AI Magazine Volume 13 Number 4 (1992) (© AAAI)

Book Review

Verification, Validation, and Test of Knowledge-Based Systems

Alun D. Preece

Verification, Validation, and Test of Knowledge-Based Systems, Marc Ayel and Jean-Pierre Laurent, eds., John Wiley and Sons, Chichester, England, 1991, 219 pp., $49.95, ISBN 0-471-93018-0 (paper).

This volume contains a selection of 14 papers presented at the First European Workshop on Verification, Validation, and Test of Knowledge-Based Systems, held during the 1990 European Conference on Artificial Intelligence (ECAI 90) in Stockholm, Sweden. In reviewing a book of this kind, it is necessary to answer three questions: (1) how important is the workshop topic, (2) how valuable are the included papers, and (3) how coherent is the volume as a whole? I address each question in turn.

In the last decade, knowledge-based systems (KBSs) emerged from being a research subfield within AI to become an application software technology. Although many specific aspects of knowledge acquisition, representation, and reasoning remained active research topics, the methods and tools required to build useful and powerful KBS applications had become sufficiently well understood to facilitate the development and delivery of systems in many diverse domains. However, as organizations began to use the technology, concerns arose about the reliability of KBSs. In a nutshell, the problem was that nobody knew how to evaluate these systems because nobody had done it before.

People involved with the problem of evaluating KBSs included researchers and developers from both the AI and software engineering communities. There was general agreement that many aspects of the KBS evaluation problem were both different from, and more difficult than, the problem of evaluating conventional software. One interesting issue concerned functional validation of KBS, the need (for many applications) to establish that KBS performs at a level comparable to human experts. Another issue concerned structural validation of KBS, given that the architecture of these systems (having separate knowledge base and inference engine components) was different from conventional imperative programs.

Once the demand for KBS evaluation techniques was established, studies were launched to determine the applicability of software engineering evaluation techniques to KBSs and to develop new techniques specific to KBSs. Several such studies were initiated by organizations that were starting to use KBSs heavily, including the National Aeronautics and Space Administration, the U.S. Department of Defense, and the European Community. In 1988, the first North American workshop on KBS verification, validation, and testing (VVT) was held at the Eighth National Conference on Artificial Intelligence, and North American workshops have been held annually since then. In addition, many recent conferences and workshops have devoted special sessions to the topic. Much of the important literature in this area has recently been collected in Culbert (1990) and Gupta (1990). These collections are especially valuable because many of the original articles are hard to obtain.

In this context, Ayel and Laurent's collection, *Verification, Validation, and Test of Knowledge-Based Systems*, is timely and valuable, making a considerable amount of European work in KBS VVT widely available for the first time. The 14 papers (with authors from 6 European countries plus 1 from the United States) fairly reflect the foci and degrees of emphasis of active researchers. Nine of the papers are primarily concerned with the development of automatic VVT tools for KBS, two present formal (nonautomatic) VVT techniques; one considers formal properties of KBS errors, one provides guidelines for performing VVT within the KBS life cycle, and one is a survey of current VVT practices.

The survey of 80 KBS developers in financial domains by Daniel O'Leary opens the collection and raises some interesting points. VVT is revealed to be a significant concern, with developers often exceeding their allotted VVT budgets. The greatest concern was with the completeness and correctness of knowledge bases. Testing with real and contrived test cases was found to account for about half the overall VVT effort on average, with direct inspection of the knowledge base accounting for another 30 percent in the survey. Revealingly, 98 percent of the VVT effort on average was performed with informal methods; there was almost no reported use of automated tools or formal analytic techniques. These findings are interesting in view of the volume's emphasis on tools and formal methods and its relative lack of emphasis on testing techniques.

Following O'Leary's survey, the paper by L. F. Pau on the KBS life cycle is a disappointment; although it espouses a basically sound software engineering methodology for KBS development, it is far too brief to do justice to the ideas presented. The lack of examples leaves the reader wondering how realistic the approach is for KBS.

Knowledge base verification, focusing chiefly on the detection of redundancy and inconsistency (also called *incoherence* in this volume), is the most common theme among the papers on tools and formal techniques. Jaak Tepandi presents a theoretical analysis of redundancy and subsumption in hybrid knowledge bases (with both rules and simple frames), including the complexity of detecting these properties. Although reasonably thorough, the paper would have been better if it had considered the relationship between these structural properties and functional errors exhibited by the knowledge base. For example, under what circumstances is subsumption likely to represent an error on the part of a KBS developer, and when is it intentional?

The paper by Evelyne Charles and Olivier Dubois on their MELODIA system goes further, presenting formal definitions of structural anomalies in knowledge bases written in an extension of propositional logic as well as results from using an automatic tool to detect the anomalies in six real KBS applications. The performance of the tool is impressive, but no indication is given of whether the detected anomalies represented serious errors in practice.

Incoherence anomalies are also detected by the SACCO tool presented in Marc Ayel and Jean-Pierre Laurent's paper. SACCO operates on a more general knowledge base language than MELODIA and uses heuristics that concern the importance of errors in certain parts of the knowledge base to focus the search for anomalies, thereby controlling potential combinatorial explosion. The basic SACCO verification mechanism is also used by the SYCOJET tool to generate KBS test cases.

MELODIA, SACCO, and SYCOJET are primarily intended to operate on knowledge bases that are largely complete; in contrast, the approach presented by Thomas Hoppe concerns incremental validation as part of the knowledge-acquisition process.

Philippe Lafon demonstrates that a knowledge base can benefit from a greater degree of verification when certain properties of the predicates within the knowledge base are specified a priori. The idea is to permit stronger checking of the semantics of the knowledge base; many of these properties appear similar in spirit to those used within the CYC system (Lenat and Guha 1990).

All the aforementioned tools focus on VVT of domain knowledge expressed chiefly in the form of heuristics. Although this use reflects the predominant area of work in the field, other approaches are considered in the volume. A technique for validating control knowledge is described by Beatriz Lopez: The CONKRET system (developed as part of the ESPRIT VALID Project on KBS VVT) compares problem-solving strategies generated by metarules with "gold standard" strategies obtained from domain experts to identify deficiencies. Vincent Guibert, Alain Beauvieux, and Marc Haziza present an example of the application of VVT techniques to a model-based diagnosis KBS. Philippe Mazas acknowledges that KBSs for design applications will often contain contradictory knowledge out of necessity, the issue being *robustness* (ability to solve contradictions) rather than freedom from contradiction. He presents an interesting automated testing system that is capable of proposing tests involving contradictions and evaluating the effectiveness of the system in resolving them.

Although the majority of the papers in the collection are concerned with qualitative analyses of KBSs, the articles by Sorin Grunwald and Susan Craw take a quantitative view: Sorin Grunwald's approach uses a model of information in a probabilistic KBS to identify the potential for failure based on the reliability of evidence; Susan Craw uses a metric for blame assignment to drive a testing-and-refinement procedure. Luc DeRaedt, Gunther Sablon, and Maurice Bruynooghe also describe a knowledge base refinement procedure that is based on interactive concept learning, using test cases generated by the system as positive and negative training examples.

Possibly the most unusual paper in the collection describes the use of a variant of the Z specification language to precisely define the required functions of a KBS inference engine to permit proofs that the inference engine will perform correctly. Such a technique complements the verification techniques for knowledge bases offered in other papers; such techniques assume but do not verify certain properties of the inference engine.

The most significant achievement of this book is its comprehensiveness: The breadth of topics covered is impressive. Also, the technical quality of the work is among the highest yet presented in this relatively youthful field. On the whole, the papers are clearly written, and the standard of the language is high.

The book has a number of minor problems, however. The structure appears somewhat arbitrary: The implied relationships between the papers in each section are not properly explained, leaving readers to infer for themselves the overall direction in which the field is heading. It would also have been nice to see some discussion and comparison of the approaches described in the papers. The effectiveness of many of the tools and techniques is hard to judge because of the general lack of application studies. Their origin as workshop presentations notwithstanding, some of the papers could have been expanded further for inclusion in this book to do the ideas full justice. Furthermore, although the coverage is impressive, several important aspects of KBS VVT are neglected, specifically, the application of software engineering techniques to KBSs (most of the methods described originate in the AI community) and human issues in VVT (the coverage is biased toward technical issues). Finally, there is some minor inconsistency and incompleteness in the bibliographies of the papers. In particular, many articles are hard to obtain, and much North American work is omitted. A separate annotated bibliography would have been of great value. Some of these problems can be attributed to the immaturity of the field itself.

In conclusion, this volume should be considered required reading for researchers in the area in addition to the earlier collections of Culbert (1990) and Gupta (1990). It provides an excellent snapshot of some of the best work in an important and emerging field. As such, it is also recommended for all interested parties in both the AI and software engineering communities.

References

Culbert, C. 1990. *Expert Systems with Applications* (special issue on verification and validation) 1(3).

Gupta, U. G. 1990. *Validating and Verifying Knowledge-Based Systems.* Los Alamitos, Calif.: IEEE Press.

Lenat, D. B., and Guha, R. V. 1990. *Building Large Knowledge-Based Systems.* Reading, Mass.: Addison-Wesley.

Alun D. Preece is a research associate in the Centre for Pattern Recognition and Machine Intelligence, Concordia University, Montreal, Canada, where he is developing tools and methods for the formal development and evaluation of knowledge-based systems. He received his Ph.D. from the University of Wales College Swansea in 1989 for his work in knowledge-based system verification.