Legal liability involving use of expert systems in business environments

William N. Bockanic, Ph.D., J.D.
Boler School of Business
John Carroll University

Marc P. Lynn, Ph.D.
Boler School of Business
John Carroll University

Abstract

In the simplest scenario, development of expert systems is typically accomplished by a knowledge engineer (KE) working with a domain expert (DE) to develop a knowledge base that is accessed via an inference engine during a consultation with a user. The basic role of the DE is to provide rules (including heuristics), facts, goals, etc. required for the decision making process within the context of the problem domain. The basic role of the KE is to work with the DE in order to extract that information and appropriately enter or codify it into the knowledge base. While this may sound simple, it is often quite a challenge.

The most basic requirements to perform the tasks associated with the role of KE include the following:

- Interpersonal communication skills,
- Some level of knowledge and understanding of the issues and goals of the final functioning product,
- Insight into the way the knowledge base should be processed from the user's point of view.
- Insight into the way the knowledge base will be processed from the "computer's point of view,"
- Knowledge of the language(s) and software environment as well as the hardware environment and associated system-oriented constraints and limitations.

In a way, the KE's role in relation to the DE and the end product, (the functioning expert system), is somewhat analogous to being untrained in medicine, anatomy, and physiology, and being asked to design a robot to perform an operation on a patient by getting instructions over the telephone from a qualified surgeon. How much must the KE know about the problem and solution domain? How can accurate and effective communications with the DE be ensured and monitored? How can the testing of the system be best accomplished to guarantee that the actual goals are being met? (After all, just getting the individual messages right doesn't necessarily mean success).

As newer Expert System Shells are being developed that make it easier for users to create rules and develop knowledge bases, it is likely that more domain experts will be willing to try their hands at building expert systems themselves. While this may seem like a good thing, increasing efficiency of the process by eliminating much, if not all of the role of the system

engineer, it may also lead to serious problems that include additional legal liabilities of the domain expert as well as other members of the organization.

The domain expert's expertise rarely includes knowledge of knowledge base structure and design, inference engine characteristics, forward and backward chaining, or user interface issues. While many newer shell environments make it easier to work with the development process, the domain expert remains the primary link in the chain, and is now responsible for not just providing rules, but entering them in appropriate ways, ensuring proper specification of variables, attributes, goals, confidence factors and eventually, testing.

One may ask why the DE should care about such things. Isn't the DE just responsible for providing the rules? The rules themselves may be individually challenged and evaluated. But when multiple DEs are involved in development of a knowledge base (KB), the interaction of the various rules causes the whole to be greater than the sum of the parts. How can responsibility be assigned for the result of such interaction rather than as the result of a single rule attributable to one DE?

Maintaining control over the environment in which the KB is used, the integrity of the KB, and the updating or altering of the KB once it is in the users' hands could be very difficult for the DE. Why should the DE care if the system is now in the control of the users? If the expertise in the KB is "misused" would it not be the users' problem or responsibility? Possibly not! If the DE has not clearly identified the limitations and constraints of the system, the way it is to be used, and the environment for which it is specifically designed to function, the DE may be liable for negligence or malpractice, and possibly breach of contract. A major problem may be the alterations a user makes to the system after it is delivered. If the system is altered by introducing a new rule, the overall performance of the system may be affected. If damage occurs, the DE may not be able to completely disavow all responsibility.

If the environment of the system or the policies or laws change after the system has been delivered, is the DE responsible for continually re-evaluating, testing, and recertifying the system? For example, an attorney may be liable for malpractice if he fails to notify a client of a change in the law affecting an estate plan which he prepared. If the old estate plan is not revised to reflect a subsequent change in the law, the client may incur greater estate taxes, and the attorney may be found negligent for failure to notify the client of the change in the law. What if the DE is no longer in a contractual relationship with the business for which he/she prepared the expert system? Would there still be a legal obligation to notify the business owner of changes that would adversely affect the system's function or efficacy? Could the DE potentially be liable to third parties who were damaged as a result of defects in the system's operation?

While economic injuries are not ordinarily recoverable under a strict liability in tort theory, if the expert system was defectively designed or constructed so as to be unreasonably dangerous to the ultimate user or consumer and caused physical injury and/or property damage, this theory may be stated as a cause of action in any ensuing litigation, as well.