

TM-0704

知識コンパイル、ジェネリック・タスク

滝 寛和

March, 1989

© 1989, ICOT

ICOT

Mita Kokusai Bldg. 21F
4-28 Mita 1-Chome
Minato-ku Tokyo 108 Japan

(03) 456-3191-5
Telex ICOT J32964

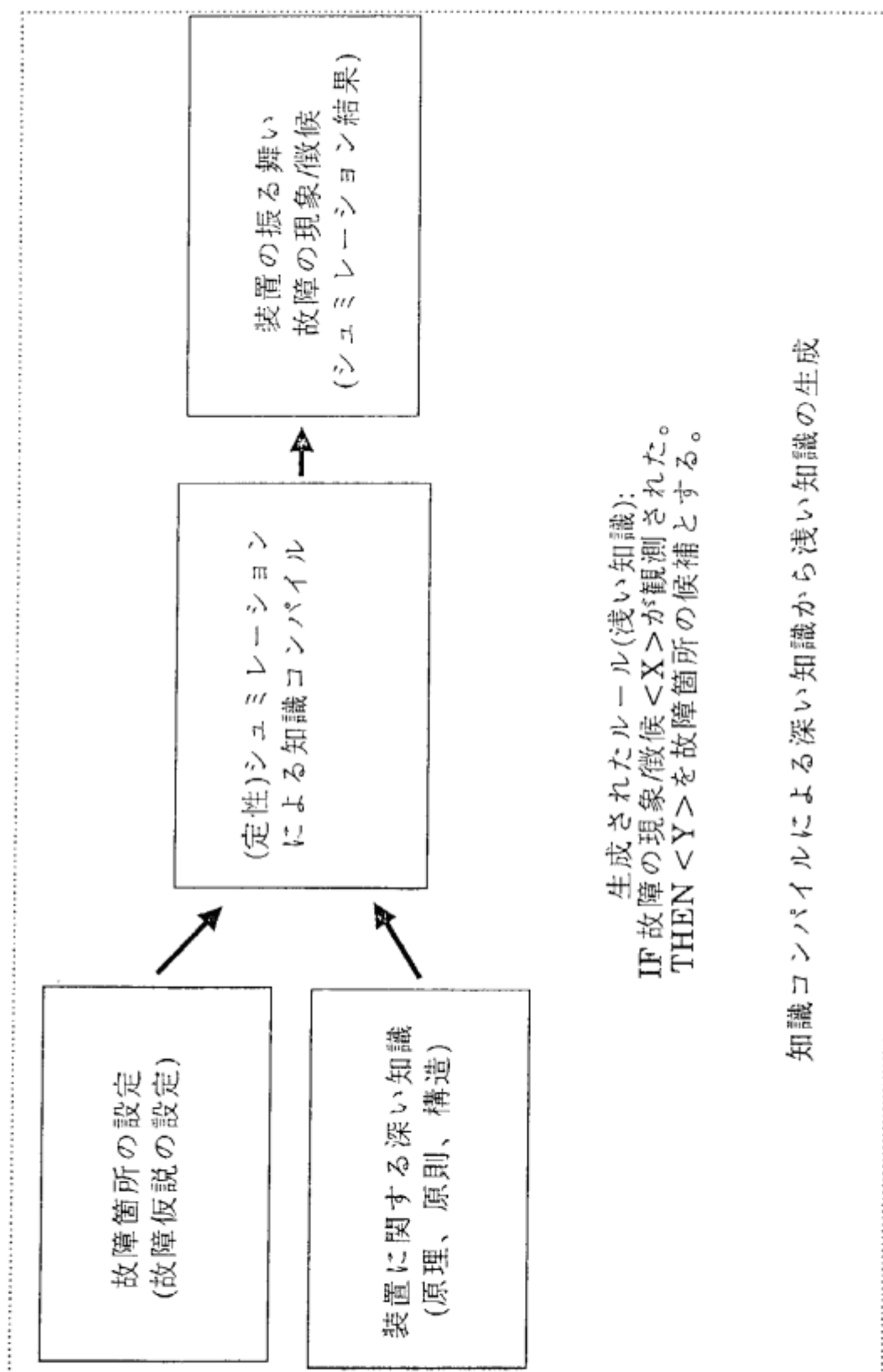
Institute for New Generation Computer Technology

2.3.2.3 知識コンパイル

一般に、知識を効率的な知識に変換することを言う。代表的な知識コンパイルには、『深い知識(原理、原則、構造に関する知識)から浅い知識(経験則など)の生成』、『知識を適用例に合わせて、効率的な知識を生成する部分計算』(前節 2.3.2.2 変換 最適化技術 参照)、『制約で表現された知識を解析して、制約を解く順序を生成する制約コンパイル』[Araya 87][寺崎 88]などがある。ここでは、深い知識から浅い知識を生成する知識コンパイルについて説明する。いま、ある装置が与えられて、その装置の診断エキスパートシステムを構築することを考える。診断過程は、故障の状況から効率的に、故障箇所を推論する過程である。しかしながら、その装置の構造情報(設計図)や装置の動作原理に関する知識は、比較的容易に用意できるものの、深い知識で推論を行うと、非常に詳細な知識であるため、推論の計算量は大きくなる。また、選択肢が複数ある場合に、どの選択肢も同じ様に扱われるために、必ず失敗する肢や可能性の低い肢を探索することになり、推論の効率は良くない。故障に関する因果関係つまり、故障の状況と故障箇所を短絡的に結ぶ知識があれば、診断に必要な推論の計算量を減らすことができる。この短絡的な知識が浅い知識である。人間の専門家は、経験により、この浅い知識を獲得している。深い知識から浅い知識の生成には、現在、2種類の方法が提案されている。1つは、定性推論を利用した方法であり、もう1つは、説明に基づく学習を利用した方法である。定性推論を利用した方法では、深い知識を定性的に扱って、故障状態での装置の振る舞いをシミュレーションすることで、故障の原因と故障の徴候(状況)の関係を求めることができる。この方法を利用したシステムとして、プロセスプラントの診断知識を生成する AQUA[石田 87]、眼圧調整の定性モデルを利用した診断ルールの生成[大和田 88]や、知識コンパイラ[山口 87]がある。説明に基づく学習では、故障診断の例(訓練例)を深い知識(領域知識)を利用して、演繹推論で説明し、その説明過程(推論木)をある基準(操作性規範)で一般化することで、マクロな知識(短絡的知識)を浅い知識として生成することができる[Mitchell 85]。

参考文献

- [大和田 88] 大和田、溝口、北沢：定性シミュレーションに基づく診断システムの構築法，人工知能学会誌，Vol.3 No.5，pp87-96，1988
- [石田 87] 石田：定性推論を利用したプロセス診断知識獲得，日本ソフトウェア科学会第4回 大会論文集，D-4-1，1987
- [寺崎 88] 寺崎、永井、滝、他：機械設計支援システム構築ツール MECAHNICOT，人工知能学会 第4回 知識ベースシステム研究会資料，pp19-28，1988
- [山口 87] 山口、溝口、角所、他：深い知識に基づく知識コンパイラの基本設計，人工知能学会誌，Vol.2 No.3，pp333-340，1987
- [Araya 87] A.A. Araya and S. Mittal: Compiling Design Plans from Descriptions of Artifacts and Problem Solving Heuristics, Proc. 10th IJCAI, 1987
- [Mitchell 85] T. Mitchell, R. Keller and S. Keder-Cabelli: Explanation-Based Generalization: A Unifying View, Machine Learning 1, 1985



2.3.2.4 ジェネリックタスク(類型的タスク)

ジェネリックタスク [Chandrasekaran 86]は、オハイオ州立大学の Chandrasekaran教授が提唱したエキスパートシステム開発の方法論である。氏は、現在のエキスパートシステムの開発が単純なルール表現(プロダクションルール)でのみ開発されていることに疑問を持ち、問題解決的視点でエキスパートシステムの知識ベースを構築する方法を提唱している。

氏は、いくつかのエキスパートシステムの開発経験から、人間の問題解決の構造の類型パターンを見出した。この問題解決の要素をジェネリックタスク(Generic Tasks)と呼ぶ。現在までに、見つかったジェネリックタスクは次の6種類である。

- (1) 階層化分類(classification)
- (2) プラン選択と詳細化による設計
(design by plan selection and refinement)
- (3) 知識データの検索(knowledge directed data passing)
- (4) 仮説生成(abductive assembly of hypothesis)
- (5) 状態のアブストラクション(state abstraction)
- (6) 仮説照合(hypothesis matching)

実際のエキスパートシステム開発には、このジェネリックタスクの幾つかを組み合わせて問題解決を行うことになる。氏は、実際のエキスパートツールとして、階層分類タスクと仮説照合タスクを組み合わせた診断エキスパートシステムシェル CSRL と、プラン選択と詳細化による設計タスクを使ったルーチン設計エキスパートシステムシェル DSPL を開発している。ジェネリックタスクを利用したエキスパートシステムでは、問題解決の視点から、その問題解決特有の知識表現で、知識を収集することで、知識ベースを構築できる。このような理由から、従来のルール型知識表現のみをサポートしているツールに比べて、知識の獲得が容易になる利点がある。

参考文献

- [Chandrasekaran 86] B. Chandrasekaran: Generic Tasks in Knowledge-Based Reasoning: High-Level Building Blocks for Expert System Design. IEEE Expert, Vol.1 No..3, pp23-30, 1986

