Report 79-10
Stanford -- KSL

Prototypes and Production Rules: An Approach to Knowledge Representation for Hypothesis Formation.

Janice S. Aikins, Jul 1979

Stanford Heuristic Programming Project
HPP-79-10

July 1979

PROTOTYPES AND PRODUCTION RULES: AN APPROACH TO KNOWLEDGE REPRESENTATION FOR HYPOTHESIS FORMATION

BY

Jan S. Aikins

Heuristic Programming Project
Computer Science Department
Stanford University
Stanford, California 94305

Prototypes and Production Rules:
An Approach to Knowledge Representation
for Hypothesis Formation

Jan S. Aikins

Heuristic Programming Project
Department of Computer Science
Stanford University
Stanford, California 94305

Abstract

A system called CENTAUR has been implemented to interpret data derived from pulmonary function tests using a knowledge representation that combines the advantages of both production rules and frames. The system uses a hypothesis-directed approach to problem solving, in which hypotheses are suggested by the initial data, further information is acquired, and then more specific hypotheses are selected. The hypotheses are represented as PROTOTYPES, frame-like data structures each of which characterizes some pulmonary disease. The prototypes guide the invocation of the production rules and focus the search for new information. Some of the advantages afforded by representing knowledge as both prototypes and rules are also presented.

Keywords: Representation of Knowledge, Frames, Production Rules, Medical Computing, Knowledge-Based Systems

1 Introduction

Much of Artificial Intelligence research has focused on determining the appropriate knowledge representations to use in order to achieve high performance from knowledge-based systems. The principal Artificial Intelligence theme being explored in this present research is that there are many advantages to a system that uses both frame-like structures and production rules to perform problem-solving tasks in knowledge-intensive domains.

In order to test this theme, a knowledge representation was designed using a combination of frames and production rules. The frames are called **Prototypes** because they represent stereotypical situations which can be used as a basis for comparison to the actual situation given by the data.[1] The domain chosen was that of pulmonary physiology. The task was to interpret a set of pulmonary function test results, producing a set of interpretation statements and a diagnosis of pulmonary disease in the patient. A system called **CENTAUR** has been written to perform this task using prototypes that characterize the typical features of each pulmonary disease. Each feature is called a **Component** of the prototype. Associated with each component are production rules used to infer a value for the component. These production rules are a form of procedural attachment with a constrained, stylized syntax that makes them easier to examine than general procedures. This constrained syntax leads to other advantages, such as ease of acquisition and modifiability as discussed in [Davis, 1977a]. The prototypes focus the search for new information by guiding the invocation of the production rules and eliciting the most relevant information from

[1] The term prototype has been given the same meaning by other researchers, for example in KRL [Bobrow, 1978a], a prototype is a special kind of unit representing a hypothetical individual that is the typical member of a class, and in [Brachman, 1978] a prototype is a "generalized individual".

the user. These prototypes are linked together in a network in which the links specify the relationships between the prototypes.

This research developed out of the MYCIN project [Shortliffe, 1976], which uses a knowledge base of production rules to perform infectious disease consultations. Initially, a MYCIN-like production rule system called PUFF [Kunz, 1978] was written to perform pulmonary function test interpretations. Problems with the production rule formalism in PUFF and similar rule-based systems, such as a need to focus the search for new information and the desire to represent characteristic patterns of disease, motivated the creation of a prototype-directed system.

This paper presents an example of the CENTAUR system performing an interpretation of a set of pulmonary function test results, focusing on CENTAUR's knowledge representation and control structure. In addition, some advantages of the prototype-directed system over the rule-based approach for this problem are given.

2 The CENTAUR System

2.1 Overview

The CENTAUR consultation system produces an interpretation of data and a pulmonary diagnosis based on a set of pulmonary function test results. The inputs to the system are the pulmonary function test results and a set of patient data including the patient's name, sex, age, and a referral diagnosis. The output consists of both a set of interpretation statements that serve to explain or comment on the pulmonary function test results and a final diagnosis of pulmonary disease in the patient.

CENTAUR uses a hypothesis-directed approach to problem solving where the possible hypotheses are represented by the prototypes. The goal of the system is to confirm one or more of the prototypes in the prototype network as matching the data in an actual case. The final set of confirmed prototypes is the system's solution for classifying the data in that case. The prototypes represent the various pulmonary diseases, their degrees and subtypes, with the result that the set of confirmed prototypes represents the diagnosis of pulmonary disease in the patient.

The system begins by accepting the test and patient data. Data entered in the system suggests or "triggers" one or more of the prototypes. The triggered prototypes are placed on a Hypothesis List and are ordered according to how closely they match the data. The prototype that matches the data most closely is selected to be the **Current Prototype**, the system's current best hypothesis about how to classify the data in the case.

In the example below, the prototype which represents a pulmonary function consultation itself, the PULMONARY-DISEASE prototype, has been selected as the initial Current Prototype. The initial data is requested and the user responses (preceded by a double asterisk **) are recorded. The system attempts to fill in values for the components of a prototype, which may cause rules to be invoked; or, if no rules are associated with the component, the system will ask the user for the value. When all of the prototype components have values, a decision is made by the system as to whether the given data values are a close enough match (to those expected for the prototype) to confirm the prototype as matching the data. Another prototype is then selected as the Current Prototype, and the process repeats. The system moves through the prototype network confirming or disproving disease prototypes. The attempt to match data and prototypes continues until each piece of data has been accounted for by some confirmed prototype or until the system has concluded that it cannot account for any more of the data. A portion of the prototype network for the pulmonary function application is given in Figure 2.1 for reference. Details of the knowledge representation and control structure for the CENTAUR system are given in Section 3 and Section 4.

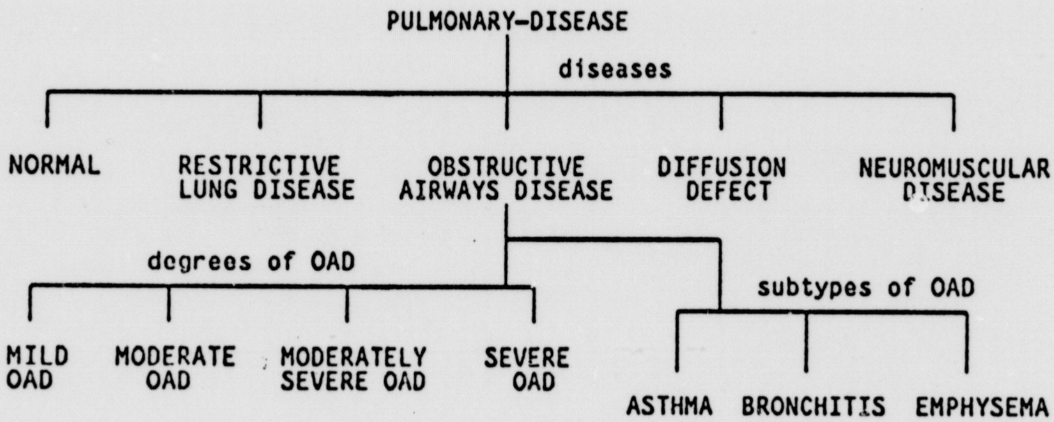

FIGURE.2.1 A Portion of the Prototype Network

2.2 CENTAUR Example

The following is an example of an interpretation of a set of pulmonary function test results for one pulmonary patient. Comments are in italics.

```
*CENTAUR*
14-Jan-79 13:54:07

The current hypothesis is that an interpretation of the pulmonary
function tests is desired.

   --------PATIENT-7--------
```
(The initial data is given by the user.)
```
   1) Patient's identifying number:
   ** 7446
   2) referral diagnosis:
   ** ASTHMA
   3) FEV1/FVC ratio:
   ** 40
```

```
    [Trigger for OAD and CM 900]
```

(Prototype OAD is triggered by the value 40 for the FEV1/FVC ratio. The Certainty Measure (CM) indicates on a numerical scale the degree of certainty with which the prototype is indicated by the data.)

```
  4) TLC(body box)observed/predicted:
  ** 139
  5) FVC/FVC-predicted:
  ** 81
    [Trigger for NORMAL and CM 500]
```

(The questioning continues and other prototypes are triggered by the data values.)

```
  6) RV/RV-predicted:
  ** 261
  7) the DLCO/DLCO-predicted:
  ** 117
    [Trigger for NORMAL and CM 700]
  8) Change in FEV1 post←dilation - pre←dilation:
  ** 31
  9) MMF/MMF-predicted:
  ** 12
    [Trigger for OAD and CM 900]
  10) The slope (F50←obs-F25←obs)/FVC←obs:
  ** 9
    [Trigger for OAD and CM 900]

TRIGGERED PROTOTYPES
  PROTOTYPE: OAD, CM: 900, REASON: FEV1/FVC was 40
  PROTOTYPE: NORMAL, CM: 500, REASON: FVC was 81
  PROTOTYPE: NORMAL, CM: 700, REASON: DLCO was 117
  PROTOTYPE: OAD, CM: 900, REASON: MMF was 12
  PROTOTYPE: OAD, CM: 900, REASON: F5025 was 9
```

(A list of the prototypes that have been triggered is given. The CM and the value that caused the trigger are also listed. The primary purpose of the PULMONARY-DISEASE prototype is to acquire the initial data from the user. It has no components and is thus confirmed when it has completed its task.)

```
MoreSpecific Prototypes chosen:  NORMAL OAD
```

(Although there are five possible, more specific disease prototypes for the PULMONARY-DISEASE prototype, only the two that were triggered by the initial data are selected as possibilities to pursue. These prototypes are filled in with the data values that are already known in the case.)

```
! Surprise Value !  261 for RV in NORMAL, CM: 700
! Surprise Value !  139 for TLC in NORMAL, CM: 400
! Surprise Value !  40 for FEV1/FVC in NORMAL, CM: -166
! Surprise Value !  12 for MMF in NORMAL, CM: -499
! Surprise Value !  9 for F5025 in NORMAL, CM: -699
```

(Any data values that are not consistent with the values expected for that disease prototype are noted as Surprise Values, and the CM for that prototype is lowered. In this case, five of the data values are not consistent with the NORMAL pulmonary function prototype.)

```
Hypothesis List:  (OAD 990) (NORMAL -699)
```

(The Hypothesis List of triggered prototypes is then ordered according to the CM of the prototypes and a new Current Prototype, OAD, is chosen.)

The current hypothesis is that there is an interpretation of Obstructive Airways Disease.

Components of OAD chosen to trace: F25 D←RV/TLC

(In order to instantiate the OAD prototype, two more components must have values. These are then asked of the user if there are no rules associated with the components that can be used to deduce their values.)

11) The flow F25:
** UNKNOWN

12) RV/TLC Observed-Predicted:
** 25

It is confirmed that there is an interpretation of Obstructive Airways Disease.

(The OAD prototype is confirmed. Control information associated with the prototype specifies that the Degree of OAD should be determined next, followed by the Subtype of OAD.)

MoreSpecific Prototypes chosen: MILD-OAD MODERATE-OAD MODERATELY-SEVERE-OAD SEVERE-OAD

(No degree prototypes were triggered by the data values, so all of them are selected as possible hypotheses to be filled in along with the data values in the case, and the consultation continues.)

...

Confirmed List: ASTHMA SEVERE-OAD OAD PULMONARY-DISEASE

(Eventually SEVERE-OAD and ASTHMA are also confirmed. Data values that can be accounted for by one of the confirmed prototypes are marked. If there are data values remaining that cannot be accounted for by the confirmed prototypes, the system will attempt to determine if there are multiple diseases in the patient. The Refinement Rules associated with the confirmed prototypes are then executed, followed by the Action slots of the confirmed prototypes. The prototype summary and interpretation is then printed.)

-----PULMONARY-DISEASE is confirmed for PATIENT-7-----

-----OAD is confirmed for PATIENT-7-----

OAD was suggested by the following findings
FEV1/FVC: 40
MMF: 12
F5025: 9

In addition, OAD is consistent with
TLC: 139
RV: 261
D←RV/TLC: 25
FEV1: 42
RDX: ASTHMA

The OAD accounts for the findings: RDX FEV1/FVC

-----SEVERE-OAD is confirmed for PATIENT-7-----

```
SEVERE-OAD is consistent with
   RV: 261
   TLC: 139
     .
     .
     .

Conclusions:
        Smoking probably exacerbates the severity of the patient's
            airway obstruction.
        Discontinuation of smoking should help relieve the symptoms.
        Good response to bronchodilators is consistent with an asthmatic
            condition, and their continued use is indicated.
        The high diffusing capacity is consistent with asthma.

Pulmonary Function Diagnosis:
        Severe Obstructive Airways Disease.
        Asthmatic type.
```

3 CENTAUR Knowledge Representation

3.1 Overview

Knowledge is represented in CENTAUR by both production rules and prototypes. Following frame terminology [Minsky, 1975], each prototype contains SLOTS of information associated with it. One of these is the slot COMPONENTS that lists a set of features that express the substantive characteristics of the prototype. Each component may, in turn, have slots of information associated with it. An example is given in Figure 3.1 where the possible types of slots are listed on the left, and an instantiation of some of the slots for the OBSTRUCTIVE AIRWAYS DISEASE (OAD) prototype is listed on the right.

PROTOTYPE	Obstructive Airways Disease
GENERAL INFORMATION SLOTS	
--Bookkeeping Information	Author: Aikins Date: 27-OCT-78 Source: Dr. Fallat
--Pointers to other prototypes (link prototype)	Pointers: (degree MILD-OAD) (degree MODERATE-OAD) ... (subtype ASTHMA) ...

--English phrases	Hypothesis: "There is an interpretation of OAD."
CONTROL INFORMATION SLOTS Control If-Confirmed If-Disproved Action	If-Confirmed: Deduce the degree of OAD Deduce the subtype of OAD Action: Deduce any findings associated with OAD Print the findings associated with OAD
COMPONENTS Plausible Values Default Value Possible Error Values Rules Importance of value to this prototype . . .	**TOTAL LUNG CAPACITY** Plausible Values: >100 Importance: 4 **REVERSIBILITY** Rules: 19, 21, 22, 25 Importance: 0 (value not considered)

FIGURE.3.1 A sample prototype with possible slots on the left and values for OAD on the right.

The production rules consist of one or more premise clauses followed by one or more action clauses. An example is given in Figure 3.2.[1] In general, the premise clauses specify a set of value ranges for some of a prototype's components, and the action clauses make conclusions about the values of other components.

[1] The rule is stored internally in Interlisp; the English translation shown is generated from that.

```
RULE013
-------
[This rule applies to any patient, and is tried in order to find out
     about the degree of obstructive airways disease as indicated by
     the MMF or the findings about the diagnosis of obstructive
     airways disease]

  If:  1) A: The mmf/mmf-predicted ratio is less than 20, and
          B: The fvc/fvc-predicted ratio is greater than 80, or
       2) A: The mmf/mmf-predicted ratio is less than 15, and
          B: The fvc/fvc-predicted ratio is less than 80
  Then:  1) There is strongly suggestive evidence (.9) that the degree
            of obstructive airways disease as indicated by the MMF is
            severe, and
         2) It is definite (1.0) that the following is one of the
            findings about the diagnosis of obstructive airways disease:
            Low mid-expiratory flow is consistent with severe airway
            obstruction.
```

FIGURE.3.2 A Sample Production Rule--English Version

3.2 Prototypes and Components

In general, each prototype represents a set of common features of a group of entities which are perceived to be similar in some way. In CENTAUR, the prototypes represent the characteristic features of some pulmonary disease. For example, in the OAD prototype in Figure 3.1, there are components for many of the pulmonary function tests that are useful in characterizing a patient with OAD; two of these are shown in the sample figure. For example, the TOTAL LUNG CAPACITY of a patient with OAD is typically higher than that of a person with normal pulmonary function. Thus there is a component, TOTAL LUNG CAPACITY, with a range of PLAUSIBLE VALUES that are characteristic of a person with OAD.

In addition to a set of plausible values, that is, values consistent with the hypothesis represented by the prototype, the components may have additional information associated with them. (The ways in which this information is used is discussed in Section 4.) There may be one or more POSSIBLE ERROR VALUES, that is, values that are inconsistent with the prototype or that might have been specified by the expert to check what he considers a measurement error. Generally, both a reason for the error and a possible fix for the error are specified. For example, the expert may specify that one of the pulmonary function tests be repeated to ensure accuracy. Data values are classified as Plausible Values, Possible Error Values or SURPRISE VALUES using the ranges of values specified in these slots. Surprise Values are all of those values that are neither Plausible Values nor Possible Error Values. They indicate facts that cannot be accounted for by the hypothesis represented by the prototype. A component may also have a DEFAULT VALUE. Thus, all of the components in a disease prototype, with their default values, form a picture of the typical patient with the disease. Finally, each component has an IMPORTANCE measure (from 0 to 5) that indicates the relative importance of a particular component in characterizing the disease, and is used in determining how closely the prototype matches the data.

In addition to the domain-specific components, each prototype contains slots for general information associated with it. This includes bookkeeping information, English phrases used in communicating with the user, and pointers to other prototypes in the prototype network, which are useful, for example, when either more general disease categories or more specific subtypes of disease are indicated. Some control information is represented explicitly in slots associated with the prototype and will be discussed in Section 4. This information includes what to do in order to instantiate the prototype and what to do when the prototype has been confirmed or disproved. Each prototype also has associated with it a CERTAINTY MEASURE (from -1000 to 1000) that indicates how certain the system is that the prototype matches the data in each case, and is used in selecting the prototype that represents the current best hypothesis.

3.3 Production Rules

The CENTAUR knowledge base also includes sets of production rules. Many of the production rules are classified as INFERENCE RULES, rules used to infer information in the domain. They refer to values for components in their premise clauses and make conclusions about values of components in their action clauses. An example of one of the Inference Rules is given in Figure 3.2 above. The RULES slot associated with a component contains a list of all Inference Rules that make a conclusion about that component. These may be applied when a value is needed for the component.[1]

The production rules in CENTAUR are grouped according to their function. Besides the Inference Rules, there are three other sets of rules. Those rules whose actions make summary statements about the results of the pulmonary function tests are classified as SUMMARY RULES; rules that refer to values of components in their premises and suggest general disease categories in their actions are classified as TRIGGERING RULES. These are used to "trigger" or suggest the disease prototypes. Those rules that are used in a second stage of processing, after the system has formulated lists of confirmed and disproved prototypes are called REFINEMENT RULES; they are used to refine an interim diagnosis, producing a final diagnosis about pulmonary disease in the patient. The Refinement Rules constitute a further set of domain expertise; they test the system's tentative conclusions, which may result in a modification of these conclusions. For example, if two diseases can account for a given pulmonary function test result and both have been confirmed in that case, a Refinement Rule may determine which disease process should account for the test result in the final interpretation.

---

[1] If no rules are associated with the component, the user will be asked for the value. If the user responds "UNKNOWN" and the component has a Default Value, that value will be used.

4 Control Structure

4.1 Control Structure Overview

The control information used by CENTAUR is contained either in slots that are associated with the individual prototypes or in a simple interpreter. Control strategies that are specific to an individual prototype are represented in slots associated with that prototype, with more general system control being expressed in the interpreter.

The control structure can be broken into three stages: a hypothesis-formation stage in which data values are acquired and an attempt is made to match prototypes to data, resulting in a list of confirmed prototypes; a refinement stage in which the Refinement Rules are applied to the list of confirmed prototypes to "debug" this list and further refine the recommendations that will be made; and a final "clean-up" stage in which, for example, findings associated with the prototype are printed.

In the hypothesis-formation stage, the interpreter attempts to match one or more of the prototypes with the data in an actual case. At any one time there is one Current Prototype that the system is attempting to match to the data in the case. Attempting a match for this prototype entails finding values for the prototype components--i.e., "instantiating" the prototype. The exact method to be used in instantiating the prototype depends on the individual prototype and is expressed in one of the prototype control slots.

When all of the facts have been accounted for by some confirmed prototype or when it has been determined that no other prototype could account for a known fact,[1] the system has completed the hypothesis-formation stage. The Confirmed List of prototypes then represents the system's hypothesis about how to classify the data in the case.

---

[1] This statement oversimplifies the actual matching criteria used by the system. Some tolerance for a mismatch between known fact values and plausible values in the prototype is allowed.

In the refinement stage, additional knowledge in the form of the Refinement Rules is applied in the case before generating the final pulmonary function interpretation and diagnosis. Further information may be sought from the user at this stage. For example, further lab tests may be suggested or additional test results may even be required before a final diagnosis is given.

The result of the execution of the Refinement Rules is a final set of confirmed prototypes and a list of all of the facts in the case with an indication of which prototypes account for which facts. In the final clean-up stage, the system executes the clauses specified in the ACTION slot of each confirmed prototype. Typically, these clauses execute Summary Rules associated with the prototype[1] or print interpretation statements. The ACTION slot of the PULMONARY-DISEASE prototype itself causes the final interpretation and pulmonary diagnosis to be printed.

4.2 Prototype Control Slots

Four of the slots associated with the prototype contain a set of one or more clauses that are executed by the system at specific times to control the consultation. Each clause expresses some action to be taken by the system in order to instantiate the prototype (CONTROL slot), upon confirmation of the prototype (IF-CONFIRMED slot), in the event that a prototype is disproved (IF-DISPROVED slot), and in the clean-up stage after the system processing has been completed (ACTION slot).

When a prototype is first selected as the Current Prototype, the system executes the clauses in the CONTROL slot of that prototype. The information in this slot indicates how to proceed in order to instantiate the prototype. Executing these clauses causes values to be

---

[1] The premise of a summary rule typically checks the values for one or more parameters and the action generates an appropriate summarizing statement.

obtained for the prototype components. If no CONTROL slot is associated with a prototype, the interpreter will attempt to fill in values for the prototype components in the order of their importance measures.

When all of the clauses in the CONTROL slot have been executed and the prototype has been instantiated, a decision is made[1] as to whether the prototype should be confirmed as matching the data in the case. The system then checks either the IF-CONFIRMED slot or the IF-DISPROVED slot to determine what should be done next. Similarly, the ACTION slot specifies steps to be taken for a confirmed prototype during the clean-up stage.

5 Advantages of the Prototype-Directed Approach

5.1 Overview

This section states some of the advantages of using the prototype-directed approach in CENTAUR for the pulmonary function interpretation task, as compared to the purely rule-based approach used in PUFF. These advantages can be grouped into two broad categories: those dealing with the knowledge base representation itself and those dealing with the system's reasoning and performance.

5.2 Knowledge Representation

Advantages in knowledge representation occur because some knowledge that was previously represented in rules is now represented more clearly in prototypes. In addition, some new knowledge is being represented in prototypes; for example, plausible ranges of

---

[1] It would be possible to associate such a confirmation criterion with each individual prototype, but this has not been found to be necessary for the pulmonary diagnosis problem. Instead, the system uses a general algorithm applicable to all of the prototypes that checks the values of the components and their importance measures to determine if the prototype should be confirmed.

values for each of the pulmonary function tests for each disease can be listed, as well as the relative importance of each measurement in a particular disease prototype.

One specific example of an advantage of using prototypes for knowledge representation is that the prototypes explicitly represent control information that was formerly represented in the PUFF inference rules. In the PUFF system, there are rules whose purpose it is to guide computation by controlling the invocation of other rules. This feature can be very confusing to the medical experts since they do not know which rules are those intended to represent descriptive medical expertise and which rules are those serving a necessary computational function. For example, a PUFF rule necessary to determine whether there is Obstructive Airways Disease (OAD) in the patient is:

> If an attempt has been made to deduce the degree of OAD, and an attempt has been made to deduce the subtype of OAD, and an attempt has been made to deduce the findings about OAD, then there is an interpretation of potential OAD.

This rule expresses some of the control structure of the system, namely, that when there is an interpretation of OAD, then the degree, subtype, and findings associated with the OAD should be determined. In CENTAUR, inference rules that guide computation have been removed from the rule-base, leaving a less confusing, more uniform rule-base where each inference rule represents some descriptive "chunk" of medical expertise. Computation is now guided by the prototypes. For example, in the OAD prototype in Figure 3.1, the IF-CONFIRMED and ACTION slots explicitly represent the steps to be taken when OAD is confirmed and when OAD findings are to be printed.

5.3 The Reasoning and Performance of the System

A second category of advantages deals with the way the system reasons about the problem. This is evident in part by watching the performance of the system, that is, the questions that are asked and the order in which information is acquired. Some of the advantages of a prototype-directed system are the following:

(A) Consultation flow more closely follows physician's reasoning.

The consultation begins with specific test results suggesting or "triggering" some of the prototypes. The prototypes serve as tentative hypotheses about how to classify the data in a given case. They also guide further inquiry. As new information is acquired, these hypotheses are revised, or, in CENTAUR's terms, prototypes are confirmed or disproved and new prototypes may then be suggested. The process of medical problem solving has been discussed by many researchers (e.g., [Elstein, 1978] and [Kassirer, 1978]) and it is widely felt that this sequence of suggesting hypotheses, acquiring further information, and then revising the hypotheses is, in fact, the problem-solving process used by most physicians. Thus there is increased conceptual clarity, in that the user can understand what the program is doing, and this factor leads to other advantages, for example, the system can offer the user a more intelligible explanation of its performance during the consultation. Giving the system the ability to explain its knowledge and performance has been a primary design goal of the present research efforts. It has been critical in our experience to have this explanation capability in order to ensure acceptance by the user. In the PUFF rule-based system, all explanations were given in terms of which rules were being used to infer information or which rules had been used previously. In CENTAUR, the prototypes provide a broad context in which more specific reasoning can be explained, and because the prototype-directed system reasons in a manner more like a human user, its behavior seems more natural and transparent.

(B) Questions are asked in a reasonable order.

In a rule-based system such as PUFF, questions are asked of the user as rules are invoked that contain clauses referring to information that is not yet known. The expert can control the order in which the questions are asked only by writing rules to enforce some order. As has been discussed, this procedure results in a potentially confusing rule base where some rules represent descriptive medical expertise and others guide computation. In the prototype-directed system, the expert specifies the order in which information is acquired for each prototype in the control slot. Thus control information is labeled explicitly as such, and the rule base remains uniformly a body of descriptive medical expertise. The expert can also specify what information must be acquired and what information is optional, using the Importance measure associated with each component.[1]

(C) Only relevant questions are asked.

Another advantage of CENTAUR over PUFF is that only those hypotheses suggested by the initial data are explored. For example, if the Total Lung Capacity (TLC) for the patient is 70, then CENTAUR would begin exploring the possibility of Restrictive Lung Disease (RLD) because a low TLC would trigger the RLD prototype.[2] In the PUFF program, the first disease tried is always OAD, so the PUFF program would begin asking questions dealing with OAD. These questions would seem irrelevant considering the data, and, indeed, if there were no data to indicate OAD, such questions would not be asked by CENTAUR.

(D) Inconsistent information is indicated.

During a prototype-directed consultation, it is also possible to point out inconsistent or possibly erroneous data as it is entered, so that a technician can repeat a test immediately

---

[1] Optional information is indicated by assigning that component an Importance measure of 0.

[2] A low TLC is consistent with a hypothesis of RLD; a high TLC is consistent with OAD.

or at least decide whether it is worth the time to continue analyzing the case. This feature is invoked when possible error values are detected for a component of a prototype, or when no prototype can be determined to account for a given fact value.[1]

6 Summary

CENTAUR was designed in response to problems that occurred while using a purely rule-based system. By changing the knowledge representation to include prototypes as well as production rules, new knowledge was represented. Further, knowledge that had been represented rather awkwardly in rules was represented more clearly in prototypes. The production rules were retained as a stylized form of procedural attachment that could be easily examined or modified. By altering the control structure so that a best-fit matching process of prototypes to data produced a current best hypothesis to guide further search, a more focused consultation resulted which more closely followed the way physicians reason. Control knowledge was explicitly labeled and made prototype-specific so that control of the consultation was adapted to the current best hypothesis. In summary, the prototype-directed system achieved better reasoning and performance than the rule-based system. In addition, although representing knowledge in production rules alone did not seem adequate for this task, the ability to represent knowledge in prototypes as well did provide the needed flexibility.

---

[1] It is also possible that there is an overly restricted range of plausible values for a prototype component, in which case the user may extend the range to encompass the indicated fact value.

Acknowledgments

I wish to acknowledge the contributions of Dr. Robert Fallat, who provided the medical expertise, and the members of the Stanford Heuristic Programming Project, past and present, who gave many helpful comments and suggestions on earlier drafts of this paper.

This work was supported by the Advanced Research Projects Agency under contract MDA 903-77-C-0322. Computer facilities were provided by the SUMEX-AIM facility at Stanford University under National Institutes of Health grant RR-00785. The author is sponsored by the Xerox Corporation under the direction of the Xerox Palo Alto Research Center.

References

[Bobrow, 1978a]
D. G. Bobrow and T. Winograd, An Overview of KRL, a Knowledge Representation Language. *Cognitive Science 1*(1): 3-46.

[Brachman, 1978]
R. J. Brachman, *A Structural Paradigm for Representing Knowledge*. BBN Report 3605, May 1978.

[Davis, 1977a]
R. Davis and J. King, An Overview of Production Systems. In E. W. Elcock and D. Michie (Eds.), *Machine Intelligence 8*. New York: Wiley & Sons, 1977. Pp. 300-332.

[Elstein, 1978]
A. S. Elstein, L. S. Shulman, and S. A. Sprafka, *Medical Problem Solving--An Analysis of Clinical Reasoning*. Harvard University Press, Cambridge, Mass., 1978.

[Kassirer, 1978]
J. P. Kassirer and G. A. Gorry, Clinical Problem Solving: A Behavioral Analysis. *Annals of Internal Medicine 89*: 245-255, 1978.

[Kunz, 1978]
J. C. Kunz, R. J. Fallat, et. al., *A Physiological Rule Based System for Interpreting Pulmonary Function Test Results.* HPP-78-19 (Working Paper), Heuristic Programming Project, Dept. of Computer Science, Stanford University, December 1978.

[Minsky, 1975]
M. Minsky, A Framework for Representing Knowledge. In P. Winston (Ed.), *The Psychology of Computer Vision*. New York: McGraw-Hill, 1975. Pp. 211-277.

[Shortliffe, 1976]
E. H. Shortliffe, *Computer-Based Medical Consultations: MYCIN*. New York: American-Elsevier, 1976.