Specification of Knowledge

Rasiah Loganantharaja
Automated Reasoning Laboratory
Center for Advanced Computer Studies
USL, PO Box 44330
Lafayette, LA 70504

(Abstract)

Before we look into how specification of knowledge helps us to solve the validation and verification problem, let us explore why the basic problems of V&V of expert systems still remain vary challenging. Even though there are different opinions on how V&V should be defined, many researchers within expert system community accept the definition of V&V as originally defined in software engineering. That is, an expert system is said to be *validated* if it can be shown that it performs or behaves according to the requirements. On the other hand, an expert system is said to be *verified* if it can be proven that the 'programming portion of the expert system' meets its specification. Here, by the 'programming portion of an expert system' we mean the set of rules, or frames, or whatever the specific representation used to encode the knowledge. The *specification* is then a restatement of the requirement in the view of the developer of an expert system.

Once we accept to go along with the definition of V&V, then there is a clear need to study and to understand the problem of representing the requirements and the specifications in an unambiguous manner if not formally. As has been noticed elsewhere, the lack of advancement in this area makes the practitioners to believe and act as if testing is the only means of solving the V&V problem. Testing of expert system is more challenging than testing of a procedural program because of non determinism in the underlying execution mechanism. The challenges include deciding when to stop generating hypothetical situations. Recently, there is a new approach in software engineering to generate automatically all possible data sets to test the program using mutation techniques. Unfortunately the prospects of applying similar technique to expert system is not very good. What are the other alternatives available for us to go forward to solve the V&V problems of expert system?

We feel that researchers should get excited in developing and studying methodology and languages to represent requirements of an expert system and to specify the relevant knowledge and the problem solving methods. Recently there are few attempts to specify expert systems [Yen et al. 93, Yost et al. 89, Slagle et al. 90]. Yen et al. [Yen et al. 93] describe task based methodology for specifying expert systems. In their approach the the knowledge is intertwined with the problem solving method and hence the ability to share the knowledge with other applications is quite limited. This approach has the flavor of the knowledge acquisition tools developed by Mc Dermotte et al. [McDermott et al. 85, McDermott et al. 86, McDermott 86].

We take a different approach in which knowledge is specified in terms of its semantics and the roles it plays. Along with the specification of the knowledge, we specify the problem solving method that integrates the various pieces of knowledge to solve the problem. We are in the process of developing a translator that transform the specified knowledge to the code of the specified target language, such as, CLIPS, KEE or ART etc. Suppose, we want to solve a diagnostic problem of a certain kinds of diseases, we would have to represent the relevant causal knowledge, domain knowledge, and the relevant background knowledge. The problem solving method in this example is heuristic classification. Knowing the target language, the translator should be able to generate the set of rules or frames as appropriate. The advantage of this approach is that there is a clear separation of knowledge from the problem solving methods and thus make if easier to share the knowledge with any other applications. Further verification need not to be done for each expert system. What is required is to verify the translator only once.

How about validation?? It is a different story. We have to check how the requirements are translated into the specifications. If there is a formal method to verify that translation, then this problem is also solved. We have to address the semantic gaps between the different levels: requirements to specifications and specifications to the target codes. It is worthwhile studying the problem. This problem is somewhat easier than that of a problem in automatic programming because in automatic programming there is no assumption of the algorithm, but fortunately the problem solving methods that integrates the knowledge is known during the synthesis process. We are very optimistic that we can make some reasonable progress in this line of research.

References

[McDermott 86] J. McDermott. Making expert system explicit. In *Proceedings of the IFIP Congress*, Dublin, Ireland, 1986.

[McDermott et al. 86] J. McDermott S. Marcus. Salt: A knowledge acquisition tool for propose and revise systems. Technical report, Carnegie-Mellon University, Department of Computer Science, 1986.

[McDermott et al. 85] J. McDermott S. Marcus and T. Wang. Knowledge acquisition for constructive systems. In *Proceedings of the IJCAI*, 1985.

[Slagle et al. 90] J. R. Slagle and D.A. Gardiner. Knowledge specification of an expert syustem. *IEEE Expert*, 5(4):29–38, 1990.

[Yen et al. 93] J. Yen and J. Lee. A task-based methodology for specifying expert systems. *IEEE Expert*, 8(1):8–14, 1993.

[Yost et al. 89] G. Yost and A. Newell. A problem space approach to expert-system specification. In *Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI 89)*, pages 621–627. Morgan Kaufmann, 1989.