Department of Computer Science and Engineering
University of California at San Diego CSE 255
Winter 1998

Comments on expert system proposals from 1997

Choosing an application domain

One team proposed an expert system to verify designs for printed circuit boards (PCBs). This system would extend existing computer-aided design (CAD) software for printed circuit boards. For such a system, it is important to ask why it does not already exist on the market. In general, if the need for an in-house technical improvement can be identified, the fact that this need has not yet been filled does not imply that filling the need is difficult. On the other hand, if an apparent opportunity for a commercial product or service exists, yet no existing company is taking advantage of the opportunity, this fact suggests strongly that filling the need is more difficult than apparent at first.

Typically the most successful expert systems have carefully limited domains of applicability. They can thereby have relatively greater depth of competence. For example, design verification for arbitrary printed circuit boards is probably too large a domain. However the PCB domain is a good choice compared to VLSI design for example, because the number of potential users is relatively large, and the domain is probably less technically complex than the VLSI domain.

One report was unclear about whether the proposed diagnosis system would diagnose multiple faults or not. If the cost of faults or the cost of troubleshooting is high, then diagnosis software should not make a single-fault assumption, but should rather detect all possible faults in one session. Of course, this increases the complexity of the expert system.

Knowledge representation issues

In any domain, the issue of "brittleness" must be addressed. Will the proposed expert system be able to recognize when a new problem is outside its scope of competence? In particular, simple decision trees always give an answer for every situation, so systems based on these can be ery brittle.

Another issue that must be addressed in many domains is the treatment of knowledge that is not purely discrete and qualitative. One team proposed to represent textures in images as single floating point numbers. Is this really adequate? Another team proposed standard forward-chaining rules to deal with the situation where "the designer had just incorrectly located a "via" hole (for interlayer connection) too close to the edge of the [printed circuit] board". Can such geometrical knowledge really be represented with CLIPS-style rules?

Knowledge representation and reasoning also become much more difficult when the expert system must integrate information received over multiple time periods. One team proposed an expert system to recognize faults in metal welds, as imaged by a a computer vision system. Presumably some faults cannot be diagnosed from a single image, but require observations from multiple consecutive images to be combined. This requires the expert system to maintain "state" information from previous images while analyzing the most recent image.

Knowledge engineering and maintenance

When written documentation is available about task that an expert system will perform, this indicates that the knowledge required can be formalized. Moreover the documentation often constitutes a good starting point for the knowledge engineering process. However almost certainly additional knowledge will need to be captured, because humans always use general knowledge and problem-solving skills to fill in gaps in written documentation.

Maintenance is a critical issue for all software, including for knowledge-based software. One team made the good suggestion that "Users [will] be encouraged to submit their best cases . . . to be evaluated and included in periodic updates. This procedure will be modeled after Microsoft's successful practice of including selected news group articles into the help module of Visual C++."

One of the most important maintenance issues is transferability to new environments. In high-technology industries the typical lifespan of a product is two years or less. Design and manufacturing technology must therefore be transferable across multiple product generations. One team proposed an expert system for troubleshooting buggy cellular telephone handsets as they come off the production line. To what extent would the knowledge in such a system be reusable for new models of handsets?

Who should be responsible for knowledge engineering and maintenance is also a difficult question. One team wrote "A specialist software engineer should be responsible for maintaining the ES and updating and adding to its knowledge base as required. Nobody else should be allowed to change [it.]" But for a comparison system called HIDAT they wrote contradictorily that "[Its] authors attribute this success mainly to the fact that HIDAT was developed by the testing and diagnostic experts themselves." There is usually a tension between having a knowledge base maintained by a specialist knowledge engineer versus by experts and users themselves, with advantages and disadvantages on both sides.

Evaluating costs and benefits

All teams included monetary estimates of the costs and benefits of their proposed systems. However there were several common weaknesses in these estimates.

- Most important, the basic assumptions behind most estimates were not explained. Why should a particular expert system require 12 months rather than (say) 18 months to develop? Why are two knowledge engineers needed, no more and no fewer? One team referred to a comparison system that "was not yet quite complete at the time the paper was written, and it had already cost them 5 man-years to develop and consisted of over 150 rules." One hopes that most projects can do better than 30 rules per year.

- There was hardly any discussion of the need for resources other than money. In particular, if experts must be made available for interviewing, where will these experts be found, and can they be spared from their current work?

- Cost estimates should distinguish between direct costs and costs including overhead, also known as indirect costs. Direct costs are mainly salaries and equipment. Overhead is typically calculated as a percentage of direct costs, and covers all ancillary services such as secretarial support, general administration, office space, and utilities.

- Last but not least, it is important to use general accounting concepts only if they are appropriate for the situation at hand. For example, software usually has no salvage value, and because software is intangible, it is difficult to treat its cost as a capital improvement to be depreciated over time.

Notes on writing

It is not sufficient for a report just to have a bibliography or list of references. In addition, each article or book in the bibliography should be cited at the specific places in the text of the report where it is relevant. If appropriate, sections or chapters of books should be cited. The point of a citation is to allow the reader to find out more about a specific topic, or to find the sources for a specific claim. In the bibliography, there should be enough information for the reader to be able to find a copy of each source cited. In particular, if an entry does not contain the name of a well-known journal or publisher, an address or other contact information should be given. This applies especially to unpublished reports.

In any report of more than a few pages, pages and top-level sections should be numbered, and the report should have a table of contents. This allows readers to skip to the parts of the report that they are interested in. It also allows the writers of the report to reflect consciously on whether the report is well-balanced. Are some sections too long? Are some needed sections missing? The lowest level sections of a report should be paragraphs. Having headings for paragraphs is unnecessary, but paragraphs should still exist as logically coherent units shorter than a page.

Be consistent in using either the present tense or the past tense to describe something that has already been created. Similarly, be consistent in using either the present or future tense to describe plans. In general, it is easiest to use the present tense.

Underlining should not be used in documents formatted with multiple fonts. Italics should be used instead. In titles, prepositions, conjunctions, and articles (i.e. words like “of”, “or”, and “the”) that are not the first word of the title should not be capitalized.

Acronyms and abbreviations should be used very sparingly. Avoid acronyms that just stand for a very generic phrase, such as “Data Mining and Diagnosis Tool” and avoid acronyms that have unfortunate other meanings, such as DDT.