

PART FIVE

Generalizing MYCIN

14

Use of the MYCIN Inference Engine

One of the reasons for undertaking the original MYCIN experiment was to test the hypothesis that domain-specific knowledge could successfully be kept separate from the inference procedures. We felt we had done just that in the original implementation; specifically, we believed that knowledge of a new domain, when encoded in rules, could be substituted for MYCIN's knowledge of infectious diseases and that no changes to the inference procedures were required to produce MYCIN-like consultations. In the fall of 1974 Bill van Melle began to investigate our claim seriously. He wrote (van Melle, 1974):

The MYCIN program for infectious disease diagnosis claims to be general. One ought to be able to take out the clinical knowledge and plug in knowledge about some other domain. The domain we had in mind was the diagnosis of failures in machines. We had available a 1975 *Pontiac Service Manual*, containing a wealth of diagnostic information, mostly in decision tree form, with branching on the results of specific mechanical tests. Since MYCIN's rule base can be viewed as an implicit decision tree, with judgments based on laboratory test results, it at least seemed plausible that rules could be written to represent these diagnostic procedures. Because of the need to understand a system in order to write rules for diagnosing it, a fairly simple system, the horn circuit, was selected for investigation.

After some consideration, van Melle decided that the problem required only a degenerate context tree, with "the horn" as the only context, and that all relevant rules in the Pontiac manual could be written as definitional rules with no uncertainty. Two rules of his fifteen-rule system are shown in Figure 14-1.

Much of MYCIN's elaborate mechanism for gathering and weighing evidence was unnecessary for this simple problem. Nevertheless, the project provided support for our belief that MYCIN's diagnostic procedures

RULE002

IF: 1) The horn is inoperative is a symptom of the horn, and
 2) The relay does click when the horn button is depressed, and
 3) The test lamp does not light when one end is grounded and
 the other connected to the green wire terminal of the relay
 while the horn button is depressed
 THEN: It is definite (1.0) that a diagnosis of the horn is
 replace the relay

[HORNRULES]

RULE003

IF: 1) The horn is inoperative is a symptom of the horn, and
 2) The relay does not click when the horn button is depressed, and
 3) The test lamp does light when one end is grounded and the
 other is touched to the black wire terminal of the relay
 THEN: It is definite (1.0) that there is an open between the
 black wire terminal of the relay and ground

[HORNRULES]

FIGURE 14-1 English versions of two rules from the first nonmedical knowledge base for EMYCIN.

were general enough to allow substitutions of new knowledge bases.¹ As a result, we began the project described in Chapter 15, under the name EMYCIN.² In Chapter 16 we describe two applications of EMYCIN and discuss the extent to which building those two systems was easier because of the framework provided. Remember, too, that the MYCIN system itself was successfully reimplemented as another instantiation of EMYCIN.

The flexibility needed by MYCIN to extend or modify its knowledge base was exploited in EMYCIN. Neither the syntax of rules nor the basic ideas underlying the context tree and inference mechanism were changed. The main components of an EMYCIN consultation system are described in Chapter 5, specifically for the original MYCIN program. These are as follows:

¹It also revealed several places in the code where shortcuts had been taken in keeping medical knowledge separate. For example, the term *organism* was used in the code occasionally as being synonymous with *cause*.

²We are indebted to Joshua Lederberg for suggesting the phrase Essential MYCIN, i.e., MYCIN stripped of its domain knowledge. EMYCIN is written in Interlisp, a programming environment for a particular dialect of the LISP language, and runs on a DEC PDP-10 or -20 under the TENEX or TOPS20 operating systems. The current implementation of EMYCIN uses about 45K words of resident memory and an additional 80K of swapped code space. The version of Interlisp in which it is embedded occupies about 130K of resident memory, leaving approximately 80K free for the domain knowledge base and the dynamic data structures built up during a consultation. A manual detailing the operation of the system for the prospective system designer is available (van Melle et al., 1981).

Contexts	Objects of interest, organized hierarchically in a tree, called the context tree
Parameters	The attributes of objects about which the system reasons
Rules	Associations among object-attribute-value triples

While these concepts were generalized and access to them made simpler, they are much the same in EMYCIN as they were in the original system.

The major conceptual shift in generalizing MYCIN to EMYCIN was to focus primarily on the persons who build new systems rather than on the persons who use them. Much of the interface to users remains unchanged. The interface to system builders, however, became easier and more transparent. We were attempting to reduce the time it takes to create an expert system by reducing the effort of a knowledge engineer in helping an expert. As discussed in Chapter 16, we believe the experiment was successful in this respect.

Much of the TEIRESIAS system (discussed in Chapter 9) has been incorporated in EMYCIN. Thus the debugging facilities are very similar. In addition, EMYCIN allows rules to be entered in the Abbreviated Rule Language, called ARL, that simplifies the expression of new relations. For example, the rule premise

```
($AND (SAME CNTXT SITE BLOOD)
      (GREATERP* (VAL1 CNTXT SICKDEGREE) 2)
      ($OR (NOTSAME CNTXT BURNED)
           (LESSERQ* (PLUS (VAL1 CNTXT NUMCULS)
                           (VAL1 CNTXT NUMPOS))
                    3)))
```

might have been entered as either

```
(SITE = BLOOD AND
 SICKDEGREE > 2 AND
 ~BURNED OR NUMCULS + NUMPOS LE 3)
```

or

```
(SITE = BLOOD, SICKDEGREE > 2, ~BURNED OR NUMCULS + NUMPOS LE 3)
```

In the sample EMYCIN dialogue shown in Figure 14-2, EMYCIN can be seen to ask about contexts, parameters, and rules for a hypothetical consultation system about business problems. EMYCIN keeps track of the properties, such as TRANS (for the appropriate English translation of the concept name), and does the bookkeeping necessary to create a new knowledge base. Then it helps the user debug the knowledge base in the context of sample cases.

```

@<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.]
                                                         [So spell it right.]

** companies
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
INITIALDATA: (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)

```

FIGURE 14-2 Sample dialogue with EMYCIN to create a new consultation program for business problems. (User's input is in boldface, and <cr> indicates that the user typed a carriage return. Comments are in italics.) [This sample is taken from *The EMYCIN Manual* (van Melle et al., 1981).]

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>

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

EMPLOYEES

RANGE: <cr>

EMPLOYEES

TRANS: (THE NUMBER OF EMPLOYEES OF *)

EMPLOYEES

PROMPT: (HOW MANY PEOPLE DOES * EMPLOY?)

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

TOOLS

LEGALVALS: **TEXT**

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

TOOLS

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

Okay, now back to COMPANY . . .

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

COMPANY

SYN: (((CNAME) (CNAME)))--Yes, No, or Edit? **Yes**

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

Creating rule group COMPANYSRULES to apply to COMPANY contexts . . .

. . 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, Parm, 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** *[Save the knowledge base.]*

<EMYCIN>CHANGES.BUSINESS;2

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?
** **1000000**

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** *[A small parameter change—we noted that ibm was not capitalized. Setting the PROPERNOUN property will fix the problem.]*
Property: **PROPERNOUN**
CNAME
PROPERNOUN: T
Property: <cr>

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

FIGURE 14-2 continued

(<EMYCIN>CHANGES.BUSINESS;3)

Do you want to enter Rules, Params, 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**

FIGURE 14-2 continued