

ディシジョンラティスに基づく診断型エキスパートシェル — その『ところ』 —

5D-5

吉田裕之, 大石和弘, 中島 淳, 原 裕貴, 松本 均
(yuki@flab.fujitsu.junet)
富士通研究所

1. はじめに

エキスパートシステム作成支援ツール (エキスパートシェル) は, 当初は既存のエキスパートシステムの推論エンジンをそのまま提供する形であったが, やがて多種多様な業務への適用を可能にすべく高機能化・汎用化がはかられた。しかしこのような汎用のエキスパートシェルの使い勝手は必ずしも良いものとは言えなかった。なぜならばこれらの利用者, すなわち特定の問題に関するエキスパートシステムの開発者は, その問題自身・その解法・そのための知識を「モデル化」する必要がある, この作業こそエキスパートシステム開発に関する多くの経験と深い造詣を必要とする高度なものだったからである。

これに対していわゆるドメインシェルという考え方がある。これは一言で言えば, 特定のタイプの問題に対する「モデル化の方針(抽象モデル)」をエキスパートシステム開発者に与えることである。このモデル化に適さない問題・業務はカバーできなくなるものの, 適している場合にはエキスパートシステム開発者は自分の問題を与えられた抽象モデルにマッピングするだけでモデル化作業が行える。またエンドユーザ向けのインタフェース構築や知識ベース管理といった点もきめこまかくサポートすることが可能になる。

我々は診断・分類問題に対する抽象モデルとしてディシジョンラティスモデルを提唱し, その理論的側面について文献 [1,2,3] で論じた。本稿では, このモデルに基づいた診断型エキスパートシェル *De La* の構成と, エキスパートシステム開発者及びエンドユーザの立場から見た特徴を述べる。

2. 診断問題に対するモデル

診断問題に対する典型的なモデル化は, 原因の階層構造に則り, 発行する質問を制御することによってこの階層をトップダウンに確認していくことで, 原因を絞り込むものと言える。この場合, 知識は「○

○であることが分かったら, 次に△△を質問すればさらに□□であることが分かる」のような形になる。ここで□□は, 中間仮説である○○をより具体化した仮説である。このような知識の連鎖を辿って最終的な仮説, すなわち原因に到達するのが診断問題の解法である。

この手法の問題点は「知識の局所性」にある。つまり各知識が前提 (○○であること) が満たされない限り適用されない点である。これによって例えば, ある質問をどの前提のもとに発行するのが最も効果的かという点を開発時に強く意識しなければならない。このことは知識ベース全体の構造が「固い」ことを意味し, 知識の追加・修正を困難にする要因となる。また例えば, 異なる前提のもとで同じ質問項目に対して相反する結論を(間違えて)与えてしまった場合にその発見が困難である。これは知識がある意味で手続き的なので, 「知識ベースが無矛盾である」ことの論理的な定義ができないことから生じると言える。さらにエンドユーザが火急に訴えたい症状があっても, それに関する知識の前提が満たされるまで待たされるという不具合もしばしば見られる。

3. ディシジョンラティスモデル

[用語]

症状: 質問とそれに対する観測値との対。

観測状況: 症状の集合。

候補原因: 原因のうちで実際に起きている可能性があるもの。

棄却原因: 原因のうちで起きている可能性が棄却されたもの。

原因仮説: 候補原因への確信度の割当てと, 棄却原因の集合との対。確信度は Dempster-Shafer 理論に基づくものである [1]。

診断知識: 観測状況とそれに対する原因仮説との対。

ディシジョンラティスモデルにおいては「観測された症状あるいは症状の組から, 専門家はどんな仮説を立てるのか」が診断のために必要な知識であって, その次に何を質問すれば良いかなどは二義的な要素であるとする。なぜならば, その次に入力される症状によって仮説がどう変化するかが分かるなら

ば、それから逆にどの質問が最も効果的かは簡単に求まるからである。

こうした立場に立てば知識は大局的かつ宣言的であるので、知識ベースの無矛盾性や不足部分の予測等を論理的に議論できるようになる[2]。またエキスパートシステム開発者は集めてきた知識を順不同に知識ベースに格納していけば済むし、与えられたケースに対してシステムが妥当な仮説を立てたかどうかだけに注目してデバッグを進めることができる。新しい知識・質問項目・原因の追加は元からある知識に(矛盾しない限り)影響しない。さらに、エンドユーザは症状を思いついた順に入力することができる。

4. De La の構成

De La システムの構成を図1に示す。De La は大きく二つのモジュールからなり(カーネルモジュール:K-Module, ユーザインタフェースモジュール:U-Module), これらは別プロセスとして動作し socket を用いて結合されている。したがってユーザが観測を入力したり診断結果を確認している間に、バックグラウンドで K-Module がさらに処理を進めておくことができる。また、これらを別のマシン上で動かすこともでき、負荷分散を図ることができる。

K-Module は Lisp で記述されており、主な機能は(1) 仮説の生成, (2) 質問の評価, (3) 知識ベース管理, (4) 知識獲得支援である。特に知識獲得支援機構は、特殊すぎる知識が入力された時にその一般化をエキスパートシステム開発者に提案し、また知識ベース管理機構の無矛盾性維持動作に伴って間違っていた知識が削除される時にその特殊化を提案する。

U-Module は C 言語で記述されており、X-window System の一つのクライアントである。De La の一つの特徴は、開発者向けのインタフェースとエンドユーザ向けのインタフェースにほとんど違いが無い点であり、U-Module は開発時にも実行時にも利用されることになる。その形態の詳細については文献 [4] で詳しく述べる。

5. De La の特徴

診断実行時における特徴は、(1) 現在の仮説を最も効果的に良くする質問をシステムが発行してくる、(2) エンドユーザは通常はそれに従って答えていけばよいが、システムの評価を無視して火急に訴えたい症状あるいは検査に伴って偶然発見した症状を入力しても構わない、(3) 以前に入力した症状を特別な操作無しで随時変更・削除できる。

開発時における特徴は、(1) 診断結果と知識(観測状況+原因仮説) がまったく同じ形なので実行時と同じインタフェースが利用できる、(2) 知識の入力順序・適用順序などをいっさい意識する必要が無い、(3) 結果の間違いを発見したら正しい結果を指示するだけで良く、間違っていた知識をトレースなどによって発見する必要が無い。

6. おわりに

現在我々は De La を自動車の故障診断に適用してその効果を評価している。実行速度上でネックになるのは質問の評価であり、質問項目数に線型以上の時間がかかることが分かっている。実際にはこの部分はバックグラウンドで動作しているので質問項目総数百程度の規模の応用例ではあまり気にならないが、さらに評価候補の刈り取り法の検討を進めている。

参考文献

- [1]原, 松本, 中島, 大石:「診断問題の推論制御に関する一考察」, 人工知能学会人工知能ツールと知識システム研究会資料, 1987年12月
- [2]原, 吉田, 松本:「ディジションラティス:診断問題の定式化」, 情報処理学会知識工学と人工知能研究会資料, 1988年7月
- [3]原, 吉田, 大石:「診断型エキスパートシェル De La における症状間の関係の利用について」, 電子情報通信学会人工知能と知識処理研究会資料, 1988年12月
- [4]大石, 中島, 原, 吉田:「ディジションラティスに基づく診断型エキスパートシェル—その『すがた』—」, 本大会

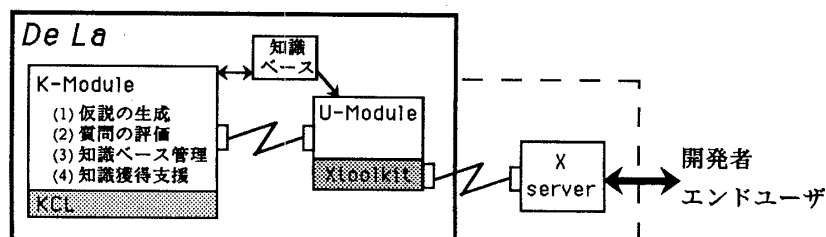


図1 システム構成