20

Developing Microprocessor-Based Expert Models for Instrument Interpretation

Sholom M. Weiss, Casimir A. Kulikowski, and Robert S. Galen

Just as PUFF was built using EMYCIN and then converted to run in a different environment, the last program discussed here was built using the EXPERT system-building tool developed at Rutgers University. In this case, however, Sholom Weiss and Casimir Kulikowski devised a scheme for developing an interpretive system and transferring it to a microprocessor environment. The scheme was successfully implemented and tested by producing a program for interpreting results from a widely used medical laboratory instrument: a scanning densitometer. Specialists in the field of serum protein electrophoresis analysis, including particularly Dr. Robert Galen, provided the knowledge needed to build an interpretive model using EXPERT. By constraining a few of the structures used in the general model, it was possible to develop procedures for automatically translating the model to a specialized application program and then to a microprocessor assembly language program. Thus model development was able to take place on a large machine, using established techniques for capturing and conveniently updating expert knowledge structures, while the final interpretive program was targeted to a microprocessor that was dependent on the application and suitable for installation as an output controller for an electrophoresis device. The experience of Weiss, Kulikowski, and Galen in

From *Proceedings of the Seventh International Joint Conference on Artificial Intelligence*, pp. 853–855 (1981). Used by permission of International Joint Conferences on Artificial Intelligence, Inc.; copies of the *Proceedings* are available from William Kaufmann, Inc., 95 First Street, Los Altos, CA 94022.

carrying out the complete process illustrates many of the requirements involved in taking an expert system from its early development phase to actual implementation and use in a real-world application. The resulting instrument produces interpretations as well as the usual protein electrophoresis curves and component percentages. It is a commercially available product—the first marketed medical device to have used AI techniques in its development.

20.1 Introduction

Most knowledge-based medical consultation systems developed during the 1970s were relatively large-scale experimental prototypes (Chapters 5 through 8). Their advice on diagnostic and treatment problems typically involved approximate reasoning over a space of many interrelated hypotheses, characteristically supported by hundreds of observations linked to them by various types of reasoning rules. By adopting symbolic reasoning methods with more powerful representations than the traditional mathematical decision-making schemes, these knowledge-based systems produced results that were generally easier to analyze, explain, and update than those from more conventional systems. Human-engineering features were often stressed as an important means of enhancing the interaction with the expert systems. Successful clinical experience with many of these systems has been reported in pilot demonstration projects, yet few are in routine clinical use at present.[1] Both technical and social factors contribute to the difficulties of introducing expert systems into the everyday practice of medicine. One often cited technical factor is the slow rate of manual data entry required by most of the larger systems. This problem is minimized for applications where most of the data can be read directly off a clinical instrument and only a few items must be entered manually. The commercial availability and use of automated electrocardiogram interpretation programs (using traditional algorithmic techniques) support this point. Regardless of the methods used in constructing a knowledge base, or its complexity, instrument-derived interpretations are more likely to be accepted because they can be seen as extensions of the instrument. And since many advanced medical instruments are already microprocessor-controlled, it may be possible to add an interpretive module that enhances the performance of such a device at relatively little extra cost.

In this paper we briefly describe how we were able to accelerate the development of interpretive software for a widely used laboratory instrument, the scanning densitometer. We did this by automatically producing

[1]See Chapter 19 for a report on the successful use of PUFF at the Pacific Medical Center in San Francisco.

a computer translation of an expert model for serum protein electrophoresis interpretation, developed on a mainframe computer, into a microprocessor assembly language version. The translation methods have been generalized so that this process can be repeated for EXPERT (Weiss and Kulikowski, 1979) models in any domain, with a few restrictions on the types of knowledge structures used.

By taking this approach, we have demonstrated that knowledge-engineering methods from expert systems can be used to full advantage in producing an effective model, which can then be transferred with ease to a microcomputer.

20.2 Methods

Several general-purpose schemes for building consultation systems have evolved from work on the earlier, more specific domain-dependent systems. Two such schemes that were originally designed for representing medical consultation problems in particular are the EXPERT and EMYCIN (van Melle, 1979) systems. Both provide built-in control mechanisms operating over specific types of production-rule models. The consultation program of EXPERT is primarily event-driven, while that of EMYCIN is predominantly goal-directed.

The EXPERT system has been used in building a number of expert medical consultation models (mainly in ophthalmology, rheumatology, and endocrinology) and pilot prototypes in several nonmedical areas (spectroscopy interpretation, car repair, hazardous spill management, and oil well log interpretation).

The process of model design and transfer that we used in developing the microprocessor-based expert model for serum protein electrophoresis interpretation involved the following tasks:

- specification of the knowledge base using EXPERT,
- empirical testing with several hundred cases,
- refinement of the knowledge base by the expert,
- further cycle of testing with additional cases and review by independent experts,
- test of the final model on the large machine,
- automatic translation of the EXPERT model to a specialized program and a microprocessor assembly language program, and
- interfacing of assembly language model with instrument.

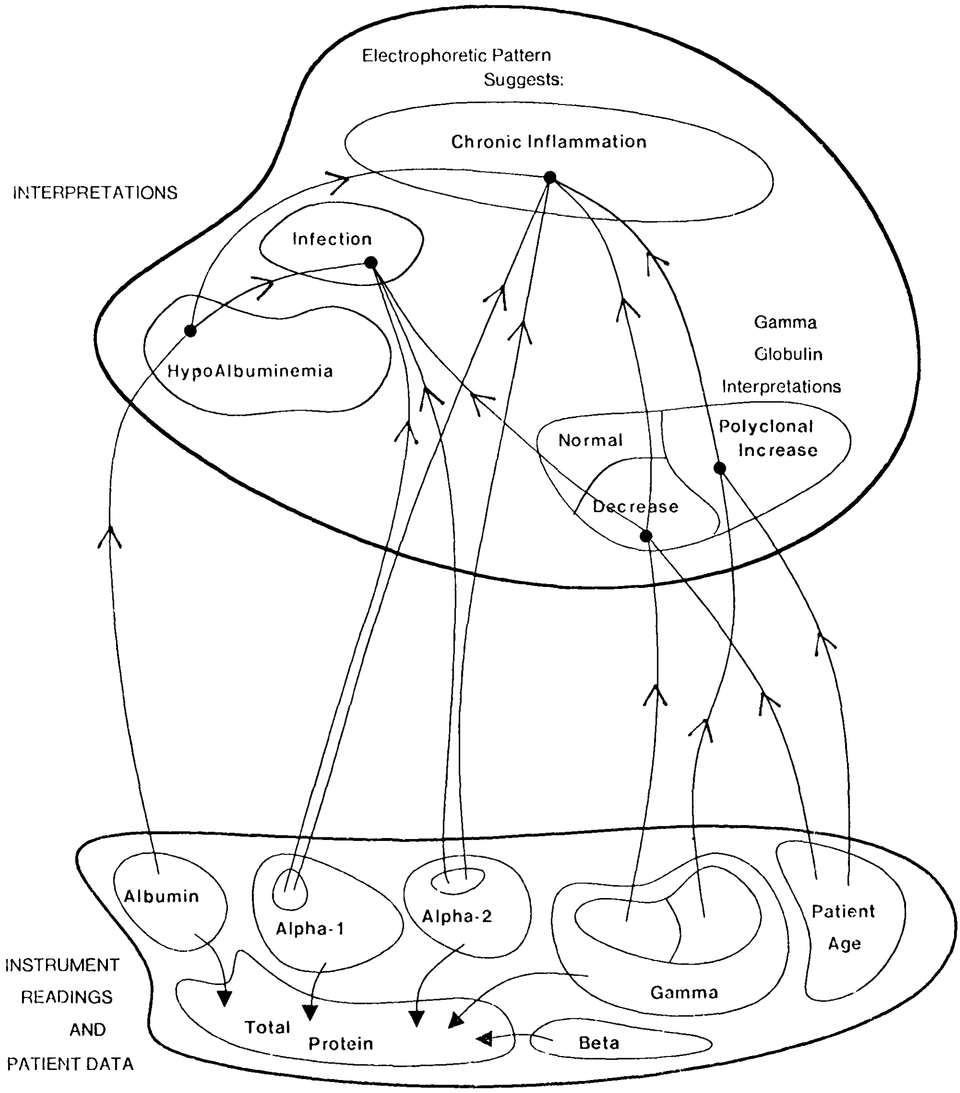

FIGURE 20-1 Sample rules (arrows) linking primary data and interpretative conclusions.

This last step requires detailed knowledge of the instrument. In this application, the manufacturer interfaced the interpretive program to the existing program for printing instrument readings.

Figure 20-1 illustrates the types of conclusions reached by the interpretive system and the type of rules used in reasoning. The most significant restriction on the type of production rules used in the model was to limit the use of confidence measures to three values, representing confirmation, denial, and unknown status. In applications of this type, it should be noted that the strategy of questioning is not a significant task because most of the information will be obtained directly from the instrument. In building the EXPERT model, we simulated this situation by entering the values of certain key features of the instrument signal (Figure 20-2) that are currently given as a digital output by the instrument. These features include peaks of the waveform and areas under certain segments of the waveform. A few items (patient identification, age, and some waveform features that are more easily scanned by the technician) are entered manually.

The serum protein electrophoresis model required several stages of refinement over a period of six months, with the aid of one principal expert and comments and suggestions from the independent experts. We began with a relatively small and simple model, having 10 conclusions and a production rule for each. After the first cycle of revision we had about 25 conclusions and 50 rules, which included many for handling interactions among the hypotheses. The current model has 38 conclusions and 82 production rules. Its performance on 256 test cases covering the full spectrum of conclusions is 100% acceptable to our experts. They expect differences of opinion on the amount of detail included in the present set of conclusions, but feel that covering infrequently found problems would detract from a model that is to be disseminated widely. An option for allowing users to add a written record of their own opinions on such unusual cases has been provided in the final microprocessor implementation.

20.3 Conclusions

The completed microprocessor version of the interpretative serum protein electrophoresis model may not look much different than it would if it had been hand-coded directly in the assembly language of the microprocessor or translated from an algorithmic language. There is, nevertheless, a fundamental difference. With our system, we can rapidly produce new versions of the microprocessor program from our high-level EXPERT model in response to any changes suggested by the experts or resulting from future empirical analysis and clinical tests in the field. In contrast, considerable effort would usually be required to recode directly on a microprocessor. Besides, the original expert-derived model is also very different from one produced by more traditional methods. Our conclusions and intermediate hypotheses were developed in such a way that they include not only diagnostic considerations but also prognostic, treatment, and future test selection decisions for motivating their use. The large amount of

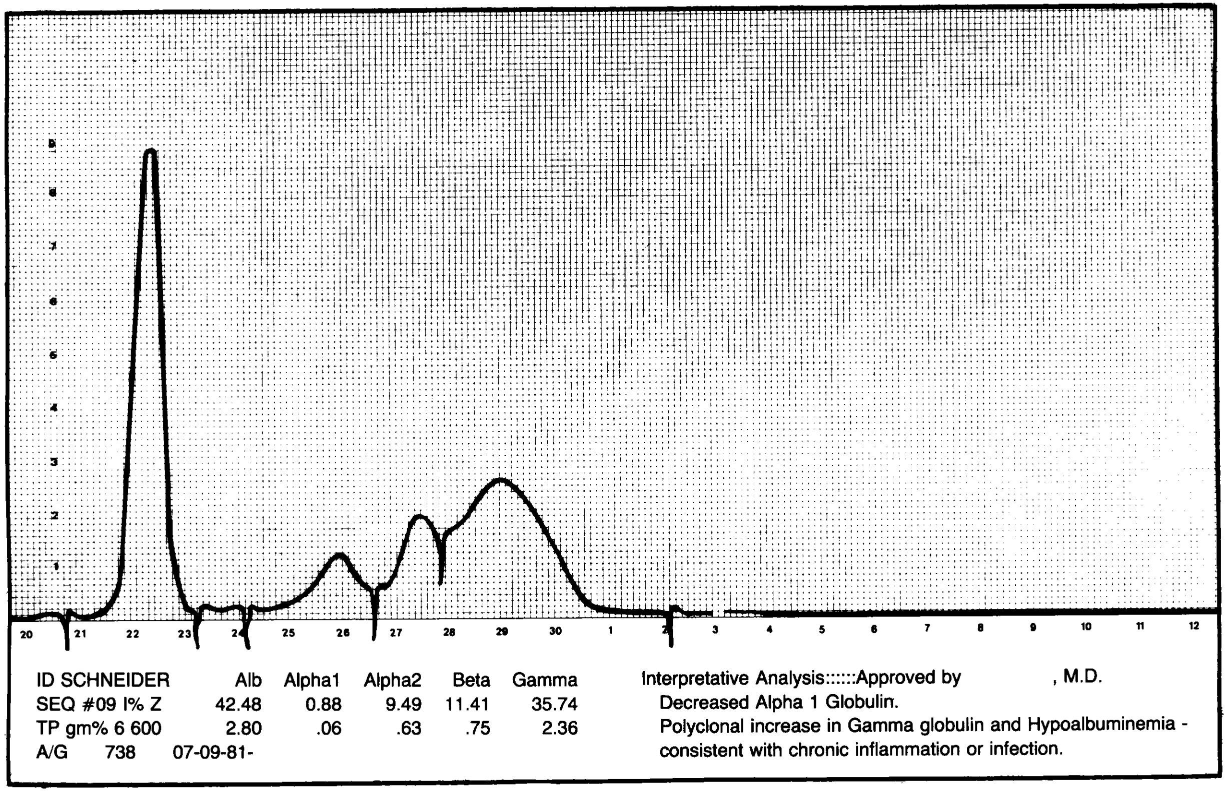

FIGURE 20-2 **Interpretative analysis: Electrophoretic pattern suggests chronic inflammation.**

experimentation that took place with the model as it went through its cycles of testing and modification could only be carried out on a larger system, with adequate facilities for analyzing many cases and knowledge-engineering tools for changing the model. A recently published version of an interpretative model in this domain, developed with very traditional programming techniques, shows a contrasting sparsity in diagnostic statements (Dito, 1977). In addition, the conclusions of that model appear to be overly specific given the nature of the supporting data. Thus, while programs of this type may be initially simple to implement, they do not incorporate the elements of expert reasoning that are essential to a clinically helpful program.

In conclusion, the work reported here is a novel illustration of the requirements encountered in taking an expert system from an early developmental phase to actual implementation and use in the real world. Such applications can lead to the increasing acceptance of expert systems in medicine and other domains where similar problems can be found.

ACKNOWLEDGMENTS

This work was supported in part by grant RR-643 of the National Institutes of Health, DRR-BRP. Technical assistance was provided by Helena Laboratories, Beaumont, Texas.