Generic expert systems

Richard Ottolini

An *expert system* is a particular software methodology for managing information. Often touted as a method of capturing the knowledge of human experts in a computer, expert systems are likely to be more common as interfaces between the computer and large systems of programs or databases. This programming style arose in artificial intelligence research and is rapidly becoming a routine tool for software engineers.

A *generic expert system* is a set of software aids for constructing an expert system. They come in two major varieties. One is a previous expert system stripped of its application dependent guts and generalized for other uses. The other is a set of add-on software to existing programming languages, usually LISP, in order to implement the functions of expert systems. A number of these generic expert systems are available from universities and commercial software houses and will be described later on in this document.

Overview

One way of defining expert system software is to give some characteristic ways of how they differ from other software systems. Expert systems are particularly good at handling non-numerical, or verbally presented information. They are also good for encoding computational procedures which are not cut-and-dried, but heuristic, judgemental, and uncertain in nature. Expert systems attain this flexibility by not encoding how-to-do-it information in a procedural, step-by-step program, but instead by dynamically configuring information stored in a general *knowledge base* while the expert system is solving a problem.

Expert system software can also be described by its major components:

• They contain a *knowledge base manager.* A knowledge base contains facts about the particular situation at hand, knowledge of how to deal with these facts, and meta-knowledge about how to deal with knowledge. The knowledge often is encoded as a set of if-then rules, but can also include taxonomies, tables, sub-routines, demons, or what ever else is useful for encoding knowledge. An illustrative example is a 'family relations' expert. The facts would include a family tree of mother-father relations and the sex of individuals. The knowledge would include a vocabulary of relations such as brother, daughter, uncle, etc. The knowledge base would also know how to deduce such family relations from the given facts. For example: 'If two children has the same parents, then they are siblings'. Knowledge about knowledge would include common sense things such as 'If someone is a brother to a male individual, then the converse is true'. Meta-knowledge would also contain information on how to reason. For example, nuclear family relations would be nearby on a family tree and far away limbs could be ignored.

- An expert system also contains a reasoning system, often called an *inference engine*. This may be as simple as following if-then rules or a substantial implementation of logical calculus. The efficiency of an expert system is often rated by its inference engine in units of 'logical inferences per second' (LIPS). The inference engine may contain many strategies to control the flow of reasoning. For example, one control of flow is called forward deduction, that is 'given these facts, what are the conclusions?' Backward deduction asks 'given this hypothetical conclusion, do the facts support it?' These strategies could be hardwired in or the user might put them into the knowledge base as meta-knowledge. The type of control strategy also depends upon the structure of the knowledge and type of problem at hand.

- An important part of the inference engine is the *scoping mechanism*. No computer is powerful enough to relate all combinations of facts and knowledge for a realistically sized expert system. There are too many different kinds of scoping mechanisms to do other than give an example here. One scoping mechanism is to separate the knowledge into groups such as particular diseases or mineral deposits in diagnostic expert systems. Another mechanism is to limit the hypotheses at hand with the use of an 'agenda' or 'blackboard'.

- Last, but not least in importance is the *user interface*. Since the domains of expert systems are mostly verbal, English language-like user interfaces are important. Interfaces include ways of displaying the reasoning history and explaining its reasoning. Interfaces also include the ability to easily modify the information base.

The best format of these various components is a hot current research topic by software engineers. Some trade-offs are in terms of efficiency versus functional power. Other trade-offs are dependent upon the particular application for the expert system, much in the same way a programmer chooses between COBOL, FORTRAN, or LISP.

Applications in geophysical data processing

I see several applications:

—As a manager for large systems of data processing programs. What is the best program to use next? How should we select the parameters? How good are the results from a processing viewpoint? A computer scientist at Bell Labs is developing such a system for an already existing statistical program package.

—As an intelligent and friendly user interface to large geophysical and geological databases.

—As a geological interpretation assistant, or automated interpretation of reflection seismic data. This expert system would coordinate and evaluate numerical, graphical, geological, and experiential knowledge for geological interpretation.

Of course there are many more applications in petroleum exploration outside of geophysical data processing, such as in data acquisition or drilling. A couples of examples are self-configuring machines and dynamic equipment manuals.

Some available generic expert systems

A complete catalog of expert systems is nearly impossible, due to rapid growth and change in this field. Stanford computer science Professor Feigenbaum made an attempt in his book *The Fifth Generation* and so did a recent issue of *IEEE Computer* magazine. Likewise, an accounting of all the generic forms are difficult. I will mainly restrict this discussion to those available through the *Heuristic Programming Project* of the Stanford computer science department ($500/system). Some generic expert systems include:

- EMYCIN is the distilled essence of a medical diagnosis expert system developed in the mid-1970's. Because it has been around a long time, it is fairly bug-free and efficient. A lot of work has gone into developing good user interfaces for this system. On the negative side it is not very flexible in the choices it offers to the expert system developer regarding methods of knowledge representation and reasoning strategies.
- AGE is the distilled essence of sonar interpretation expert system. It is of the 'blackboard' model control strategy described earlier. This tends to give it a certain efficiency in deciding how to proceed to a solution. Also it is one of the successful expert systems in the signal processing domain, a important component of geophysical data analysis.
- MRS is a smorgasbord of expert system functions built upon the LISP programming language. It has been used to model computer circuits and adjudicate legal conflicts. It has a more developed logical calculus than the other two generic systems and a very

flexible choice of knowledge representations. The price of this flexibility, of course, is decreased efficiency. However, the hope is that improved reasoning implementations may be able to compensate for brute-force LIPS. I have been using this system on our VAX for more than a year. The main exercise in the A.I. programming methods class taught at Stanford is to write a simple version of MRS.

Some non-Stanford generic expert systems include:

- OPS is a basic expert system written at Carnegie-Mellon in Pittsburg. Digital Equipment Company has used it to build an expert system which configures VAX computer systems. Rumor has it that there is a version of this system in PASCAL.
- The LOOPS system is LISP superset containing expert system functions running on Xerox LISP computers. It has been used for circuit designs and self-debugging hardware. Xerox computer workstations probably have the best user interfaces around.
- The PROLOG computer language, developed in English universities, may become widely available for many computers due to its adoption by the Japanese Fifth Generation Computer Project. It contains many of the elements of an expert system. It has a well developed logical calculus which also tends to slow it down. Early versions were also lacking good scoping mechanisms and user interfaces which tended to decrease its popularity in this country.

Some cautions: Most of the better experts systems are written in the LISP language, though some of the A.I. companies may have or soon will have PASCAL or FORTRAN versions. The trade-off is a loss of functional power for increased efficiency. In the development stage of an expert system it may be better to use a LISP version.

Another problem is a shortage of LISP implementations and major incompatibilities between various implementations, particularly in the case of efficiency functions and I/O. There are LISP systems for DEC 10's and 20's, UNIX VAX's, and dedicated minicomputers such as Symbolic's and Xerox's. Rumor has it that a member of the HP 9800 series has a very good LISP implementation for the cost of the machine, and that LISP will be available on VMS VAX's and some IBM machines.

My expert system plans

After studying a few simple expert systems I am currently looking at a geological application. This application would model the reasoning of field geologist as it makes deductions from field observations. Basically it would know about stratigraphic columns, simple structural geology, and cartographical relationships. This is contrasted from two other geological expert systems which include one that finds mineral deposits mainly based conclusionary information given to it by an economic geologist (Stanford Research Institute) and another system which tries the capture the concept of geologic *process* in terms of spatial and temporal relationships (Simmons at MIT). In the long term I am interested in the automated interpretation of geophysical data. This problem encompasses many levels of information representation– geophysical data, signal detection, imaging, and geology– much in a manner similar to speech understanding. I am not particularly interested in the geophysical data processing applications which attempt to build better user-computer interfaces because most of this work is relatively routine.

References for further reading

The clearest introduction to expert systems is Feigenbaum's book on the fifth generation computer project. Another good introduction is the quarterly journal of the American Association for Artificial Intelligence, *AAAI Magazine*, which reviews work in A.I. centers around the world. Some of the better detailed information about expert systems is given in the three volume *Handbook of Artificial Intelligence* and the A.I. memo series of the MIT, Stanford, and Xerox artificial intelligence laboratories.

***** At the time this article was going to press a new book called** ***Building Expert Systems*** **edited by Hayes-Roth et. al. and published by Addison-Wesley appeared in the bookstores. It seems to be the best book on the subject. Chapter 9 describes some generic expert systems. *****