

知識ベースの保守支援機能に関する検討

5 D - 9

浅見 徹 山口 典男 橋本 和夫
国際電信電話株式会社研究所

1 はじめに

実用的なES(エキスパートシステム)あるいは知識ベースシステムを開発する上で、考慮すべき最も重要な点は、システム導入後も引き続き行なっていかなければならない知識の更新作業を誰に行なわせるかということにある。通常システムの開発段階では、開発すべき分野の専門家とKE(場合によっては、システムエンジニア、プログラマ等も参加することもある)からなるプロジェクトチームを編成し、組織力でシステム作成を行なうことができる。対象タスクの分析、モデル化、システム化、プログラミングいずれを取っても最適の人材を一定時間拘束して作業させることができる。従って、新しいプログラミング言語を修得するのに要するコストや期間を考慮すると、所謂エキスパートシステム作成用のツールを使わず通常プログラミング言語や環境を使っても、システム構築コストの面で大幅な差異が生じない場合もある。

一方、システム導入後、その効果を問われたとき、(1)専属のKEを配置できることは稀であり、(2)分野専門家を長期にシステムのために拘束することはできない条件下で判定されることが多い。従って、作成したESの維持の際には、上記の条件下でコストパフォーマンス(そのシステム導入による省力化の程度/維持コスト)を問われることになる。即ち、分野専門家が自分達の作業環境を向上させるため、作業時間の合間を見てシステムにアクセスし、システムの知識の更新を図れるようなシステム設計であることが、必要条件と言える。

知識獲得に関する各種のツールは、従来図形の入力、表示、マウス/アイコンの使用等を前提としたものが多い。このように、システムのコンソールでなければ知識の修正ができない構成を採っていると、分野専門家がパートタイムでアクセスし、システム維持を図る機会を著しく損ねることになる。本稿は、診断型やコンサルテーション型を対象としたシステムの保守支援のため開発を進めている、(i)通常の漢字CRT端末で知識の入力編集を行なえ、(ii)表形式での知識の入力編集機能を持ち、(iii)表を判断用と手続き記述用に最適化して持ったES保守支援機能について述べる。

2 システム保守環境

ここでは、ESの保守支援に関して、保守者が、図1に示すように何らかの意味でコンソール以外の端末から知識の投入編集を行なうことを想定している。これは、専

門家が複数いて知識の入力を分散して行なう場合、編集対象となる知識ベース自体を分散させてしまうと、知識の整合性を保つことが非常に難しくなるため、編集対象を一つにすることによってこの問題を回避する必要があるからである。従って、使用端末としては、VT100程度

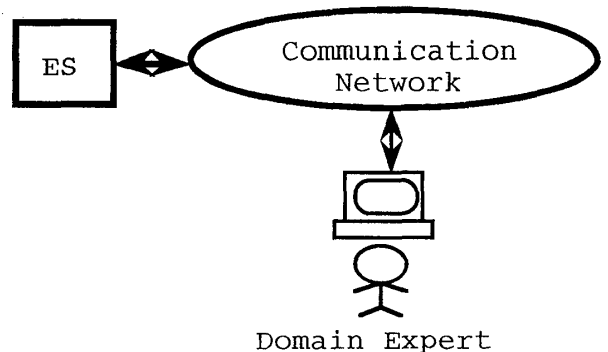


図1. システムの知識編集環境

の漢字(シフトJIS)端末程度のものを想定し、野線等の表示機能はある端末を仮定するにとどめている。入力には、キーボード入力のみである。

3 知識の表形式入力

表形式の入力の場合、入力者は特定のプログラミング言語等を特に意識せず、日常使い慣れている表の意味の延長で、書かれている知識の解釈ができるため、非計算機分野の人々に入力させる場合、有用である。既にいくつかの開発事例で、表形式の入力が簡単な診断型のES開発支援に有用であることが、明らかとなっている[1]。また、多くの知識獲得ツールにおいて、システムの知識の入力を対象分野専門家のような非KE/非プログラマに任せる試みを行なう場合、多かれ少なかれ表形式に似た形式の知識入力になっていることも事実である[4]。

しかし、表形式入力の場合、表の各行/列にいかなる意味づけを行なうかに関しては、完全に入力者任せである。従って、例えば表の各要素が特定のターゲットプログラミング言語の制御構造に対応していたりする場合、意味付けに自由度があり過ぎ、対象分野専門家のような非計算機技術者に理解させて作業させるのはかなり無理がある。そのため、その過程に関する知識をルールにして入力を自動化する検討もすすめられている[2]。

ここでは、診断型の ES 作成時のプログラミング作業に、事実から判断を下す意志決定部と、事実を集めるための作業手順があることに着目し、それぞれに対して、判断記述表、および手続き記述表を用意して、各フェーズにおける知識の入力・編集作業を容易化することを図っている。

3.1 判断記述表

判断記述表は、判断を下すべき一つ一つの問題項目ごとと一つの表が対応し、(1) 診断を下す問題の名前、(2) 初期症状、(3) その判断を支持する事実、(4) その判断を阻止する事実、(5) 取り得る措置手続きの名前などをエントリに持つ。(図2)ここで、事実のリストには、実際の実行順序を考慮せずに記入することができる。実行時には、判断を支持する事実や判断を阻止する事実を得るために、処理系が手続き記述表を参照することにより、手続き記述表とのリンクが生ずる。

判断記述	空腹である	状況:yes	時刻:t1
初期症状	お腹が痛い	状況:yes	時刻:t2
サポート事実			
	立ち眩みする	状況:yes	時刻:t3
=	t3		t2
否定の事実			
	食べる	主体:猿	対象:バナナ
<	t5		t2
措置手続き			
	食べる	主体:猿	対象:バナナ
			時刻:t4

図 2. 判断記述表

M.A.Musen によれば、知識獲得ツールは、一般に Symbol level、Method-based、および Task-based モデルに分けられる [4]。この分類に従えば、診断を下すべき症状、原因、それをサポートする事実の入力等を宣言的に入力すればよいので、Method-based (それも問題解決手順を明示した) ツールの色彩が強いと言える。

3.2 手続き記述表

手続き記述表は、目標となる事実を入手するために必要な一連の作業手順を記述する表である。入手すべき一つ一つの事実ごとと一つの表が対応し、(1) 事実設定処理 (事実の真偽を設定する)、(2) 表内処理 (テキスト形式の入出力用、ある変数の値を設定するためにユーザに問い合わせる簡単な質問文、ユーザの許される入力データ値、他の表の呼び出し等を規定する)、(3) 外部処理 (表形式で表せない下位のプログラミング言語で記述された処理関数名、関数の返す値、引き数リストなどを規定する)、(4) 条件 (条件を表す 2 項関係を与え、分岐先を行

の名で与える)、(5) 作業停止 (表から抜け出すことを指示する)、(6) プログラムの停止 (異常処理用) 等に対応するエントリを持つ。(図3)従って、この機能は、入力端末が CRT であることを考慮し、KBMS などで採られているフローチャート入力法 [3] を、表で代用したものと云える。この場合、ある事実を入手するための具体的

手続き記述	食べる	主体:猿	対象:バナナ	時刻:t5
1	IF	判断記述	同じ位置	主体:猿
	THEN	do	食べる	主体:猿
		return		対象:バナナ
	ENDIF			
2	get	X	バナナ	X位置
3	get	Y	バナナ	Y位置
4	do	移動する	主体:猿	X位置 Y位置
5	goto	1		

図 3. 手続き記述表

な処理手続きなどを詳細に記述する必要が多いため、プログラムの色彩が濃厚である。しかし、専門家がある具体的な事実を収集するために日常行なっている作業を彼ら自身がまとめた場合も、一般にフローチャート的な記述をしていることが多いので、ここで提供する知識入力環境は専門家にも受け入れられ易いものと考えている。

4 まとめ

診断型の ES 作成時に適した、判断記述と手続き記述を分離した表形式入力提案を行なった。ここで、知識の保守時における人的要因を考え、知識の入力を専門家が、作業の合間に手近の端末を用いて ES の保守が行なえるよう特殊な端末の機能をできるだけ使わない方針でまとめている。

日頃ご指導いただき、KDD 研究所小野所長、浦野次長、山本室長、及び知能処理研究室の皆様、並びに、本稿をまとめる際に L^AT_EX フォーマット作成指導をして頂いた堀田孝男氏に深謝いたします。

参考文献

- [1] 古谷、服部、“KBMS における ES 開発機能”，情報処理学会第 39 回全国大会，平成元年 10 月，3B-8
- [2] 金井，“スプレッド・シートとプロダクションシステム融合時のセル名の役割について”，情報処理学会第 39 回全国大会，平成元年 10 月，3B-4
- [3] 飯田，“KBMS における ES 開発支援機能 - フローチャート形式での知識入力機能 -”，情報処理学会第 39 回全国大会，平成元年 10 月，3B-9
- [4] Musen, M.A., “Conceptual models of interactive knowledge acquisition tools”, Knowledge Acquisition, 1, 73-88, 1989