OUKA
The University of Osaka
Institutional Knowledge Archive

Title	エキスパートシステム構築のための言語と支援ツール
Author(s)	溝口, 理一郎
Citation	大阪大学大型計算機センターニュース. 1985, 58, p. 35-53
Version Type	VoR
URL	https://hdl.handle.net/11094/65658
rights	
Note	

The University of Osaka Institutional Knowledge Archive : OUKA

https://ir.library.osaka-u.ac.jp/

The University of Osaka

エキスパートシステム構築のための言語と支援ツール

大阪大学産業科学研究所　溝　口　理一郎

1. はじめに

人工知能の研究は知識の表現とその利用法の研究であるといっても過言ではない。最近、「知識情報処理」という言葉がしばしば使われているが、この基本的な問題が一般にも広く認識されだしたことを意味しているものと思われる。人工知能用言語としてLISPが発明されて以来、多くのシステムがLISPで書かれてきたが、1970年台以降の新しい人工知能システムはLISPそのもので作るのではなく、LISPで書かれた専用の知識表現言語を用いて作成されるようになった。このことにより、人工知能の研究がより豊かになったことは良く知られている。

一方、初期のエキスパートシステムはLISPと既存の知識表現言語（例えば、プロダクションシステム、意味ネットワーク等）を用いて構築されていた。MYCIN〔3〕やCASNET〔4〕等はその代表的な例である。その後、エキスパートシステム構築の経験を生かして、構築の際に獲得されたKnowhowを抽出して一般化したものがエキスパートシステム構築支援ツールである。本稿では、プロダクションシステムの概説を行なった後、いくつかの代表的なツールを紹介してその特徴を述べると共に、簡単な使用例を示す。

2. プロダクションシステム

プロダクションシステムは基本的には文字列に対する書き換えシステムとして考えられ、次の3つの基本的な構成要素からなる。（図1参照）

(1) working memory(WM): 単なる文字列の集合。

(2) production memory(PM): if A then Bの形式をとるルールの集合で、通常WM内に現われたAをBに置き換える。Aを条件部、Bを実行部と言う。実際のPSでは条件部、実行部共に任意の関数を書くことができるように拡張されていることが多い。

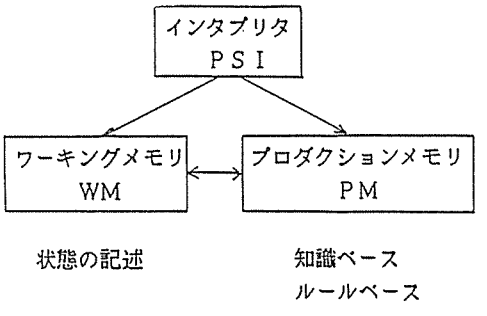

図1　プロダクションシステムのブロック図

(3) production system interpreter(PSI): PMのルールの中からWMに対して条件部がマッチするものを探してきて、WMを実行部通りに書き換える働きをする。

プロダクションシステムには次に挙げる3つの特徴がある。

(1) modularity:PMの中の各々のルール間の相互作用はWMの変更（挿入、削除など）を通じてのみ間接的になされるので、ルールの記述に際して他のルールとの関係をあまり考慮する必要がない。その為、ルールの追加、削除、変更が容易にでき、システムの能力を段階的に増大させて行くことが可能である。

(2) readability:従来のプログラミング手法では、使用される知識の内容が制御の流れの中に含まれている為、どのような知識が如何に使われたかを知る為には、そのプログラムの処理の流れを順次見ていかなければならない。ところが、PM中の各々のルールはそれぞれが一片の完結した知識であるため、その意味の理解が容易である。

(3) self-explanatory:処理の全過程がWMの内容を順次変化させたルールの系列として判るので、如何にして結論が得られたかをシステムが説明することができる。

これらの性質は知識工学にとって本質的なことである。即ち、推論機構としてPSIを採用して、専門知識をプロダクションルールとして表現することにより、知識をある一定の表現形式の中で他の手続きと分離して取り扱うことを可能にする。この意味でプロダクションシステムは知識工学における知識表現ツールとして極めて重要な位置を占めている。

プロダクションシステムにおける推論方式として

・前向き推論

・後向き推論

の2つのものがある。前向き推論では、現在の状況に適した条件部を持つルールが適用されていき、ルールの条件部から実行部の方向へ推論が進むが、後向き推論では、まず達成すべきゴールが定められた後、それを満たす実行部を持つルールが検索され、そのルールの条件部が次のゴールとなり、更にそのゴールを満たす実行部を持つルールが検索される、という逆方向の処理方式がとられる。

3. エキスパートシステム構築支援ツール概観

3.1 エキスパートシステムの特質

1977年第5回人工知能国際会議においてE.A.Feigenbaumが提唱した「知識工学」〔1〕の最も顕著な特徴は、一般的な推論機構にではなく「知識」に問題の本質があることを真正面から主張することにある。その結果、エキスパートシステムは次のような基本的なアーキテクチャ上の特徴を持つ。

① 知識の宣言的な記述

② 知識と推論機構の分離

このことから、タスクに依存する部分は「知識」の部分だけであり、推論機構はタスクに依存しない汎用性のある部分であることがわかる。従って、知識ベースの部分だけを入れ換えれば原理的には別のエキスパートシステムができることになる。実際にはエキスパートシステムを作成する為には多くの作業が必要であり、その中の出来るだけ多くの部分を支援することが望ましいことは言うまでもない。

3.2 エキスパートシステム構築のフェイズ

エキスパートシステムを構築する過程は図2のように5つのフェイズに分けて考えることができる。

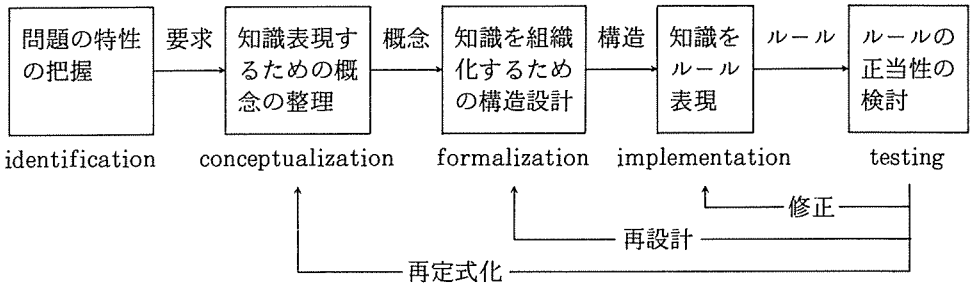

図2 エキスパートシステム構築の過程〔2〕

各操作共に等しく重要であるが、現在のところでは最初の3つは支援することは極めて困難であり、ツールはない。それ故、ここではimplementationとtestingのフェイズについて考察する。あるタスクの専門家から抽出された知識をルール表現して知識ベースを構成し、それを完成させるまでには次の2つのフェイズが必要である。

1. 知識のルール表現
   - 自然言語 ⟶ ルールへの変換
   - 冗長性の除去、矛盾の検出
2. 知識のデバッグ
   - ルールの自然言語変換
   - 説明機能

     why：何故そのような質問をするのかを説明する。

     how：いかにしてあるゴールを達成したかを説明する。

エキスパートシステム構築支援ツールは、通常上記の各フェイズを支援する機能を持つことが望まれる。もちろん、推論機関は持っている。

3.3 概観

現在幾つかのツールが開発されているが、そのほとんどが実際に構築したエキスパートシステムを汎用化したものである。従って、汎用といっても対象とされたタスクの問題の性質をある程度反映したものとなっている。表1に支援ツールの幾つかの例を示す。

表1. 支援ツールの一覧

	元になったエキスパートシステム	知識表現	制御構造	確信度	説明機能	自然言語
EMYCIN〔5〕（KS300）	MYCIN〔3〕感染症における抗生物質投与支援	ルール	後向き推論	有	有	有
EXPERT〔6〕	CASNET〔4〕緑内障の診断	FF, FH, HHルール	前向推論	有	有	有
HEARSAY-III〔7〕AGE〔8〕	HEARSAY-II〔9〕音声理解システム	発火パターン付き手続き, モジュール化	黒板モデル, 豊富	無	無	無
KAS〔10〕	PROSPECTOR〔11〕鉱脈探査	ルール, ネットワーク	前向き, 後向き推論	有	有	有
KEE〔12〕（UNIT PACKAGE〔14〕）	MOLGEN〔13〕	フレーム, ルール オブジェクト指向	前向き, 後向き推論, メソッド	無	有	無
ROSIE〔15〕	汎用言語	ルール, モジュール化	状態駆動, 変化駆動, 後向き推論	無	有	有
OPS5〔16〕	汎用言語	ルール	前向き推論	無	無	無
LOOPS〔24〕	汎用言語	ルール, オブジェクト指向, データ指向	メソッド, 前向き推論	無	無	無

3.4 EMYCIN〔3〕〔23〕

EMYCINは比較的小規模な問題を取り扱ったMYCINシステムの汎用化ツールであるため、プロダクションルールの後向き推論という限定された制御構造を持っている。又、その制御構造及び知識表現の単純さ故、説明機能や自然言語インタフェース等に優れているという特徴を有する。持っておくべき機能のバランスがとれているという意味でEMYCINはEXPERTと並んでエキスパートシステム構築支援ツールの典型と言うことができる。

EMYCINはよく利用されており、その実績は豊富である。中でも、PUFF〔17〕、肺機能検査結果の解釈システム、SACON〔18〕、有限要素法による構造解析プログラムの利用法の助言システム、はEMYCINで作成されたシステムとして著名である。DEC 20, XEROX 1100 SIPの上で稼働している。

3.5 EXPERT

EXPERTの特徴はルールが次の3種類に分かれているところにある。

① FFルール：所見（FINDINGS）から新たな所見を推論する

② FHルール：所見から仮説（HYPOTHESES、病理状態）を推論する

③ HHルール：仮説から新たな仮説を推論する

CASNETと呼ばれる緑内障の診断エキスパートシステムから汎用化されたツールである。FORTRANで実現されているところも特徴的である。CASNETはCausal-ASsociational NET workから名付けられたことからも分かるように、意味ネットワーク表現を用いて作られた。その後、汎用ツール化される時に、プロダクションルール表現に変更された為、制御構造が少し固い点がある。その代わり、ルールのコンパイルをする機能があり、効率の点では優れている。

3.6 HEARSAY-III

HEARSAY-IIIは音声理解という極めて困難な問題を扱った経験に基づくものであることから、汎用ツールの中では最も強力な制御構造を持っている。その反面、知識の入力、説明機能等の点において多少機能が劣っている。BLACKBOARDと呼ばれる階層構造を持つデータベースとその状態を監視している知識源（KS）とからなるモデルが中心になっているが、BLACKBOARDが更に、ドメインBLACKBOARDとスケジューリングBLACKBOARDの2つに分かれており、柔軟性に富んだ制御を行うことができるようになっている。

3.7 KEE

MOLGENと呼ばれる分子遺伝学の実験計画の支援システムを作成するために作られたUN

IT PACKAGEというツールから発展したものである。顕著な特徴として、ルールだけでなくフレーム型（オブジェクト指向）の知識表現を採用している点が挙げられる。ルールによる推論に加えて、フレームに付加された手続き（attached procedure）による推論も可能になっている。XEROX 1100SIP上で稼働し、リスプマシンの優れたマン・マシンインタフェースを生かしたツールとなっている。

3.8 ROSIE, OPS5, LOOPS

これらの言語は元になるエキスパートシステムはなく、汎用のプログラミング言語である。ROSIEはルールをかなり英語に近い形で入力できる点が優れている。又、OPS5はカーネギーメロン大学で開発された高速なプロダクションシステムとして著名である。DEC社は「知識ネットワークプロジェクト」と称して、VAX11の販売から保守までの多様な作業を支援する一連のエキスパートシステムの構築を計画している。その一環としてカーネギーメロン大学で開発されたVAX11のシステム構成を行うエキスパートシステムR1〔20〕（後にXCONと改称）はOPS5で構築され、現在では実用のレベルにまで達していることで知られている。又、IBMは改良したOPS5を用いて、自社の大型機のオペレータの仕事を代行するエキスパートシステムYES／MVS〔21〕を作成している。これは大型機のパフォーマンスをモニタリングして、各種のパラメータを動的に調整し、効率の良い運用を行うものである。OPS5はVAXとSYMBOLICS 3600上で稼働している。LOOPSはInterlispをベースにした、XEROX社で開発された言語である。オブジェクト指向、データ指向、ルール指向、及び通常の手続き指向の4つのプログラミングパラダイムを組み合わせた新しい言語の思想を具体化したもので、XEROXにおいてVLSIの設計に用いられている。もちろん1100SIP上で稼働する。

3.9 その他日本で利用可能なツール

(1) BRAINS〔22〕

EXPERTを元にEMYCINの確信度と後向き推論の機能を追加したツール。

IBM、富士通、日立のUTILISP版

VAXのFRANZ LISP版

SYMBOLICS 3600版

がある。

(2) ESHELL

AGEを参考にして作られたツール。

富士通のUTILISP版、LISPマシンα版がある。

(3) MECS-AI〔19〕

東大で開発された、EMYCINに患者の履歴を扱えるように拡張したツールである。

(4) ART

INFERENCE社によって開発されたツール。機能は豊富であり、VIEWPOINTと呼ばれる推論の道筋を複数個持つことができ、それを動的に切り替えるところが特徴的である。

4. ツールの具体例

4.1 ツールの選択

1）知識表現

専門知識の内、手続き的なものに関しては、現状ではプロダクションルール表現がほとんど全てであるので選択の余地はない。しかし、手続き的な知識以外のもの、例えば対象の構造を表わす様な際には、その表現をサポートする機能が必要となる。

2）制御構造

前向き推論と後向き推論とではかなり対象とする問題の性格が異なる。故障診断のように、故障の現象が分かっていてその原因を因果関係の連鎖をたどって発見するような問題には後向き推論が適している。又、解釈型、設計型、制御型等のような明確なゴールが得られていない問題には前向き推論が適している。利用者が制御構造の変更をできるような機能が必要な場合もあろう。このような場合には、EMYCINやEXPERTのように制御が固定されたものは適さず、HEARSAY-IIIのような汎用的なツールが適している。更に、既存のソフトウェアとのリンク、即ち他言語との結合及びOSの機能の利用のしやすさ等も重要な要素となる。

3）インタフェース、ファシリティ

知識（ルール）の記述の容易さ、知識ベースの維持、管理の機能などが豊富であることが望ましい。

4.2 EMYCINの概略〔5〕〔23〕

全てのツールについて詳細を述べることは紙面の都合で行なえないので、ここでは最も代表的なEMYCINの詳細を紹介する。

図3にEMYCINのブロック図を示す。シス

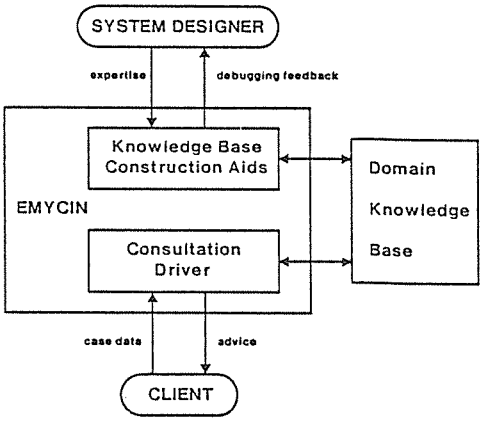

図3　EMYCINのブロック図

テムの作成者と(作られた)システムの利用者との2種類の利用者がいることになる。

4.2.1 EMYCINの構成要素

EMYCINには次の3つの構成要素があり、ルールだけでなく、コンテクストタイプとそのパラメータ及び両者の属性を与えることによってエキスパートシステムが構成される。

1）コンテクストタイプ（一種のフレーム）

例えば、patient, culture（培養検体）のような診断対象のオブジェクトを表わす。コンテクストは階層構造をなしているが、図4にMYCINにおけるコンテクスト木を示す。各コンテクストタイプはパラメータを持ち、それらの値を求める操作が診断の主な操作となる。又、コンテクストタイプは属性を持ち、次のように用いられる。

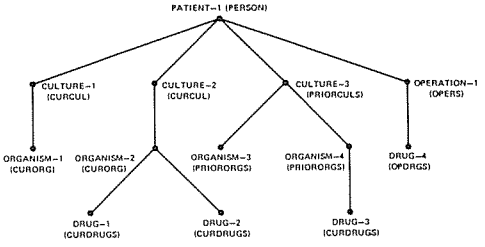

図4　コンテクストタイプの木の例

- パラメータ ⟶ 事実及び推論結果の格納
- 属性 ⟶ 対話の流れの大枠の決定
  自然言語インタフェース（英語への変換）

2）パラメータ

コンテクストタイプと同様に属性を持ち、次の用途に用いられる。

- 属性 ⟶ 対話インタフェース、値のチェック

3）ルール（専門知識）

- 属性 ⟶ 条件部、実行部、コメント、正当性の格納

4.2.2 曖昧推論

データをレコード表現した例を以下に示す。

パラメータ		コンテクスト	値	(CF)
IDENTITY	OF	ORGANISM-1	PSEUDOMONAS	(.8)
IDENTITY	OF	ORGANISM-1	E.COLI	(.15)

これに対応するEMYCINのルール上での表現は、例えば

(SAME CNTXT IDENTITY PSEUDOMONAS)

となり、値としてCF値（Certainty Factor）0.8が返される。

ＣＦ値の計算

確信度は次の要領で計算される。

- 条件部の $AND は各要素のＣＦ値の最小値
- ルールの結論は

  条件部のＣＦ値×結論部の係数

- 同一結論の取り扱い（or ノード）

CFCOMBINE(CF1,CF2) ＝

CF1＋CF2－CF1*CF2	CF1, CF2＞0
(CF1＋CF2)／(1－MIN(|CF1|,|CF2|)	CF1*CF2＜0
－CFCOMBINE(－CF1,－CF2)	CF1, CF2＜0

このようにEMYCINでは確信度（CF）と呼ばれるデータの信頼性を表わす尺度が導入されており、曖昧なデータと知識に基づく推論ができるようになっている。

4.2.3　コンテクストタイプの属性の例

INITIALDATA	このタイプのインスタンスが生成された時に利用者に質問して値を求めるパラメータのリスト、ASKFIRSTパラメータの部分集合
GOALS	このタイプのインスタンスが生成された時に現在の作業を中断してこのパラメータを推論する。
TRANS	ルールや質問の中でのコンテクスト（例えばculture）の翻訳 ex.（the positive culture of＊）ここで、＊は1レベル上のコンテクストタイプを示す。
PROMPT1ST	このコンテクストタイプのインスタンスが存在するかどうかを質問する時のプロンプト ex.（Is *currently receiving therapy with any antimicrobial agent？）
PROMPT2ND	2つ目以降のインスタンスを質問する時のプロンプト
ASSOCWITH	コンテクストタイプの親タイプ
OFFSPRING	コンテクストタイプの子タイプ

4.2.4　パラメータの属性の例

ASKFIRST	MYCINではLABDATAと呼ばれていた。推論する前に先ず値を聞く。利用者が答えられなければ推論を試みる。これが“T”でないパラメータは先に推論して失敗してから利用者に聞く。GOAL属性を持つパラメー

タは絶対に聞かない。

PROMPT	質問するときのプロンプト。これがなければ質問しない。
TRANS	パラメータの翻訳
EXPECT	予想されるパラメータの値のリスト、or次の値
	NIL　YES／NO
	NUMB　数値
	DATE　日付
MULTIVALUED	T：1つずつ聞く
	ASK-ALL：1度に全部聞く。

```
POSCUL [CONTEXTTYPES]
TRANS:  (positive cultures obtained from *)
PARMGROUP:  PROP-CUL
PRINTID:  CULTURE-
INITIALDATA:  (SITE WHENCUL COLLECT)
GOALS: (INFECTION)
PROMPT1ST:  (Are there any cultures for * which may be related to the
               present illness, and from which organisms have been
               grown successfully in the microbiology laboratory?)
PROMPT2ND:  (Are there any other cultures from which organisms have
               been successfully isolated in the laboratory?)
ASSOCWITH:  (PATIENT)
OFFSPRING:  (KNOWNORG)
```

図5　コンテクストタイプの属性記述の例

```
AGE  a single valued parameter of a PATIENT context
  TRANS:  (the age of *)
  PROMPT:  (Age:)
  ASKFIRST:  T
  EXPECT:  NUMB
  RANGE:  (0 100 CONFIRM Is the patient really VALU years old?)
  UNITS:  YEARS

GRAM  a parameter of an ORGANISM context
  TRANS:  (the stain of *)
  PROMPT:  (The stain (Gram, India ink, or Ziehl-Neelsen acid-fast)
        of * :)
  REPROMPT:  (There are three principal stains used for examining smears
          of organisms.  Most bacteria stain Gram positive or
          Gram negative.  Mycobacteria are typified by acid-fast
          Ziel-Neelsen staining.  Cryptococcus, a fungus, is
          demonstrated by a positive India ink stain.)
  ASKFIRST:  T
  EXPECT:  (GRAMNEG GRAMPOS ACIDFAST INDIAINK)
  PROPERNOUN:  T
```

図6　パラメータの属性記述の例

RANGE　　値の範囲

(0 100 CONFIRM Is the patient really VALU years old？) ここで、VALUには利用者が答えた値が入っている。

図5－7にコンテクストタイプ、パラメータ、ルールの例を示す。

If: 1) The infection which requires therapy is meningitis, and
2) The type of the infection is fungal, and
3) The patient has been to an area that is endemic for coccidiomycoses, and
4) The race of the patient is one of: black asian indian
Then: There is suggestive evidence that coccidiomycoses is one of the organisms which might be causing the infection.

もし, 1) 治療を要する感染症が髄膜炎であって,
2) 感染症のタイプが真菌であって,
3) 患者がコクシディオミコーシィス（真菌の一種）にあたる風土病の地域にいたことがあって,
4) 患者は黒色, アジアインド系人種であるとすると,
そのとき, コクシディオミコーシィスが感染症を起した原因菌の有力な所見となる.

図7　ＭＹＣＩＮにおけるルールの例

4.2.5　ルール

ルールの実行部の後にあるCF値が条件部で求められたCF値と掛け合せられ、結論のCF値が求められる。ARL(Abbreviated Rule Language)と呼ばれるルール入力用の簡易言語とルールを英語表現に戻す機能があり、良いインタフェースを実現している。各ルールは実行部で言及しているパラメータごとにまとめて記憶される。処理の初めにコンテクスト木の根に相当するコンテクストが生成され、INITIAL DATA属性を持つパラメータの値を質問した後、それのゴール属性を持つパラメータを推論するルールが検索される。次に、検索されたルールの条件部で言及しているパラメータを求めることを目的として、それがASKFIRST属性を持っていれば利用者に問い合わせ、そうでなければ、それをゴール部に持つルールを検索するというように処理が続けられる（後向き推論)。

4.2.6　EMYCINの評価〔23〕

SACON, CLOTの作成を通して行なった、EMYCINの評価結果を紹介する。

・SACON （1978）建築物の構造解析用有限要素法プログラムの使用のコンサルテーション

ルール数	１７０	
パラメータ数	１４０	
期間　タスクと知識ベースの定式化		２人月
インプリメントとテスト		２人月
最初のルールの作成とインプリメント		４Ｈ／ルール
追加のルールの作成とインプリメント		２Ｈ／ルール

・CLOT （1980）出血性の傷の存在とそのタイプの同定のコンサルテーション

ルール数	６３
パラメータ数	５０

期間　必要な概念や戦略の整理　　1日

ルールの作成とインプリメント　2日

このように、ENYCINを用いることによって、極めて短期間にエキスパートシステムが構築されることが分かる。もちろん、これはEMYCINの専門家がEMYCINに適した問題を選択して構築したものであるが、それにしてもその有用さには目を見張るものがある。

4.2.7　使用例

以下に簡単な使用例を示す。ルールが一つしかない簡単な知識ベースであるが、基本的な操作は含まれている。右側に注があるので説明は省略する。

```
@<EMYCIN>EMYCIN                                    [Run EMYCIN from the appropriate
                                                    directory.]

EMYCIN 12-Dec-80 . . .

Do you wish to create a new knowledge base?  Yes
Enter a word or phrase to describe your domain, i.e. fill in the blank in
      "this program performs a --- consultation". (e.g.
      infectious disease, structural analysis).
** business problem
Enter a one-word name for the root of your context tree, the central "object" with which the consultation is
      concerned (e.g. PATIENT, STRUCTURE):
** COMPANY                                         [The "root" of the context tree]
Please enter a word or phrase as "translation" of COMPANY:
** company--Okay?  Yes                            [EMYCIN makes the obvious
                                                    suggestion, we accept it. The
                                                    translation might have been fancier,
                                                    say "municipal agency."]

And the plural form is:
** companys--Okay?  No                            [EMYCIN doesn't know some
                                                    spelling rules.]
** companies                                       [So spell it right.]
Parameter group: COMPANY-PARMS--Okay?  Yes
COMPANY
PRINTID: COMPANY--Okay?  Yes                       [i.e., there will be a COMPANY-1,
                                                    COMPANY-2, in consultations]
Enter a list of parameters which will always be asked to the user when a COMPANY is created:
COMPANY
INTIALDATA: (CNAME PROBLEM EMPLOYEES)
And now a list of parameters to infer whenever a COMPANY is created:
COMPANY
GOALS: (TOOLS)
Do you want the program to automatically display the values concluded for some or all of the GOALS?  Yes
                                                   [Automatically print the values of
                                                    TOOLS at end of each consultation.]
```

Now please define each of the parameters listed above. Each is a member
of COMPANY-PARMS.
What kind of parm is CNAME--Yes/No, Singlevalued, Multivalued, or
Ask-All? **Singlevalued**
CNAME
EXPECT: **ANY**
CNAME
TRANS: **(THE NAME OF *)**
CNAME
PROMPT: **(WHAT COMPANY IS HAVING A PROBLEM?)**

What kind of parm is PROBLEM--Yes/No, Singlevalued,
Multivalued, or Ask-All? **Singlevalued**
PROBLEM
EXPECT: **(PAYROLL INVENTORY)**
PROBLEM
TRANS: **(THE TYPE OF PROBLEM)**
PROBLEM
PROMPT: **(IS THE PROBLEM WITH PAYROLL OR INVENTORY?)**

What kind of parm is EMPLOYEES--Yes/No, Singlevalued, Multivalued, Ask-All? **Singlevalued**
EMPLOYEES
EXPECT: **POSNUMB**
EMPLOYEES
UNITS: **<cr>**
EMPLOYEES
RANGE: **<cr>**
EMPLOYEES
TRANS: **(THE NUMBER OF EMPLOYEES OF *)**
EMPLOYEES
PROMPT: **(HOW MANY PEOPLE DOES * EMPLOY?)**

[<cr> here gives the property a value of NIL.]

What kind of parm is TOOLS--Yes/No, Singlevalued, Multivalued, or
Ask-All? **Multivalued**
TOOLS
LEGALVALS: **TEXT**

TOOLS
TRANS: **(THE TOOLS TO USE IN SOLVING THE PROBLEM)**

[Values produced by CONCLUDETEXT, the results to be printed.]

Okay, now back to COMPANY . . .

COMPANY
SYN: (((CNAME) (CNAME)))--Yes, No, or Edit? **Yes**
Creating rule group COMPANYRULES to apply to COMPANY contexts . . .

[Now that we've defined those parms, finish defining the context type.]

[The company name (CNAME) will be used to translate a COMPANY.]

. . Autosave . .
Please give a one-word identifier for your knowledge base files:
** **BUSINESS**
<EMYCIN>CHANGES.BUSINESS;1
Are there any descendants of COMPANY in the context tree? **No**

Rules, Parms, Go, etc.? **Rules**
Author of any new rules, if not yourself: <**cr**>
Will you be entering any of the rule information properties? **No**

[This is asked upon the first entrance to the rule editor.]

Rule# or NEW: **NEW**

[Now enter rules to deduce each of the GOALS defined above; in this case, just TOOLS.]

RULE001
PREMISE: **(PROBLEM = PAYROLL AND EMPLOYEES > 1000)**
RULE001
ACTION: **(TOOLS = "a large computer")**

Translate, No further change, or prop name: **TRANSLATE**

RULE001

[This rule applies to companies, and is tried in order to find out
about the tools to use in solving the problem]
If: 1) The type of problem is payroll, and
2) The number of employees of the company is greater than 1000
Then: It is definite (1.0) that the following is one of the tools to use in solving the problem: a large computer

Translate, No further change, or prop name: <**cr**>

Rule# or NEW: <**cr**>

[Finished entering rules.]

Rules, Parms, Go, etc.? **Save**
<EMYCIN>CHANGES.BUSINESS;2

[Save the knowledge base.]

Rules, Parms, Go, etc.? **Go**

[Run a consultation to test the knowledge base.]

Special options (type ? for help):
** <**cr**>

[No options needed.]

20-Oct-79 14:16:48

--------COMPANY-1--------
1) What company is having a problem?
** **IBM**
2) Is the problem with payroll or inventory?
** **PAYROLL**
3) What is the number of employees of ibm?
** **10000000**
Conclusions: the tools to use in solving the problem are as follows: a large computer.

Enter Debug/review, Rules, Parms, Go, etc.? **Parameters**

Parameter name: **cname**
Property: **PROPERNOUN**
CNAME
PROPERNOUN: **T**
Property: <**cr**>

[A small parameter change—we noted that ibm was not capitalized. Setting the PROPERNOUN property will fix the problem.]

Parameter name: <**cr**>

[Finished entering parameters.]

```
Rules, Parms, Go, etc.? Save                    [Save these changes to the
                                                 knowledge base.]
<EMYCIN>CHANGES.BUSINESS;3

Rules, Parms, Go, etc.? Quit
@
.
.
.
                                                [Sometime later...]
@<EMYCIN>EMYCIN
EMYCIN 12-DEC-80 . . .

Hi.

Should I load <EMYCIN>CHANGES.BUSINESS;3?  Yes
File created 25-Sep-81 10:49:24
CHANGESCOMS
(<EMYCIN>CHANGES.BUSINESS;3)

Do you want to enter Rules, Parms, Go, etc. (? for help)? New consultation
  [confirm] <cr>
Special options (type ? for help):
** <cr>

23-Feb-91 10:28:37

  --------COMPANY-1--------
  1) What company is having a problem?
  ** STANFORD
  2) Is the problem with payroll or inventory?
  ** INVENTORY
  3) How many people does Stanford employ?
  ** 10000

I was unable to make any conclusion about the tools to use in solving
     the problem.                               [No rules have yet been entered for
                                                 making conclusions about inventory
                                                 problems.]

Enter Debug/review phase, or other option (? for help)? Quit
```

4.2.8　説明機能

エキスパートシステムが自分が出した結論に関する説明をすることができるということは、「自意識」を持つことを意味しており、システムの設計者にはデバッグに、利用者には結論を納得するのに極めて有効である。WHY と HOW コマンドによる説明の例を以下に示す。説明機能についての詳細は次号に譲る。

このような、通常汚染されていない部位に菌1が侵入した経路として可能性のあるものは何か？

（医師が説明を求める）

＊＊WHY

（そこでTEIRESIASがその“推論過程”を説明する）

〔つまり、なぜ菌1がこの部位に侵入した経路を決定することが重要なのか〕

〔3.0〕…治療が対象とすべき菌の種類を明らかにするために以下のことがすでに成立している

〔3.1〕…検体の採取部位は血液である

〔3.2〕…菌のグラム染色性は陽性である

したがって、もし

〔3.3〕感染経路が消化管であり

〔3.4〕〔腹部が感染部位であるかまたは骨盤部が感染部位である〕

ならば、そのとき

治療の対象とすべき菌が腸内細菌層に属すると思われる強い根拠がある（確実度0.9）

〔規則 095〕

＊＊HOW〔3.4〕

注：〔3.4〕は上記の例の中で興味の持たれている条件節の番号である。

〔つまり、どのようにして腹部が感染部位であるかあるいは骨盤部が感染部位であるかを決定するつもりか〕

〔4.0〕この場合には規則021が用いられる。

（もしも医師が規則021を見たいと思うならば、TEIRESIASにその規則がどのように用いられるかを、規則に言及している応答につけられた番号を指定して、HOWコマンドを用いて質問することができる。）

＊＊HOW〔4.0〕

〔つまり、規則021はどのように用いられるか〕

〔5.0〕次のことがすでに成立している

〔5.1〕検体は最近のものである

それゆえ、もしも

〔5.2〕この菌に関連があると考えられるような、治療上重要な疾患が考えられる

ならば、そのとき

検体の採取部位が患者の感染部位であることが確実である（確実度1.0）

5. むすび

1970年代前半におけるエキスパートシステム構築の経験を生かして70年代の後半にはエキスパートシステム構築のための汎用ツールが作成された。現在は大学で開発された汎用ツールが商用化され、それを用いたエキスパートシステムの構築が行われている。一方、研究レベルではより強力なツールの開発が行われている。上に見たように、現在のツールはプロダクションルールによる専門知識の表現を基本としている。ところが、プロダクションルールで記述される知識は自ずから限定されたものであり、その適用範囲には制限がある。EMYCINに代表されるツールはいわゆる診断型のツールであり、基本的には（故障の原因の）分類を行っているということができる。EMYCINのルールはそのような分類を行うためのヒューリスティックスを表現したものとなっている。しかしながら、そのようなツールではうまく扱えないような問題が現実には沢山ある。現在次の2つの大きな問題があると考えられている。

1）故障診断において対象の構造と動作原理に関する情報を有効に利用すること。

2）分類以外の問題、例えば設計型、教育型等の問題に適用できるツールの開発。

知識工学とはプロダクションシステムで表現できる問題を探して、その問題を解決するエキスパートシステムを構築することではない。知識工学とは人間のエキスパートが持つ「専門知識」をできるだけ宣言的に記述しそれを利用することを通して、「知識」を体系化することを目指す学問である。現在、プロダクションルールでかなりの知識が書けることは明らかになった。しかし、同時にそれだけでは不十分であることも分かった。他の多くの問題領域においても有効な知識の表現法の開発とそれを汎用化し、ツールとして発展させることは知識工学の研究における中心的な問題となるものと思われる。

＜参考文献＞

1）E.A.Feigenbaum:"The art of artificial intelligence: I. Themes and case studies of knowledge engineering", Proc. of the 5th IJCAI, pp. 1014−1029, 1977.

2）F.Hayes−Roth et al.:"Building Expert Systems", Addison−Wesley, 1983.

3）神沼他訳："診療コンピュータシステム", 文光堂, （MYCINの翻訳）1981.

4）S.M.Weiss et al.:"A model−based method for computer−aided medical decision−making", Artificial Intelligence, Vol. 11, pp.145−172, 1978.

5）W.V.Melle et al.:"The EMYCIN manual", Dept. of Stanford Univ., STAN−CS−81−885（HPP−81−16）1981.

6）S.M.Weiss et al.:"A guide to the use of the EXPERT consultation system", Computers in Biomedicine, Dpet. of Comp. Science, Rutgers Univ., 1978.

7) L.D.Erman et al.: "The design and an example use of HEARSAY-III", Proc of of the 7th IJCAI, pp.409-415, 1981.

8) N.Aiello et al.: "AGE reference manual", Heuristic Programming Project, Computer Science Dept., Stanford Univ. 1981.

9) L.D.Erman et al: "The HEARSAY-II speech understanding system: Integrating knowledge to resolve uncertainty", Computing Surveys, Vol. 12, 2, pp.213-254, 1980, 又は中島訳: "HEARSAY-II 音声理解システム: 不確定性の解決の為の知識の統合"、bit, 別冊11, pp.53-91, 1981.

10) R.Reboh et al.: "Knowledge Engineering techniques and tools in the PROSPECTOR environment", SRI International, Tech. Report 243, 1981.

11) R.Duda et al.: "Model design in the PROSPECTOR consultant system for mineral exploration", in Expert Systemes in the Micro Electronic Age, pp. 153-167, 1979.

12) IntelliGenetics: "The Intelligenetics KEE: A Knowledge Engineering Environment" 1983.

13) M.Stefic: "Planning with constraints (MOLGEN: Part1) ", "Planning and meta-planning (MOLGEN: Part2) ", Artificial Intelligence, Vol. 16, pp.111-169 1981.

14) M.Stefic: "An examination of a frame-structured representation system", Proc. of the 6th IJCAI, pp.845-852, 1979.

15) F.Hayes-Roth et al.: "Rationale and motivation for ROSIE", Tech. Note N-1648-ARPA, Rnad Corp., Santa Monica Monica, 1981.

16) C.L.Forgy: "OPS5 user's manual", Carnegie-Mellon Univ., CMU-CS-81-135 1981.

17) J.C.Kunz et al,: "A physiological rule-based system for interpreting pulmonary function test rules", Stanford Univ., HPP-81-19, 1978.

18) J.S.Bennett et al,: "SACON: A knowledge-based consultant for structural analysis", Proc. of the 6th IJCAI, pp.47-49, 1979.

19) 開原他: "MECS-AI ユーザー・マニュアル", 東大病院情報処理部 1982.

20) J.McDermott: "R1: A rule-based configurer of computer systems", Artificial Intelligence, Vol. 19, pp.39-88, 1982.

21) J.H.Griesmer et al.: "YES/MVS: A continuous real time expert system", Proc. of the AAAI-84, 1984.

22) 東洋情報システム: "推論システム BRAINS 利用者マニュアル Version 1.1".

23) B.G.Buchanan et al.: "Rule-based expert systems: The MYCIN experiments of the Stanford Heuristic Programming Project", Addison-Wesley, 1984.

24) D.G.Bobrow et al.: "The LOOPS manual", Xerox PARC, Memo KB-VLSI-81-13, 1983. M.Stefik et al.: "Rule-oriented programming in Loops", Xerox PARC, Memo KB-VLSI-82-22, 1983.