Report 83-06
Stanford -- KSL

Scientific DataLink

Graphical Access to the Knowledge Base of a Medical Consultation System.
Shoko Tsuji, Edward H. Shortliffe,
Feb 1983

card 1 of 1

Stanford Heuristic Programming Project
Memo HPP-83-6

February 1983

Graphical Access to the Knowledge Base of a Medical Consultation System

by

Shoko Tsuji and Edward H. Shortliffe

Departments of Medicine and Computer Science
Stanford University
Stanford, California 94305

Table of Contents

Abstract

As medical consultation systems using artificial intelligence techniques have become more sophisticated, they have also become large and complex. The time-sharing systems that were used to develop them are largely inadequate due to the burden placed upon computing and storage resources. Consequently, some of this research is being shifted to personal workstations which provide a sophisticated graphics interface in addition to satisfying the requirements for computational speed and memory. This paper examines the use of that graphics interface in the development of a tool for a system builder.

Introduction

Medical expert systems are consultation programs designed to give advice using both formal knowledge and the judgmental expertise of clinical specialists. When these systems first began to appear in the 1970's, many observers doubted their future role because their size and complexity placed an inordinate burden on the processing and storage resources of conventional time-sharing systems. In the 1980's, however, we are beginning to see the introduction of powerful single-user workstations that satisfy the consultation program's requirements for high speed and large memory. In addition, these workstations offer graphical capabilities which expand the range of interactive styles possible, both for the medical personnel who will use the programs *and* for the medical computer scientists who will build them. In this paper we shall focus on the second class of users, describing one developing system and the way in which graphical capabilities of a workstation have allowed us to design and implement new tools for the system builder.

Medical artificial intelligence (AI) programs are based on symbolic reasoning techniques that consume a great deal of computing power. In a time-sharing environment, this can result in two problems: system development may be slowed and poor response time may inhibit the use of the consultation program by physician users. In addition, AI expert systems tend to contain large amounts of information. This information is housed in a *knowledge base* of synthesized data, empirical judgments, problem solving strategies, and facts about objects. As knowledge bases grow in size and complexity, they become difficult to manage, even for those people who originally encoded the information and know its organization. Essential tasks, such as examining information, updating existing pieces of the knowledge base, adding new information to augment the reasoning capabilities, and debugging the complex interactions of the consultation system, become cumbersome without the use of tools to simplify access to and understanding of the various knowledge components.

Problems such as these were not easily solved even a few years ago. Since hardware costs were prohibitively high, it was not economically feasible to develop or distribute expert systems on dedicated main frame computers. Smaller computers could not provide the computational speed, memory, and storage devices that are necessary for system development.

The ONCOCIN System

ONCOCIN is a medical consultation system designed to assist physicians with the management of cancer patients enrolled in chemotherapy protocols [1]. It uses reasoning techniques developed in earlier medical expert systems. Cancer therapy is an appropriate domain for the use of AI techniques because the judgment involved in evaluating a patient's response to therapy and in formulating a treatment regimen cannot be completely rendered in an algorithmic form but it can be captured in rules.

Our early work on ONCOCIN has had to contend with time-sharing restrictions. One of the primary concerns has been to minimize the amount of time ONCOCIN takes to provide a treatment recommendation. In order to accomplish this, ONCOCIN is split into two separate programs which run in parallel under a time-sharing monitor. One takes care of the interaction with the physician, while the other uses patient data entered by the physician and reaches decisions about therapy. This prototype version runs on a DEC system 2020 computer and has been used effectively by Stanford oncologists since mid-1981. However, the size and complexity of ONCOCIN's knowledge base remains potentially overwhelming for both system builders and our collaborating physicians. Tools that have been developed using line-oriented text, input on conventional character display terminals, cannot optimally aid in visualizing the interaction of various components of the knowledge base or in anticipating its consequences. In addition, ONCOCIN is intended for eventual dissemination among community physicians and clinics. The DEC 2020 version is not suitable for this purpose, given the high cost of the machine and ONCOCIN's current dependence on that system's architecture.

High performance professional workstations may provide a solution. They have the necessary computing resources for large AI systems, yet they are small enough to fit inside an office and are available at a fraction of the price of a large main-frame computer. They also provide an opportunity to experiment with high quality graphics, which may provide a mechanism for dealing with knowledge base complexity. Graphics, more than any other medium, has the power to augment explanation capabilities for system users [2] and to allow examination of the knowledge base for system builders. What follows is a description of the personal workstation we have chosen to use and a description of a new tool to allow research staff on our project to examine ONCOCIN's knowledge base during knowledge base development and debugging.

Personal Workstations

Many observers view personal workstations as the next generation of computers and as an advantageous and viable alternative to time-sharing systems that dominated the 1970's. Workstations are being produced or are under development by a number of manufacturers but they share common features: a high resolution graphics display, a keyboard for conventional typed input, and a pointing device such as a *mouse*. A color display and audio output are optional. Workstations can be connected to one another by a high bandwidth network thereby allowing high cost peripherals, such as remote file servers and high quality printers, to be shared. Each machine has about a megabyte of main memory and one hundred megabytes of local disk storage which is used to provide a large virtual memory. They also come with a large user-writable microstore, which provides the support for graphics and high level languages such as PASCAL and LISP. AI researchers are primarily interested in LISP machines due to the suitability of this language for symbolic processing applications.

Moving research to personal workstations offers several advantages. First, a high quality interactive user interface plus the instant availability of enough processing power to update the graphics display would be impossible to support in a time-shared environment. Graphics has great potential impact on computer interface development: the interaction can be simplified by using a pointing device as the primary source of input, diagrams can take advantage of structured information, and simulations can clarify complex interactions.

Second, workstations offer improved and constant response since they can be dedicated to the task at hand, e.g., meeting the computational demands of a large program or dealing with highly interactive jobs such as editing. With ongoing improvements in memory, address space, computational speed, and disk hardware, a user will see little or no difference from the capabilities of dedicated large main frame computers.

Third, the user has access to the full power of the workstation without the loss of any resources that are normally available on a time sharing system. If one workstation becomes unavailable none of the others is affected, and if files are stored on a remote file server users can move to other machines and continue their work. An added benefit is that special experimental hardware, such as is needed for some applications, can be added and it will get the immediate response it needs; it can be difficult, both technically and politically, to add such devices to time-sharing systems.

Finally, systems developed on these computers will be easily exported if users purchase the same or comparable machine. Since these machines are already much less expensive than their main-frame counterparts and unit costs are continuing to drop dramatically (a 50% reduction in one workstation's price in the last two years, for example), it is reasonable to expect that interested users will be able to obtain the necessary hardware and to apply it in a cost-effective manner. The trend is clear: as personnel costs increase, the price of computers goes down and with the continued improvement of hardware, current and future AI system development will move to networks of personal workstations.

For our workstation experiments we have chosen the Xerox 1100 Professional Workstation (the *Dolphin*). The Dolphin is a medium-sized, microprogrammed personal computer with a medium-sized local disk, a large graphic display (808 points vertically, 1024 points horizontally for a net resolution of approximately 75 points per inch in each direction), a keyboard, and a mouse pointing device. A mouse is a small hand-held device resting on a metal tracking ball which can be moved across the tabletop beside the terminal. The movement of the mouse directs the movement of a cursor on the screen and, consequently, can be used to select items. The Dolphin runs Interlisp-D, an implementation of the LISP dialect we used for ONCOCIN's development on the DEC 2020. In addition to the standard Interlisp software, Interlisp-D provides a complete set of graphics operations to exploit the display capabilities of the Dolphin.

The ONCOCIN Knowledge Base

ONCOCIN's knowledge base is comprised of four main data structures that are used to represent oncology domain knowledge. These are contexts, parameters, rules, and control blocks. *Contexts* provide a high level description of a patient in terms of known chemotherapy plans. They embody domain concepts such as diseases, protocols, and chemotherapies which ONCOCIN uses as a means to focus it's reasoning process. *Parameters* are selected attributes of the patient, test results, drug doses, etc. that ONCOCIN uses to formulate a recommendation. Values of attributes are defined based on context, deduced through rules, requested from the user, or concluded by default. *Rules* are the standard data type found in rule-based production systems like MYCIN [3]. They are used to conclude a value for a parameter, thereby emulating the reasoning that settles issues such as how much to attenuate a dose, whether to delay therapy, or how to recognize a toxic reaction. Rules can be invoked in either a data-driven mode or a goal-directed mode. Data-driven simply means that someone has supplied the value of a parameter and rules are tested to see if other parameter values can be inferred from it. Goal-directed means that the system attempts to find a value for a parameter either by asking the user or by invoking other rules [4]. *Control blocks* provide the separation of control information and domain specific information. They are a high level description of a system's method for performing tasks, such as formulating a therapeutic regimen or calculating the correct dose of a drug.

Currently, ONCOCIN's knowledge base contains more than three hundred rules, hundreds of parameters, and encoded information for thirty-four Hodgkin's disease and lymphoma protocols. Furthermore, we plan soon to finish adding non-Hodgkins protocols and to begin including protocol regimens for other cancers.

The main difficulty in dealing with ONCOCIN's knowledge base is that without some processing much of the information that can be retrieved is either irrelevant or poorly organized for use by either a system builder or a physician. Consequently, it may take a long time to comprehend. Some experimentation was done on a prototype static query system to access all data structures in ONCOCIN's knowledge base on the DEC 2020. The system builder could select a number which represented a choice from a numbered list of interactions the system would display. This mode of interaction was slow and repetitive. Line-oriented text output was satisfactory for displaying information about individual facts, but it failed to provide an overall picture of the relationships with other knowledge structures. There was a clear need for a more powerful interface, something that could provide the power and ease of natural language without the problems of ambiguity.

Knowledge Base Access on the Dolphin

In response to these needs, we have developed a graphical query system which provides access to information about protocols, chemotherapies, drugs, rules, parameters, and control blocks. The Dolphin's graphic facilities provide a powerful means for displaying the relationships among these entities. Much of the interaction with the system is accomplished through the use of windows, menus, and the mouse rather than through a query command language and the keyboard. Windows are special regions of the screen which can be treated like pieces of paper since the user can create and destroy them at will as well as control their location and size.

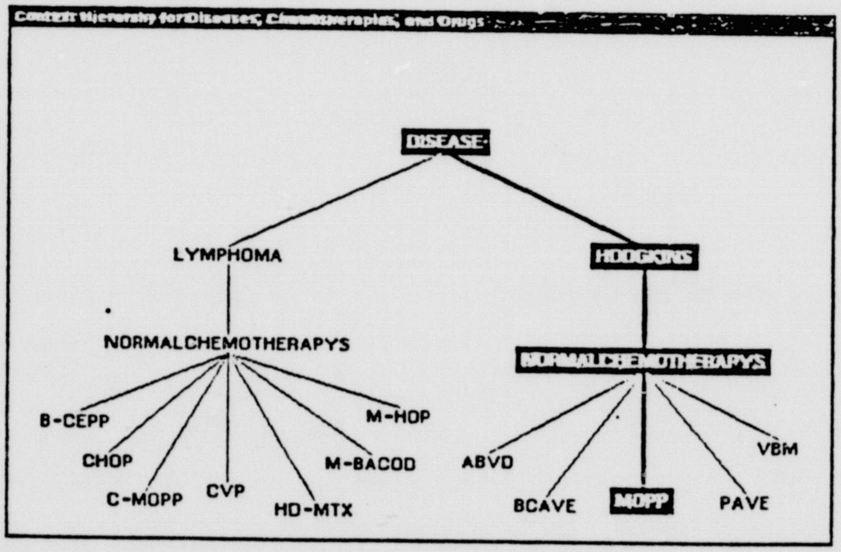

Figure 1

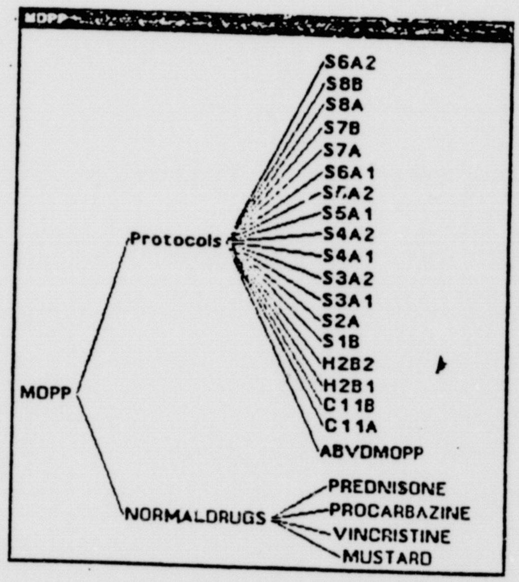

Figure 2

At the beginning of the interaction the query system displays a window which contains a graph of the hierarchical relationship between diseases and chemotherapies (Fig. 1). The bold face lines and inverted text indicate that the user has used the mouse to constrain the situation to the chemotherapy MOPP in Hodgkin's disease. Information requested from this point on will be valid only for that therapy. The mouse is used to prompt for more information about MOPP by simply pointing and pressing a mouse button. A menu will appear with the choices *General Information* and *Inspect Property List*. When *General Information* is selected, the results appear in two windows; one contains a graphical display (Fig. 2) and the other contains a text description of MOPP (which includes information about the duration of the A and B cycles as well as dosing for the drugs). Fig. 2 shows in what protocols MOPP is used and what drugs normally comprise it. If further information is desired about either protocols or drugs in Fig. 2, the mouse is used again to select an item. Fig. 3 shows the result of selecting procarbazine.

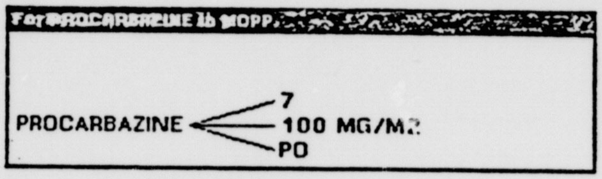

Figure 3

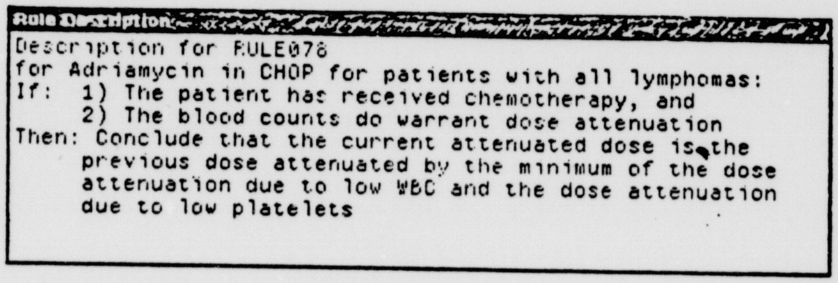

Figure 5

The interaction of rules and parameters is probably hardest to visualize and is certainly difficult to convey through character-oriented output alone. A more natural representation of a network structure is utilized, where rules and parameters are the nodes of the network and their relationships are links. The network illustrates the effects of a rule or parameter. In order to examine this portion of the knowledge base, the mouse is used to select the option *Rules* from the global menu. The user can specify a context to focus the reasoning chain, once again by using the mouse and a series of menus. Fig. 4 shows the result of examining

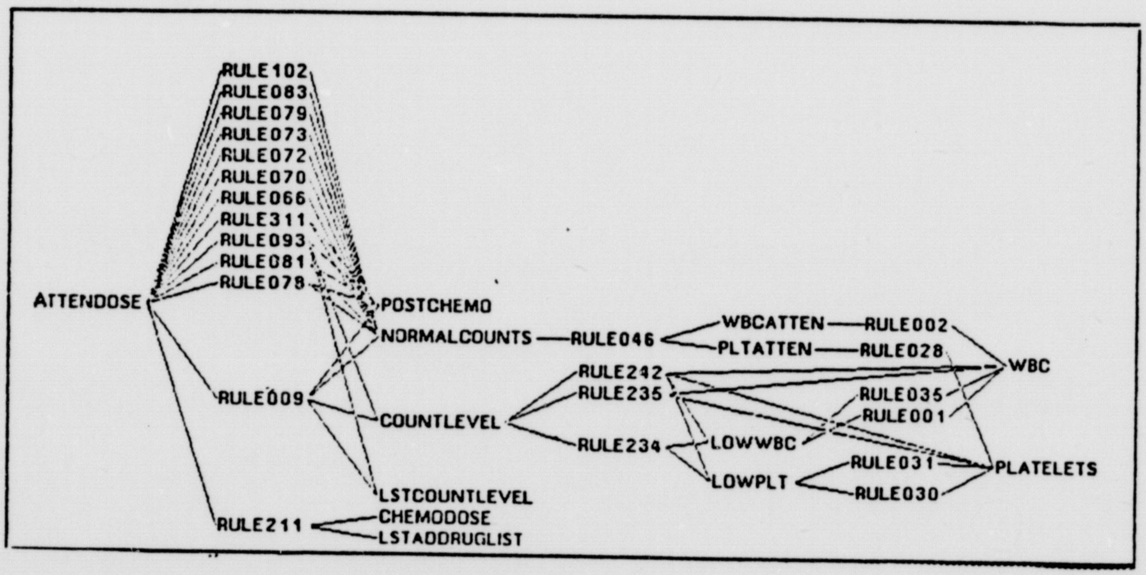

Figure 4

the parameter Attendose where the context is defined to be *adriamycin in CHOP for patients with any lymphoma.* The translation of any rule in the network may be obtained by selecting that rule with the mouse (Fig. 5). Rule translations appear in a separate window on the screen with the graphical display of Fig. 4 remaining for reference.

It is worth noting three important aspects about the impact this system. First, the user has control over the amount of detail that is to be displayed and this control is *unobtrusive and intuitive.* Second, a variety of representations of the same information increases one's understanding of the material. Depending on the problem at hand, the system builder may wish to rely strictly on the diagrams or to alternate attention between text and graphs. For a novice system builder this approach provides a comprehensive overview of the system which is hard to glean from written documentation. Third, windows provide a means to segment information so that the information content doesn't become overwhelming. Judicious placement of windows can be used to compare and contrast the form of the information presented.

Conclusion

Our early experiments with a professional workstation will proceed with an expansion of the query system to answer simple questions about protocol management. Work will also begin on the development of a tool, to be used by a system debugger during a consultation, that will provide graphical simulation of the reasoning chain and extensive use of windows for information display along with conventional debugging facilities. In addition, we are adapting the graphical techniques for the consultation program itself and will soon begin experiments to assess the response of physician-users to this new kind of interactive style.

There is a clear need for system building tools that will take advantage of the capabilities of workstations. Graphics can enhance text explanations with diagrams, present a comprehensive view of the system as well as detailed descriptions of individual facts, and provide a mechanism for focusing attention on different aspects of an explanation. Development of tools on personal workstations has only begun to be explored. The potentials are enormous.

Acknowledgements

This work has been supported by the National Library of Medicine under Grants LM-03395 and LM-00048, and by the Office of Naval Research under contract NR 049-479. Computing facilities were provided by the SUMEX-AIM resource under NIH Grant RR-00785.

References

[1] Shortliffe, E.H., Scott, A.C., Bischoff, M.B., Campbell, A.B., VanMelle, W., and Jacobs, C.D.
ONCOCIN: An Expert System for Oncology Protocol Management.
In *Seventh Proceedings of the International Joint Conference on Artificial Intelligence*, pages 876-892. Vancouver, British Columbia, August, 1981.

[2] Langlotz, C.P. and Shortliffe, E.H.
Adapting a Consultation System to Critique Physician's Therapy Plans.
Technical Report HPP-83-2, Stanford University, 1983.

[3] Shortliffe, E.H.
Computer-Based Medical Consultations: MYCIN.
New York: Elsevier/North Holland Publishing Company, 1976.

[4] Shortliffe, E.H.
Hypothesis Generation in Medical Consultation Systems: Artificial Intelligence Approaches.
MEDINFO 83, 1983.
in press.