

知識獲得支援システム (EPSILON) における専門家モデル

滝 寛和*, 椿 和弘*, 岩下 安男**

* (財) 新世代コンピュータ技術開発機構

** 日本電信電話株式会社

本報告では専門家タスクの表現モデル (専門家モデル) と知識獲得方法 (プリ・ポスト法) について説明する。知識獲得の作業には2種類の主要な作業がある。その最初のフェイズは専門家タスクの表現モデル (専門家モデル) を構築するモデリング作業であり、もうひとつのフェイズは専門家モデルの具体化とリファインの作業である。専門家モデルは知識を獲得する為のタスクオリエンテッドな知識表現である。プリ・ポスト法は専門家に専門作業のタスクと関連知識の連想を促す知識獲得方法である。

EXPERT MODEL for Knowledge Acquisition

Hirokazu TAKI*, Kazuhiro TSUBAKI*, Yasuo IWASHITA**

* Institute for New Generation Computer Technology
4-28, Mita 1-Chome, Minato-ku, Tokyo 108 JAPAN
Csnnet:htaki@icot.jp@relay.cs.net

** Nippon Telegraph and Telephone Corporation
NTT Science. Bldg., 2-2-2 Otemachi Chiyoda-ku,
Tokyo 100 JAPAN

This paper describes an expert task representation model (EXPERT MODEL), and a knowledge acquisition method (Pre/Post Method) based on the EXPERT MODEL. There are two major phases in the knowledge acquisition process. The first phase is the modelling phase which builds an expert task representation model. The second phase is the EXPERT MODEL instantiation and refinement phase. The EXPERT MODEL is a task-oriented representation to extract the expert knowledge. The Pre/Post Method stimulates the human expert to remember his expert tasks and associative knowledge.

1. はじめに

知識獲得は知識システム構築におけるもっとも重要な作業である。知識獲得支援システムの開発には、ナレッジエンジニア（以下、KEと略す）の知識獲得作業の基本タスクと獲得方法を見付け出すことが非常に役立つ。「エキスパートシステム開発事例にみる知識獲得の諸相」[諏訪 86]によれば、知識システム構築過程はいくつかのフェイズに分けられることが報告されている。筆者らは知識獲得過程を次の4つのフェイズに分類している。

[1] 問題分析フェイズ

a) 問題の設定 b) 既存技術の評価 c) 知識源の同定

[2] 専門家モデル構築フェイズ

a) 概念構造の抽出 b) 専門家モデルの同定
c) ユーザモデルの同定 d) 専門家モデルの構築

[3] 専門家モデルの具体化フェイズ

a) 知識の抽出 b) 知識の整理とリファイン
c) 知識の変換 d) 知識ベースの評価

[4] 知識ベースの管理フェイズ

現在、開発されている知識獲得の方法は第3フェイズを知識獲得対象としているが、筆者らは第2フェイズを中心とした知識獲得支援システムを研究している。知識獲得の対象とする専門家の知識表現はいくつかの概念構造

[Sowa 83] [Bennett 85]を含んでいる。近年、いくつかの知識獲得支援ツールが開発される、これらのツールはそれぞれ概念構造を持っている。ETSは分類作業のための単純な概念構造を持っている

[Boose 84]、その概念構造には「結果項目」と「コンストラクト（特徴とその反対の特徴）」が含まれる。

MORE [Kahn 85]はETSより複雑な診断問題向きの概念構造を利用しているが、それには「仮説」、「微候」、「テスト」、「仮説の条件」、「微候の条件」、「テストの条件」、「微候の属性」、「リンク」と「パス」が含まれる。ROGET [Bennett 85]で扱う概念構造は実際のエキスパートシステムから抽出しており、タスク・オリエンテッドな概念構造となっている。ETSやMOREの概念構造には診断問題に内在する基本的概念構造を含んでいる。一般に専門家のタスクには基本知識（対象知識）とメタ知識（戦略的知識）がある。メタ知識には問題解決における基本知識の利用方法が含まれている。上記で説明したツールには、メタ知識が概念構造に入っていない。ユニークなエキスパートシステム構築方法はジェネリックタスク [Chandrasekaran 85]に見られる。ジェネリックタスク (Generic Tasks) のプリミティブは知識獲得の際に専門家の知識を表現するのに役立つ。ジェネリックタスクとしては、6種類が発見されているが、エキスパートシステムの構築に必要とされるジェネリックタスクがすべて発見されているとは言えない。また、ジェネリックタスクの表現レベルはまちまちであり、まだ十分に整理されているわけではない。筆者らは、効率的な知識獲得には基本的知識とメタ知識が明確に定まっている知識表現（専門家モデル）が必要であると考えている。ヒューリスティックの分類におけるジェネリックオペレーション [Clancey 85]も専門家のタスクをモデル化するのに役立つが、知識獲得に使用するにはオペレーションの単位が大きい。

そこで、筆者らは専門家モデルを知識獲得の為に概念構造とメタ知識を表現するモデルとして設計した。それには3つのレベルの知識表現が含まれている。専門家モデルの中心概念は構造ジェネリックオペレーションである。この構造ジェネリックオペレーションは7種類のオペレーションから成っている。それらは、「選択」、「分類」、「ソート」、「組み合わせ」、「変換」、「入力」と「出力」である。これらの基本オペレーションはプロダクションルールで書かれた診断エキスパート [諏訪 86] [ICOT 87]を分析することで得られた。専門家モデルの具体化作業においては、これらオペレーションに意味付けが行われる。意味付けされたオペレーションを意味ジェネリックオペレーションと呼ぶ。たとえば、「テスト装置の選択」は「選択」オペレーションに意味付けしたものである。ひとつのジェネリックオペレーションは「オペレーション属性」、「エバリュエータ（評価機構）」と「要素グループ（評価対象、評価結果）」から構成されている。エバリュエータは、ジェネリックオペレーションの実行部である。選択オペレーションのエバリュエータは、選択基準を評価する。ソートオペレーションのエバリュエータは、要素グループの要素の属性を比較して、比較結果の順に要素を整理する。ジェネリックオペレーションは、モデル化と具体化作業を通じて、知識ベースの一部となる。ジェネリックオペレーションを管理する機構が必要であるが、その表現として、オブジェクト指向アーキテクチャを採用した。これは、オブジェクト指向のメッセージパッシングモデルが人間の上層知識表現に適していると言われているからである。専門家モデルでは、ジェネリックオペレーションが1つのオブジェクトとなる。次章において、専門家モデルと効率的知識獲得方法（プリ・ポスト法）及びPSI [淵 84]上へのこの知識獲得方法の実現について説明する。

2. 専門家モデル

専門家モデルのオリジナルアイデアと、専門家モデルの構造・機能表現について詳しく説明する。

2.1 専門家モデルの基本アイデア

専門家モデルは、2つのアイデアに基づいて考案された。それは、「単純化した専門家タスクモデル」と「プロダクション記述の診断エキスパート知識の分析とグループ化」である。

(1) 単純化専門家タスクモデル

エキスパートシステムは、いくつかのタスクタイプに分けることができる。「Building Expert Systems」[Hayes-Roth 83]によれば、知識工学の応用には10の類型カテゴリーがある。それは「解釈」、「予測」、「診断」、「設計」、「計画立案」、「モニタリング」、「デバッグ」、「修理」、「教育」と「制御」である。著者らは、これらのカテゴリーをいくつかの簡単なモデルで表現することを試みた。なぜなら、単純化専門家タスクモデルは専門家に自分の知識を表現するための専門家作業イメージを提供することが出来るからである。ここでは、2つの例を示す。

＜単純化診断タスクモデル＞

第1図は診断タスクを示している。一般に、診断タスクは「診断木」と「探索アルゴリズム」で表現できる。しかし、その構造はあまりに一般的であり、専門家タスクをユニークに表現できない。より単純化したモデルが筆者らによって発見された。それはフィルターモデルである。第1図において、G1は故障可能な箇所のグループを意味する。F1は実際に故障箇所となる解を選択するフィルターである。G2は診断タスクにおける解（故障箇所）のグループを示している。

第2図も診断タスクを示している。このモデルでは第1図のタスクをサブ・タスクに持っている。

＜単純化設計タスクモデル＞

第3図は非常に単純化した設計タスクを示している。一般に、設計タスクはたいへん複雑なタスクであるので、このモデルでは設計作業の一部をモデル化してある。モデルは「部品モディファイ」、「部品の組み合わせ」と「良い結果の選択」の3つの部分から構成されている。G4とG5は部品グループであり、F4はモディファイヤー（改良修正のための変換処理）である。G10はモディファイされたG4のグループで、F5は組み立て、あるいは、組み合わせ処理である。G11は設計結果のグループである。F6は選択処理であり、最終設計結果を選び出す。G12が最終設計結果のグループである。（注：G5はモディファイの必要がないと仮定した。）

著者らは単純化専門家タスクモデルを検討し、これらのモデル（オペレーション）は「ソース要素グループ（評価対象グループ）」、「エバリュエータ（評価機構）」と「デスティネーション要素グループ（評価結果グループ）」から構成出来ることを発見した。

（2）診断エキスパートシステムのルールの分析

プロダクションルールはエキスパートシステムの為の一般的な知識表現である。それゆえに、KEは種々の知識を表す為にルール記述テクニックを持っていなければならない。筆者らはルールの記述形式にそのテクニックが表われているはずであるという仮説を立てた。プロダクションルールセットの中から次の7つの記述形式が見出された。それらは、「選択」、「分類」、「ソート」、「組み合わせ」、「変換」、「入力」と「出力」である。上記2つのアイデアの組み合わせ結果が「構造ジェネリックオペレーション」である。

2.2 専門家モデルの構造

専門家モデルは専門家からスムーズに専門家タスクを収集するために特殊化された知識表現である。知識獲得のためにはプロダクションルールのように一般化された知識表現は必ずしも必要ではない。むしろ、知識獲得に特化した表現が必要である。この節では専門家モデルの構成要素について説明する。

2.2.1 ジェネリックオペレーション

ジェネリックオペレーションは専門家モデルの中心的知識表現である。ジェネリックオペレーションは単純化専門家タスクモデルと診断型エキスパートシステムのプロダクションルールの分析から生れた。このジェネリックオペレーションには2つのタイプがある。それらは、基本的な専門家作業の構造を表す「構造ジェネリックオペレーション」と作業の詳細を表す「意味ジェネリックオペレーション」である。どちらも、知識獲得に役立つ枠組みである。その利点については第3章の知識獲得の方法の内容で説明する。

（1）構造ジェネリックオペレーション

構造ジェネリックオペレーションには上述のように「選択」、「分類」、「ソート」、「組み合わせ」、「変換」、「入力」と「出力」の7つオペレーションがある。各オペレーションは「ソース要素グループ」、「エバリュエータ」と「デスティネーション要素グループ」を持つ。各オペレーションについて説明する。

＜選択オペレーション＞

選択オペレーションはエバリュエータの選択条件に従ってソース要素グループから要素を選び、デスティネーション要素グループを作る。（第4図参照）

＜分類オペレーション＞

分類オペレーションは、分類条件（エバリュエータ）に従って、ソース要素グループの要素をいくつかのグループ（デスティネーション要素グループ）に分ける。選択オペレーションはこのオペレーションの一種である。（第5図参照）

＜ソートオペレーション＞

グループ内では各要素は評価順序を持っている。ソートオペレーションは並べ換え条件（エバリュエータ）に従って、ソース要素グループを並べ換えてデスティネーション要素グループを作る。

＜組み合わせオペレーション＞

組み合わせオペレーションは組み合わせ方法（エバリュエータ）に従って、ソース要素グループの要素を組み合わせ、新しい要素を作り、デスティネーション要素グループを組み上げる。（第6図参照）

＜変換オペレーション＞

変換オペレーションはデータ・アブストラクションやデータの解釈などに使用される。このオペレーションでは要素の属性が変更/追加される場合と新しい要素が生成される場合がある。

＜入力と出力オペレーション＞

エキスパートシステムとシステム利用者（または、観測

機など)との対話オペレーションである。入力オペレーションはソース要素グループなしにデスティネーション要素グループを生成する。出力オペレーションはデスティネーション要素グループを生成しないで、利用者ソース要素グループの情報を出力する。

(2) 意味ジェネリックオペレーション

意味ジェネリックオペレーションは構造ジェネリックオペレーションに意味と制御を付加したものである。意味ジェネリックオペレーションは意味と制御情報を表現するために、次の付加属性を持つ。(第7図参照)

星(*)で始まる属性はジェネリックオペレーションを具体化するときに使用される、それらを具体化属性と呼ぶ。他の属性は具体化作業のときに具体化属性の値を検査するのに使用される、それらを獲得タイプチェッカーと呼ぶ。もし、獲得タイプチェッカーの値が空なら、具体化時に検査は行われない。意味ジェネリックオペレーションには推論機構に対する(制御)情報が含まれている。それは、"Control_Type_Default"と"*Control_Type"の中に宣言される。探索(推論)のタイプには「全解探索」(All solution search)と「部分解探索」(partial solution search)が選択できる。(第8図参照)

逐次推論制御と並列推論制御に機能上で注目するような違いはないが、エキスパートシステムの実行効率性は並列推論の方が良い、また、専門家のタスクには本質的に並列表現のものがあるので、ジェネリックオペレーションには並列推論の表現がサポートしてある。通常のエキスパートシステムがサポートしている求解タイプは「全解探索」と「ひとつの解探索」である。一方、人間は複数解を求めてその中から最良と思われるものを選ぶ方法をよく行う。それゆえに、「部分解探索」をサポートしている。求解タイプがわからないときには、「Control_Type_Default」の値が利用される。

2. 2. 2 要素

要素は項目と属性から構成されている。要素はフレームベースの知識表現で、属性スロットとインヘリタンスを持つ。デーモンなどの付加手続は持たない。各要素はいくつかの要素グループに属し、ひとつ以上のジェネリックオペレーションからアクセスされる。

2. 2. 3 エバリュエータ

エバリュエータは条件と実行部(ときには、手続)から構成されている。記述形式はプロダクションルールの形式である。エバリュエータのグループをソース要素グループに持つジェネリックオペレーションもある。たとえば、計画問題での束縛条件の緩和処理では、沢山の束縛条件を持つ大きなエバリュエータグループをモディファイして小さなエバリュエータグループを作ることが要求される。選択ジェネリックオペレーションで処理されたエバリュエータグループ(デスティネーショングループ)をその計画問題のジェネリックオペレーションのエバリュエータとすればよい。

2. 2. 4 メタオペレーション制御:メタ・スクリプト

専門家タスクは問題解決のためのシナリオを持っている。それゆえ、専門家モデルではこのメタレベルのオペレーション制御をメタ・スクリプトと呼んでいる。メタ・スクリプトはメタレベルの知識を表現する知識表現言語である。知識ベースと専門家モデルにおける知識表現の階層と対応を第9図に示す。

専門家モデルはオブジェクト指向モデルであり、ジェネリックオペレーションがオブジェクトに対応する。ジェネリックオペレーション間の通信には、継承による通信とポスト・オペレーション呼出しの2種類がある。継承は"Operation_Class_Name"でサポートされており、ポスト・オペレーション呼出しは"Post_Operation_List"でサポートされる。ポスト・オペレーション呼出しのためのメッセージパッシングの制御を第10図に示す。

並列処理においては、"*Starting_Condition"の条件を満足するすべてのポスト・オペレーションはそのプリ・オペレーションの実行後に同時に並行処理される。

3. 専門家モデルに基づく知識獲得

この章では、専門家モデルに基づく知識獲得方法、プリ・ポスト法について紹介する。専門家から知識を獲得する作業には2つある。それは、専門家の知識の構造をイメージとしてまとめるモデリングフェイズ(専門家モデルの構築)とそのモデルを詳細化する具体化フェイズ(専門家モデルの具体化)である。KEの作業では、知識ベースは段階的に作られる。最初の知識ベースは初期知識ベースと呼ばれる。それは、「専門家タスクの中での問題は何か」、「専門家はいかにその問題を自分の知識を使って解くか」をKEと専門家と利用者が学ぶのに役立つ。一般に初期知識ベースには問題解決に利用する主要な項目とその関係のみが含まれているだけで、問題解決の効率化のためのメタレベルの知識が含まれていることはまれである。この段階的知識ベース構築を第11図に示す。

3. 1 プリ・ポスト法

プリ・ポスト法は第11図の横方向(モデル構築-)モデル-)モデル具体化-)知識ベース)の処理を行う知識獲得方法である。この方法にはいくつかのモデル構築戦略が含まれる。この方法の主たる戦略は、あるオペレーションに関係する前後(プリ・ポスト)オペレーションの連想を専門家に促す。専門家にとって、あるオペレーションの前後にどのようなオペレーションが必要であるかを述べることは比較的易しい。たとえば、車が動かないときに、「エンジン検査のまえに何を行うか?」の質問に対して、人間は、「ガソリンの残量を調べる。」や「バッテリーを調べる。」などを容易に答えることができる。プリ・ポスト法に従った知識獲得過程を次に示す。

[ステップ1] 専門家作業のオペレーションを幾つか収集する。各オペレーション間には関係がなくてもよい。オペレーションはランダムに収集してよい。これらのオペレーションは知識獲得の出発点となる。

【ステップ2】収集したオペレーションについてその前後（プリ・ポスト）オペレーションを質問する。このステップは専門家から新しいオペレーションが出なくなるまで続けられる。（新しく収集したオペレーションについてもその前後オペレーションを収集する。）

【ステップ3】ここまでで、多くのオペレーションとその前後関係情報が集まった。このステップは前後関係をチェックするためのステップである。前後関係をチェックする方法のひとつに専門家にグラフ的にこの関係を表示する方法がある。

第12図は、オペレーションの前後関係を表示している。関係-1はPre-1からPre-3がポイント(Point)オペレーションのプリオペレーションであり、Post-1からPost-3がそのポストオペレーションであることを示している。関係-2では、新たな順序として、Pre-1がPre-3のプリオペレーションでPre-2がPre-3のポストオペレーションで、Post-1がPost-2のポストオペレーションであることが分かったときの関係を示している。前後関係情報が収集されるにつれて、その関係はより明確になる。オペレーションによっては、順序が不明確なものもある。そのようなオペレーションのグループは「弱い順序戦略グループ」として取り扱う。このタイプのグループに属するオペレーションの処理が完了したときに、そのオペレーションの結果が残りのオペレーションに影響することがある。オペレーション間の影響の束縛情報を収集することがこれらのオペレーションにとって必要である。

【ステップ4】このステップは各オペレーションを具体化するためのステップである。各オペレーションは専門家によって構造ジェネリックオペレーションの何れかに対応付けられなければならない。ときには、専門家の示したオペレーションが大きなタスクであり、いくつかのサブタスクを含んでいることがある。これらの大きなタスクを、小さなタスクに分解するには、サブタスクのいくつかを専門家に述べてもらい、そのサブタスクについて、ステップ2以下の作業を行う。

【ステップ5】このステップはオペレーションをマージ（融合）するためのステップである。前後関係とジェネリックオペレーションのタイプ（7つのタイプ）から、おなじオペレーションをひとつにマージする。

【ステップ6】各オペレーションはエバリュエータを持っているが、このエバリュエータの内容を明確にするステップである。オペレーションの役割を明確にするのは易しい、なぜなら、オペレーションの位置は前後関係より分かり、機能はジェネリックオペレーションのタイプで示されているからである。

【ステップ7】このステップではソース要素グループの要素を決めなければならない。ときには、オペレーションのソース要素グループやその要素を決めるのが難しい場合がある。ソース要素グループはそのプリオペレーションの実行後に定まることの多いからである。そのような場合、プリオペレーションのデスティネーション要素グループの中からソース要素グループの候補を見付けるようにするのがよい。

【ステップ8】このステップは要素の属性とその値を決定するためのステップである。各要素はエバリュエータで評価されるべき属性を持たなければならない。一般に各要素は幾つかのジェネリックオペレーションを通過する。そのジェネリックオペレーションのエバリュエータが評価する属性を持つ必要がある。

3. 2 プリ・ポスト法のインプリメント

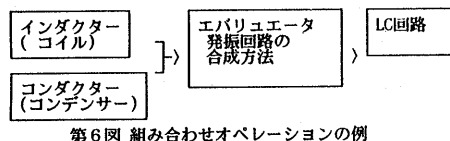
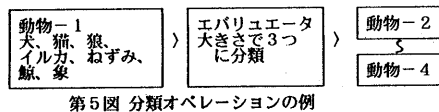
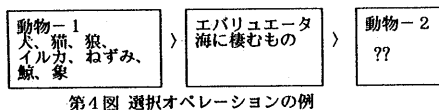
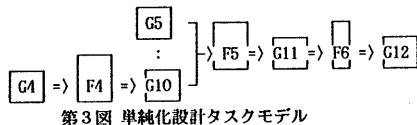
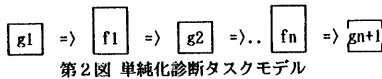
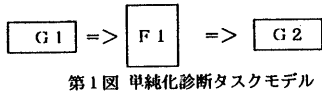
プリ・ポスト法のインプリメントにはオペレーションの前後関係を表示し、専門家がその関係をリファイン出来る環境でなければならない。マルチ・ウインドウ表示機能を持ったワークステーションはその関係を表示するのに適した装置である。また、獲得した知識ベースは、専門家タスクを十分に満足するかを、実際に推論機構により、動的に処理して評価されなければならない。専門家モデルはエバリュエータにルール形式を、要素にフレーム形式を、メタ・レベル制御にオブジェクト指向アーキテクチャーを採用している。そのため、“EPSILON/EM”と呼ぶプリ・ポスト法を使った知識獲得支援システムをPSI上に開発中である。あるオペレーションの内容を表示したウインドウの例を第13図に示す。

4. 結論

診断エキスパートシステムのプロダクションルールの分析と単純化専門家タスクモデルから考案した専門家モデルについて紹介した。また、専門家モデルに基づく、知識獲得方法として、プリ・ポスト法について説明した。知識獲得過程には、専門家モデルを構築するフェイズとそのモデルを具体化し、リファインするフェイズがある。プリ・ポスト法は専門家タスクとそれに関係する知識の連想を専門家に促すことにより、上記の知識獲得過程を支援する。専門家モデルはメタレベル知識（オペレーションレベル知識）を中心とした知識表現である。対象知識レベルの構造は要素の属性表現としてサポートされているが、その属性の決定は専門家の示す情報に依存している。対象知識についてもオペレーション同様に類型(generic)構造のサポートが必要である。また、オペレーションの階層構造の表現とその効率的獲得方法も今後の研究課題である。

5. 謝辞

本研究はICOT-KAS（知識獲得システム）ワーキンググループの資料に基づいている。ワーキングの諏訪博士（電総研）、小林助教授（東工大）ならびに各委員に深く感謝します。また、専門家モデルの研究に御助力いただいた第5研究室の藤井室長に感謝します。最後に、第5世代コンピュータシステムプロジェクトにおいて本研究の機会を与えていただいたICOT淵所長に深く感謝致します。



Semantic Generic Operation:

Operation_Name: XXXX

Operation_Class_Name: YYYY

Pre_Operation_Class_List: {01,...|0n}

Post_Operation_Class_List: {Pol,...|Pom}

Syntactic_Operation_Class: {Select,..|Output}

Source_Group_Type: ZZZZ

Destination_Group_Type: AAAA

Evaluation_Group_Type: BBBB

Control_Type_Default: {Sequential/Parallel..}

*Operation_Instance_Name: xxxx

*Pre_Operation_List: {01,...|0n}

*Post_Operation_List: {pol,...|pon}

*Source_Group_Name: zzzz

*Destination_Group_Name: aaaa/auto_generate

*Evaluation_Group_Name: xxxx

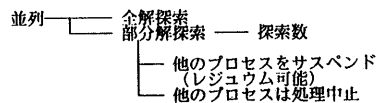
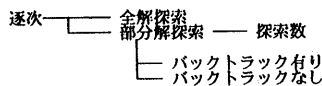
*Control_Type: {Sequential/Parallel..}

*Starting_Condition: {....}

*Send_Message: {....}

第7図 オペレーションの属性

<制御タイプ: ジェネリックオペレーション内>



第8図 ジェネリックオペレーション内の制御

専門家モデル	具体化された知識ベース
メタスクリプト	大局的メタ知識
ジェネリックオペレーション	局所的メタ知識
要素・エバリュエータ	基本的知識 対象知識

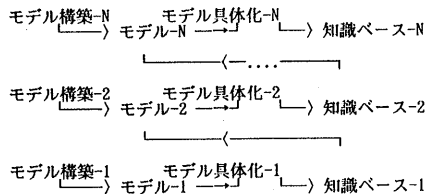
第9図 専門家モデルにおける知識レベルの階層

<メタレベルの制御タイプ: オペレーション外>

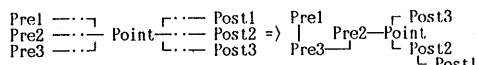
オブジェクト選択 (逐次) — 他オペレーションはバックトラックで他のオペレーション実行可能

並列処理 — 条件が満足されるオペレーションをすべて並行に実行

第10図 メタスクリプトレベルオペレーション制御



第11図 段階的知識ベース構築



第12図 プリ・ポスト関係の例

第13図 オペレーション内容のウィンドウイメージ

[参考文献]

[源助 86] 源助 基, 他: エキスパートシステム開発事例に
みる知識獲得の諸相, 計測と制御, Vol.25, No.9, 1986.

[Bennett 85] Bennett, J.: ROGET: A Knowledge-Based
System for Acquiring the Conceptual Structure of a
Diagnostic Expert System., Journal of Automated
Reasoning 1, 49-74, 1985.

[Boose 84] Boose, J.: Personal construct theory and
the transfer of human expertise., In Proceedings
of the National Conference on Artificial
Intelligence. Austin, Texas, 1984.

[Butler 86] Butler, K.A. and Corter, J.E. : Use of
Psychometric Tools for Knowledge Acquisition : A
Case Study, Gale ed., Artificial Intelligence &
Statistics, pp.295-319, 1986.

[Chandrasekaran 86] Chandrasekaran, B. :
Generic Tasks in Knowledge-Based Reasoning:
High-Level Building Blocks for Expert System
Design., IEEE Expert, Fall 1986.

[Chikayama 84] Chikayama, T. : Unique Features of
ESP, International Conference on Fifth Generation
Computer Systems, November 1984.

[Clancey 85] Clancey, W.: Heuristic
classification., Artificial Intelligence 27, 1985.

[Eshelman 86] Eshelman, L. and J. Mcdermott.:
MOLE: A Knowledge Acquisition Tool That Uses its
Head., In Proceedings of the National Conference on
Artificial Intelligence., 1986.

[Hays-Roth 83] Hays-Roth, F., Waterman, D.A., Lenat, D.B. :
Building Expert Systems, Addison-Wesley Publishing
comp., 1983

[ICOT 87] ICOT Technical Report, "Knowledge Bases
for Expert Shell Evaluation", 1987 (to appear).

[Kahn 85] Kahn, G., S. Nowlan, and
J. Mcdermott.: Strategies for knowledge acquisition.,
IEEE transactions on Pattern Analysis and
Machine Intelligence 7(5), 1985.

[Miki 87] Miki, M. and Taki, H. : The Knowledge
Acquisition System with Fault Tree and Diagnostic
Flow Interface., Proceedings International Workshop
on Industrial applications of machine vision and
machine intelligence, Tokyo, 1987.

[Sowa 83] Sowa, J.F. : Conceptual Structures,
Addison-Wesley, 1983.

[瀬 84] Taki, K., Yokota, M., Yamamoto, A.,

Nishikawa, H., Uchida, S., Nakajima, H. and
Mitsuishi, A., Hardware Design and Implementation of
The Personal Sequential Inference
Machine (PSI), International Conference on Fifth
Generation Computer Systems, November 1984.