *Motohisa Funabashi*  
 増位 庄 一\* *Shōichi Masui*  
 明石 吉 三\*\* *Kichizō Akashi*  
 篠本 学\*\*\* *Manabu Shinomoto*  
 長谷川 邦 夫\*\*\*\* *Kunio Hasegawa*

### 1 緒 言

「知価革命」<sup>1)</sup>という言葉に代表されるように、知識情報の重要度が極めて増大し、外部情報の的確な選別収集、個々人のもつ情報の表象化による共有化などが急務となっている。更に、科学技術のパラダイムも、従来の分析的な物の考え方であるアトミズムの限界を打破するものとして、問題を総合的にとらえるホロニズムの重要性が認識され、取り扱うべき情報も数値的なものから記号的なものへと拡大している。これらを背景とし、人工知能科学及び計算機技術の進展に支えられ、知識処理システムの実用化が急速に始まろうとしている。

日立製作所では、このような知識処理システムを構築するための核ソフトウェアとして、EUREKA(Electronic Understanding and Reasoning by Knowledge Activation)を開発した。本稿では、FA(ファクトリーオートメーション)分野での知識処理ニーズを概観するとともに、これにこたえるべく開発したEUREKAの機能概要、応用事例について紹介する。

### 2 FAにおける知識処理

#### 2.1 知識処理システムの特徴

知識処理システムのねらいは「対象とする問題領域の専門知識やノウハウを利用して、専門家でなければ解けない問題を、専門家と同等の能力で判断し、制御や診断を支援・代行する。」ことにある。なぜ計算機システムでこのようなことが可能になったか、従来の処理方法と知識処理とを比較することによって明らかにし、その特徴を述べる。

図1は、従来処理と知識処理との特徴的な相違を模式的に表わしたものである。従来の手続き処理の方法では、知識プログラムの中に散在し、これらがまとまってはじめて機能できる。これに対して知識処理では、個々には独立した知識を与えておくだけでよく、要求に応じて推論エンジンがこれを結合して解を出力する。このため、

- (1) 個々の知識の独立性が保たれ、知識の追加、修正、削除が極めて容易になる。
- (2) 個々の知識の記述の順序が処理に影響せず、人間のもつ

並列处理的な思考パターンにマッチし、人間と計算機との親和性<sup>2)</sup>が高まる。

という特徴がある。人間の思考パターンは極めて複雑なため、従来のアプローチでは、思考判断過程を模擬するプログラムを作ることが難しかったが、知識処理という新たなプログラミングスタイルを生み出すことにより、専門家を代行するシステムが容易に実現できるようになったと言える。

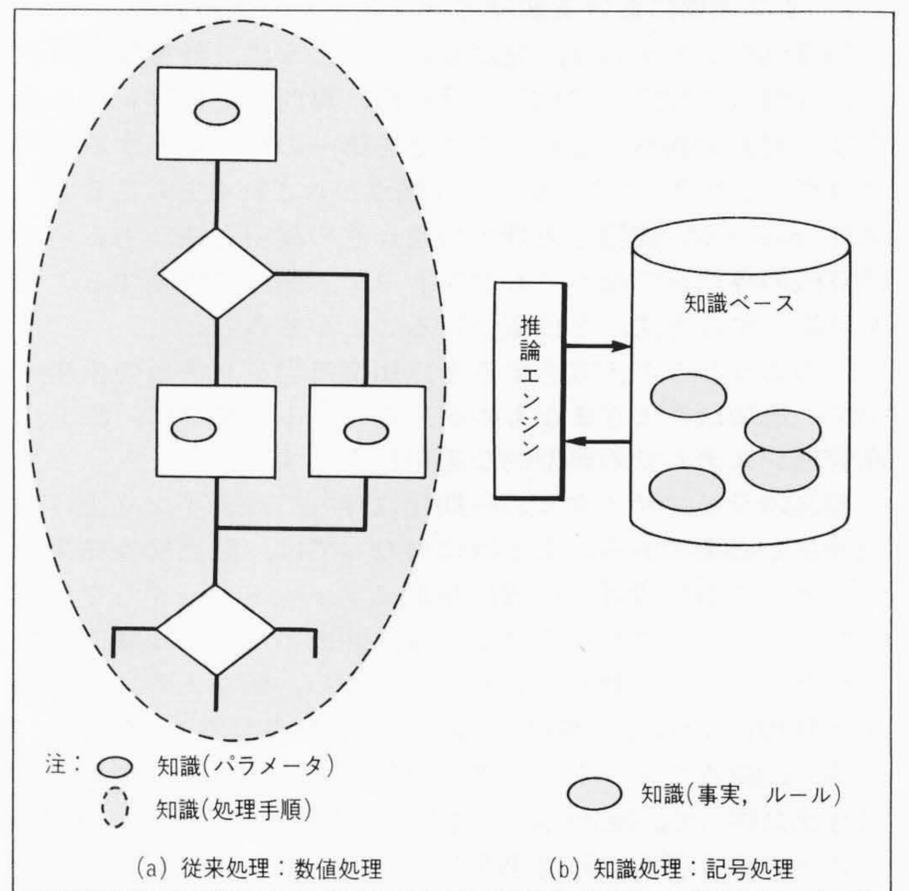
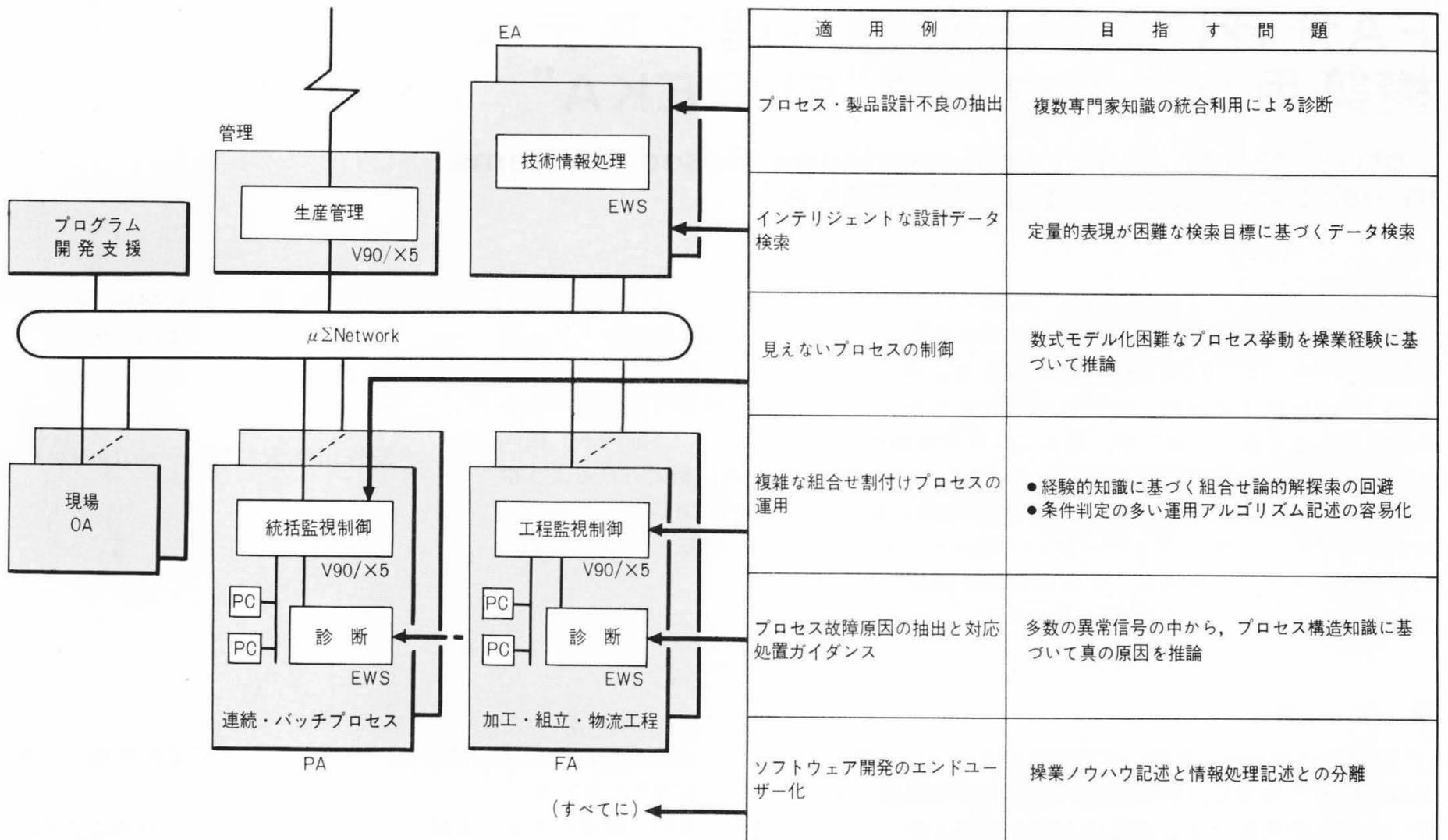


図1 従来処理と知識処理との比較 従来の処理方法では手順、パラメータという形で知識が表わされるのに対し、知識処理では手順を明示的に記述する必要がない。このために、個々の知識の独立化が図られ、プログラムが柔軟化する。

\* 日立製作所システム開発研究所 \*\* 日立製作所システム開発研究所 工学博士 \*\*\* 日立製作所大みか工場 \*\*\*\* 日立製作所機電事業本部



注：略語説明 EA(Engineering Automation), PA(Process Automation), FA(Factory Automation), EWS(Engineering Work Station), PC(Programmable Controller)

図2 総合生産管理システムにおける知識処理の適用例 知識処理は、従来では困難であったさまざまな業務の計算機化を促進すると同時に、ソフトウェア開発のエンドユーザー化にも貢献する。

### 2.2 生産活動における知識処理システムの適用分野

知識処理システムは、先にも述べたように、計算機システムに専門家の知識を移植し、計算機に専門家と同等レベルの高度な判断業務を行なわせることを第一のねらいとするが、プログラミングスタイルという観点からこれをとらえると、条件分岐の多い複雑な処理への適合性の高い方法、あるいは計算機の専門家でなくてもソフトウェア開発に参画することを可能とする方法、と位置づけることができる。

このようなとらえ方をすると、知識処理システムの生産の場での適用はさまざまなものが想定される。図2に、総合生産管理システムでの典型的な適用例を示す。

製品のライフサイクルの短期化に伴って、ライン立上げの迅速化が必要である。立上げに当たっては、製品検査結果をプロセスや製品設計の改善に早期にフィードバックしなければならないが、これらの生産・設計診断のための知識は複数の専門家の間に分散していることが多い。個々人の診断知識を計算機化すれば、分散した知識が統合化されることになり、迅速、的確なフィードバックが可能となる。また、製品の多品種化に伴って、設計効率を向上してゆかねばならない。CADの普及により過去の設計事例はデータベース化されていることが多く、設計事例を再利用することによってこの課題を解決するというアプローチが考えられる。このためのデータベース検索、更に設計合成は、従来の方法では不十分で、専門家の知識をうまく活用してゆくことが必要であり、これを計算機化することによって設計のスループットを向上させることができる。

生産現場でも、さまざまな適用が考えられる。熟練オペレ

ータの操業経験を計算機化することにより、従来では数式モデル化困難なため制御理論の適用が難しかった複雑なプロセスの運転自動化が可能となる。また、複雑な組合せ、割付けプロセスの運用では、運用アルゴリズムの計算機プログラム記述が極めて困難であり、更にプログラム保守も容易ではなかったが、知識処理を導入すれば、モジュール性の高いアルゴリズム記述が可能となり、従来の問題を解決できる。プロセス異常処理は、運転自動化の最大の障壁であったが、冷静な状況下で組み上げた専門家の異常処理理論を緊急時に走らせることができるため、的確な対応、二次災害への突入防止などを図ることができる。

これらの例は実問題に適用され、既に効果も確認されているが、4章では代表的な事例について詳述する。

### 3 知識処理システム構築用ソフトウェアEUREKA

EUREKAは、推論エンジンと対象問題に関する知識の容器である知識ベースから成り、対象問題に関する知識を与えることによって、個々の問題に応じた知識処理システムとして機能するはん(汎)用的なソフトウェアである。このようなソフトウェアの有用性を決定する主要因は、次の二つである。

#### (1) 知識表現及び推論エンジンの機能性

どのような形で知識を核ソフトウェアに与えることができ、また利用することができるかという使い勝手の水準である。

#### (2) 推論処理実行速度

知識処理では、パターン、マッチングを基本として処理が進行する。このため、処理負荷が必然的に増大する。実用的な応答性が得られるかという性能面での水準である。

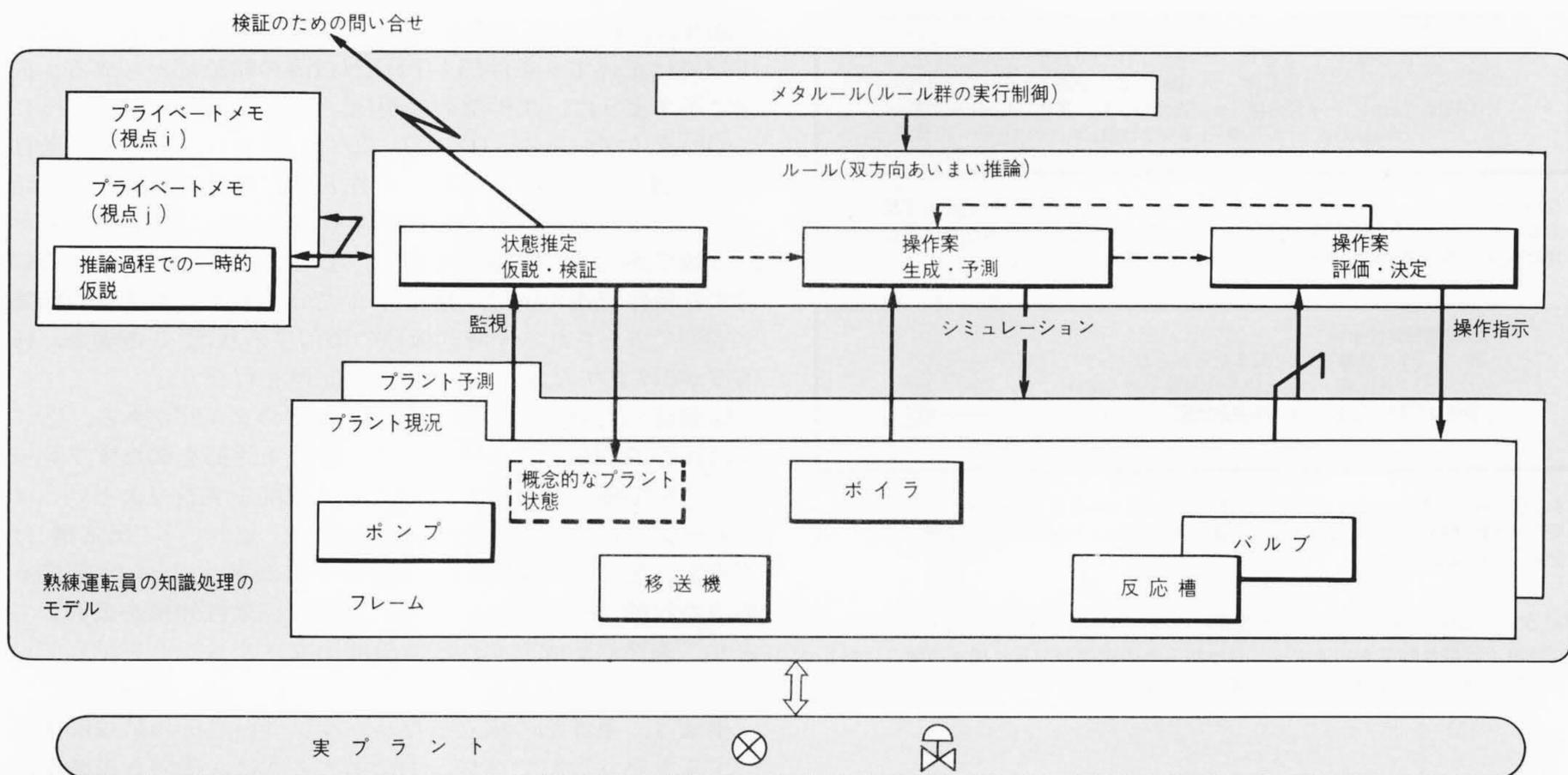


図3 プラント運転における知識処理モデル プラント運転に関する知識のコンポーネントは、メタルール、ルール、フレーム、プライベートメモによって表わされる。EUREKAはそれぞれに対応した知識表現方法を提供する。

本章では、上記の観点からEUREKAの機能、特徴について概説する。

### 3.1 知識のコンポーネント

FA分野で、どのような知識の種類があるかをプラント運転を例に模式的に表わしたのが図3である(それぞれの知識をここではコンポーネントと呼ぶ)。同図は、四つの知識のコンポーネントが必要であることを示している。

#### (1) フレーム

プラントの機器・構成など物理的な実体に関する知識。これらは、もしある操作を行なったらどうなるかといった予測結果も併行して考慮する必要があり、多重性をもたせることが必要である。

#### (2) プライベートメモ

推論の過程ではさまざまな一時的仮説、例えば、「反応が不安定状況に入りそうである。」といった仮説が生まれる。これらの知識は、フレームの一部とみなせないこともないが、物理的な実体に関する知識と比較すると未構造なものであること、推論の進行に伴って段階的に生成されることなどから別個に扱うべき知識である。

#### (3) ルール

「温度分布にばらつきがあれば反応は不安定である。」「レベルが基準値を超えればバルブを開け。」といった規則を表わす知識である。前者は因果性に関する知識であり、後者は操作に関する知識である。因果性に関する知識は、「風が吹けばおけ(桶)屋がもうかる。」といった三段論法的な推論(前向き推論)と同時に、「桶屋がもうかる」は正しいかどうかを確認するために、「風が吹く」の成否を調べる後向き推論にも用いられるものである。

#### (4) メタルール

知識の使い方に関する知識である。図3に示すように、操作案の決定に当たっては、プラントの状態の推定、操作案の生成、評価といった知識を逐次利用することが必要であり、これを制御するものである。

ここでは、プラントの運転に関する知識を事例として、4種の知識のコンポーネントが必要であることを示したが、このほかのさまざまな問題についても知識はこの四つに帰着することができる。EUREKAは、これら4種類の知識のコンポーネントの組合せによって、問題に関する知識を表現するようにしている。

### 3.2 知識のコンポーネントの記述とその利用方法

EUREKAでの知識の記述は四つのコンポーネントを用いるが、その中心をなすのはフレームとルールである<sup>3)</sup>。

プラントの機器状態、機能などを表わすのに都合のよいフレームは、図4に示すように階層的な表現をとることができるようになっており、複数のフレームにまたがって共通的な知識は上位のフレームに記述すれば、これらの知識は自動的に

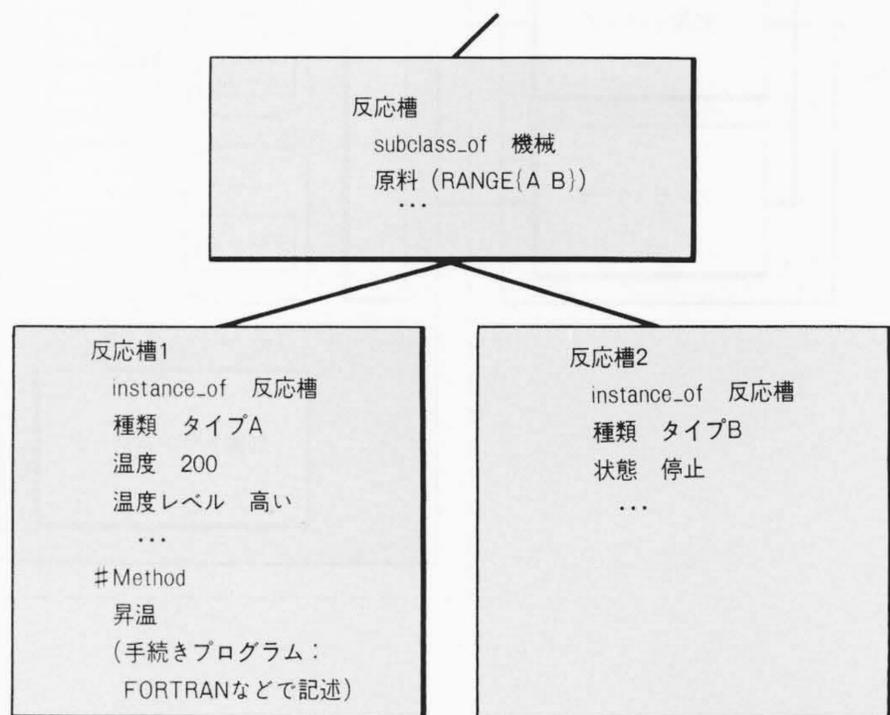


図4 EUREKAのフレーム記述例 プラントの構成機器の状態や機能に関する知識を、フレームによって記述する。階層的にフレームは表現できるので記述量を節約することができる。

```

(反応槽チェック
IF (?反応槽 の $種類 が タイプA であり..... ①
    $温度 が 200 以上である) ..... ②
THEN (send ?反応槽 set(温度レベル, 高い) ..... ③
      (reasoning([?反応槽]は不安定状態, >, 0.9))..... ④)
    
```

\$種類がタイプA, \$温度が200以上を満たすフレームがあればその名称を?反応槽とし、フレーム?反応槽の\$温度レベルを高いと設定し、「?反応槽は不安定状態」という事項の確信度が0.9以上になるまで後向き推論せよ。

```

(反応槽安定性推論
IF  [?反応槽] は 輝度むらがある >0.7 ..... ①'
    (?反応槽 の $温度レベル が 高い) ..... ②'
THEN [?反応槽] は 不安定状態 ..... ③' )
    
```

\$温度レベルが高いを満たすフレームがあれば、その名称を?反応槽とし、「?反応槽は輝度むらがある」という事項の確信度が0.7以上であれば「?反応槽は不安定状態」と結論づけよ。

図5 EUREKAのルール記述例 IF~THEN~形のルールによって操作形知識や因果性知識を記述する。日本語ふうの記述ができ、種々の項目に対して変数記述ができるので記述性に優れている。

に下位のフレームに受け継がれるようになっている。フレームは、データ部とメソッド部から成り、データ部にはフレームの属性値を記述し、メソッド部にはフレームの機能を表わすプログラムを記述する。このプログラムは、次に述べるようにルールから「昇温」というようなメッセージを受信すると自動的に起動するようになっており、ルール知識で表現する世界からみた場合のフレームの機能の表現になっている。

図5は、EUREKAでのルールの記述例を示す。ルールは、IF以降に記述する条件部とTHEN以降の結論部から成る。同図に示すように、条件部の表現は、①、②、②'と①のように二つのタイプがある。①、②、②'は、フレームに対する条件記述の例で、フレーム名称、属性名称、属性値に対する述語という三つの記述の組合せで表現する。ここに、?付きの名称は変数であることを示している。①'は、プライベートメモに対する条件記述の例で、括弧 [ ] で示された変数を含む任意文字列で表わされる一時的仮説の確信度の状況(この場合、確信度が0.7より大きいこと)で条件記述を行なう。

結論部の表現には、③、④、③'などのタイプがある。③、④は操作的知識を表わすもので、③は条件部を満たすフレーム「?反応槽」に対して、setという機能を実行せよというメッセージの送信を表わしており、④は、仮説「[?反応槽]は不安定状態」の確信度は0.9以上かどうか調べよという後向き推論の起動命令を表わしている。③'は因果性知識を表わしており、条件部が成立すれば結論部が成立することを示している。

推論は、条件部の成立したルールについてその結論部を実行するという形態で進む。④に示すように、後向き推論が起動されると、仮説の成否を調べるために、その仮説を結論部にもつルールの条件部が成立しているかどうか、ルールを後向きにたどってゆく。③に現われているset(値の代入)のような共通的な機能は、核ソフトウェア自体が準備しており、このような機能をもつフレームがシステムフレームとしてユーザー定義フレームの最上位に位置するために、フレームの階層構造的によって、ユーザー定義フレームに自動的に受け継

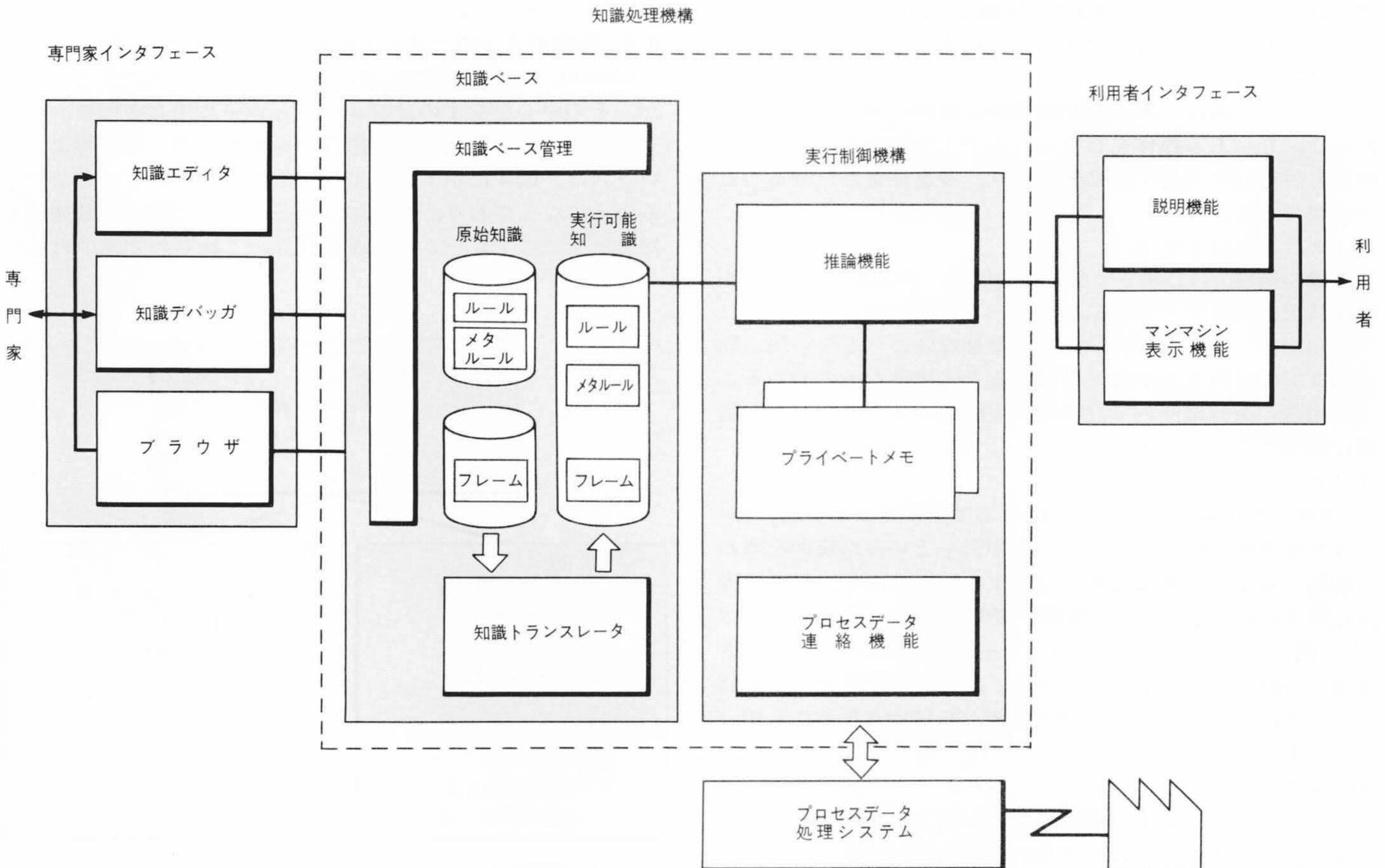


図6 EUREKAのソフトウェア構成 EUREKAは知識処理機構と専門家・利用者インタフェースから成る。入力された知識は、推論処理高速化のためにトランスレータによって内部コード(実行可能知識)に変換される。

がれる。

### 3.3 核ソフトウェアの構成と実行形態

EUREKAのソフトウェアの構成を図6に示す。このソフトウェアは、(1)問題解決の知識の保有者である専門家とのインタフェース、(2)知識処理機構としての知識ベース、実行制御機構、(3)知識の利用者とのインタフェースなどから成る。知識は、知識エディタを介して知識ベースに蓄積され、知識ベース内の知識は推論機能によって解釈・実行され、推論結果は利用者に提示されたり、プロセスに送信される。

#### (1) 専門家インタフェース

エディタ、デバッガ、ブラウザなどから構成される。知識ベースへの知識の入力、編集をエディタを介して行なうとともに、デバッガにより推論の試行を行ない、知識ベースの適切さを判断、修正などを行なう。ブラウザによって、入力した知識の構造を知ることができる。図7にこの画面例を示す。

#### (2) 知識処理機構

入力された知識は、推論機構が高速に処理できるように、内部的な表現に知識トランスレータによって変換される。推論機構は、プロセスの状況を表わすフレームの値の変化に応じて推論し、結果を利用者に提示したり、プロセスへ設定値として送信する。

推論処理の高速化のために、知識トランスレータは個々の知識の関連性を調べる<sup>4),5)</sup>。推論処理で、最も時間がかかるのは、ルールの条件部の成立判定である。ルールの条件ごとにその成否を調べるのでは、ルールの個数に比例して処理時間が増大する。このため、すべてのルールに対して、条件部の重複状況を調べ、重複した条件は一つにまとめた条件部成立判定のための木を作る。更に、推論機構での成立判定では、木の中に部分的な条件の成立状況を記憶しておき、判定に用いるデータが変化したところだけ追跡するというデータ駆動形の処理を行なう。このような推論処理の高速化手法の開発によって、ルールの数には依存せず、かつ実行速度も100 rule/s(V90/25)と実用的な推論処理性能を得ている(図8)。

#### (3) 利用者インタフェース

知識の利用者にとっては、なぜそのような結論に至ったかを知りたくなることがしばしばある。特に、仮説の成否を調べる後向き推論では顕著である。このために、結論に至った道筋を自動的に説明する機能を備えている。更に利用者への



図7 EUREKAの専門家インタフェースの画面例 デバッガを通じて推論にどんなルールやフレームが使われているか知ることができる。ブラウザによってフレームの階層的関係も知ることができる。

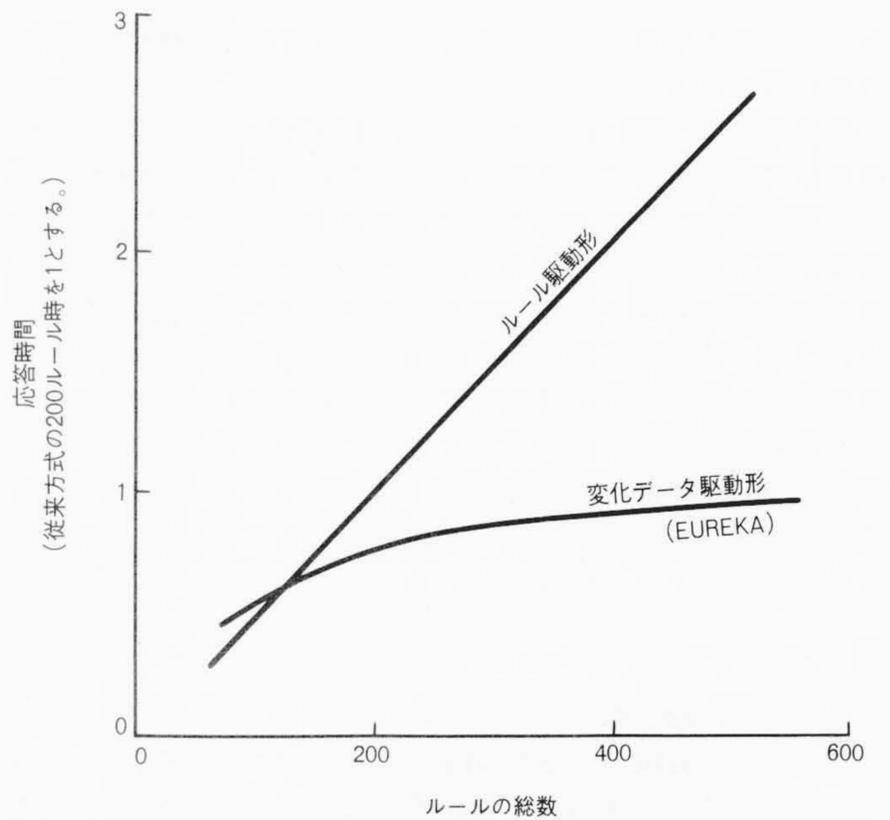


図8 変化データ駆動形推論方式の効果例 変化データ駆動形推論方式の開発により、ルールの総数に依存しない応答性能を実現している。

情報表示の方法としては、グラフィカルな表現が望ましい。このためには、グラフィック出力パッケージとの接続が必要であるが、これはフレームのもつメソッド機能によって実現することができる。

## 4 適用例

EUREKAは、既にプラント運転ガイダンス、診断など様々な業務に適用されている。その代表的な事例を紹介する。

### (1) 浄水場監視制御システム

大規模な浄水場では、監視制御項目数は1万点以上に及んでいる。このような多量の項目を監視し、個々の故障信号から故障箇所を判断し、処置を考えることは容易ではない。特に停電や主機の故障では、一度に大量の状態変化信号や故障信号が報告される。この状況では、真の故障原因を短時間に判断し、適切な処置をとるのはベテランの運転員でも容易ではない。更に、プラントは水の需要とともに成長し、時間とともにその構成は変化する。このため故障箇所の判断方法も変わり、かつ浄水場プラントは、土木、機械、電気の複合技術であり、各々の技術者のノウハウの集約化が要求される。

このような背景から、各々の技術ノウハウを容易に組み込むことができ、かつその内容の修正、追加が容易に行なえることをねらいとして、受変電設備を対象に故障診断のための知識処理システムを開発した。図9に示す知識は、機器信号の組合せに応じて、故障箇所がどこかを特定するものの例であり、このようなルール知識が約200個集められ、制御用計算機V90/50に蓄えられて故障時に備えられている。

### (2) 半導体プロセス診断システム<sup>6)</sup>

半導体製造では、多数のプロセス工程を経て製品ができる。このため、製品不良が発生したとき、どの工程が原因していたかを摘出するのは容易ではない。製品ペレットと同時に診断用テストパターンTEG(Test Equipment Group)がウェーハに組み込まれており、この電気的特性から不良原因を推定するという方法がとられていたが、プロセス工程が多数にわたるため、多人数のプロセス専門技術者の判断を集約して結論を出すのは容易ではない。このため、専門技術者の判

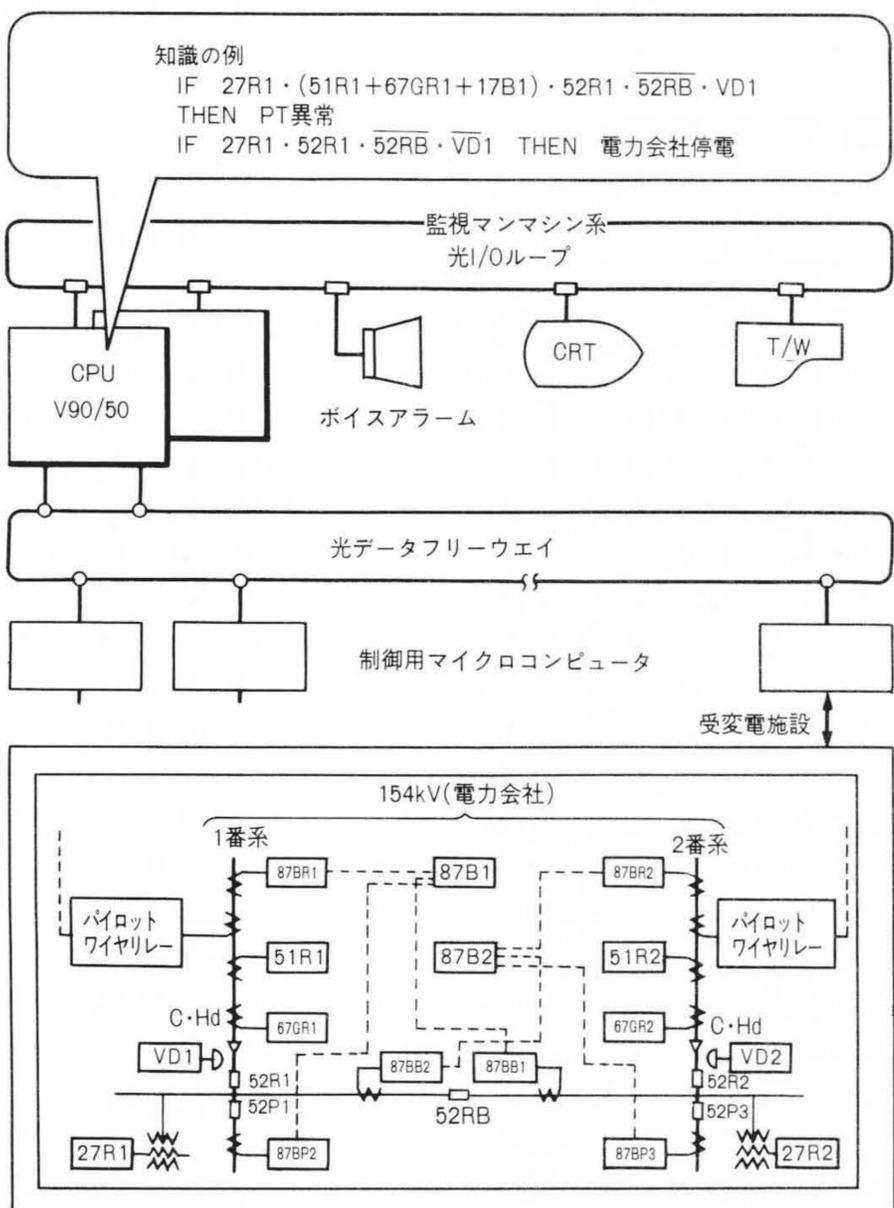
断ノウハウを知識ベース化することにより、統合的な判断を迅速に行なえるようにした。

具体的には、図10に示すように、TEGテストなどによって収集した電気特性データをエンジニアリングワークステーションES-310によって処理し、不良原因候補を提示するものである。不良原因の絞り込みは、二つのステップを踏む。まず、電気的特性から素子の構造的なパラメータを推定し、次にこのパラメータから不良原因を推定する。前者の推定では、両者の関係が理論あるいは実験的につかみやすいため数式モデルを用いている。後者の推定では、プロセス条件にかかわることが多いため、定量的関係づけは困難で、ルール形の知識により因果性を記述し、この知識を用いることによって最終的な結論を得るようになっている。

### 5 結 言

FAでの知識処理システムのニーズを概観するとともに、このシステムを実現するための核ソフトウェアEUREKAについて述べた。本ソフトウェアは、明快な知識表現形式をベースとし、多様な推論機能を備えていることから、ユーザーニーズに応じた幅広い知識処理システムの構築が可能となる。

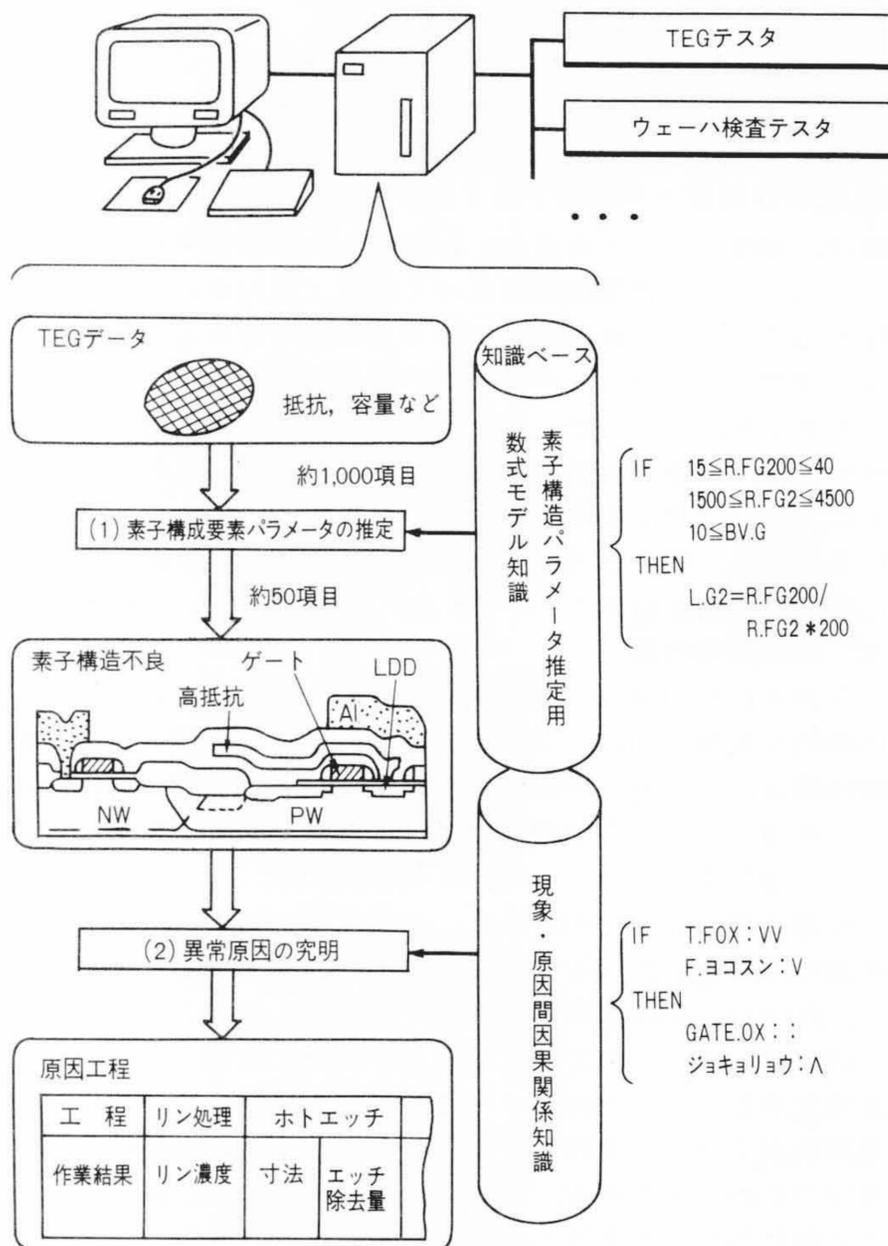
知識処理システムは、専門家ノウハウの計算機化という意味で一種の電子マニュアルととらえることができ、この観点に立つとその適用もさまざまなものが想定される。機能の高



注：略語説明 PI/O(Process Input/Output)  
CRT(Cathode Ray Tube), T/W(Type Writer)  
CPU(Central Processor Unit)

図9 浄水場監視制御システムへの適用例 プロセス構造に関する知識を利用して、多数の故障信号の中から真の原因を抽出し、処置方法をガイドする。

エンジニアリングワークステーション



注：略語説明 LDD(Lightly Doped Drain), TEG(Test Equipment Group)  
NW(Nウエル), PW(Pウエル)

図10 半導体プロセス診断システムへの適用例 多数のプロセス技術者のノウハウを集約して、検査データから不良原因工程を抽出する。

度化に伴って複雑化している種々の電子事務機器、自動車機器などの故障診断はその典型的な応用分野であろう。

一方、本文中にも指摘したが、知識処理システムは単に専門家の代行システムという位置づけだけにとどまらず、人間と計算機との親和性を高めることからソフトウェア人口を拡大し、ソフトウェア開発危機を乗り越える有力な手段の一つとみなすこともできる。本ソフトウェアをベースとし、組織に蓄積された知的資源をいっそう活性化させる知的分散システムが形成されてゆくものと期待している。

### 参考文献

- 1) 堺屋：知価革命，PHP研究所(1985)
- 2) 大須賀：知的情報処理の現状と課題，計測と制御，25，4，299～307(昭61-4)
- 3) 増位，外：知識制御核ソフトウェアEUREKAの記述体系，第29回情報処理全国大会論文集，1373～1374(昭59)
- 4) 田野，外：知識処理型ソフトウェアEUREKAにおける推論機構の高速化，第31回情報処理全国大会論文集，993～994(昭60)
- 5) 田野，外：知識処理ソフトウェアEUREKAにおけるルールネットワークの効率化方式，第32回情報処理学会全国大会論文集，1517～1518(昭61)
- 6) 栗原，外：知識ベースに基づく半導体プロセス診断システム，日立評論，67，12，963～966(昭60-12)