Achieving Comprehensiveness in Verifying Hybrid Systems
Extended Abstract

Rose F. Gamble
Department of Mathematical and Computer Sciences
University of Tulsa
600 S. College Ave.
Tulsa, OK 74104
gamble@utulsa.edu

1. Introduction

Rule-based systems have been the traditional foundation for knowledge-based systems (KBSs). In this respect, research in verification and validation of KBSs has focused primarily on rule-based systems (Preece & Suen, 1993; Culbert, 1990; Gupta, 1991; Murrell & Plant, 1997; O'Leary, 1994). Some researchers have extended the verification and validation concepts to KBSs that comprise other components, such as rules, frames, objects, and methods (Vermasan 1996, Lee & O'Keefe 1993, Mukherjee & Gamble 1995, Mukherjee, Gamble, Parkinson 1997, O'Leary 1989). These systems are called *hybrid* KBSs and can be developed using KBS shells such as Kappa-PC[1] and CLIPS[2]. We have built a tool called KBS-DetectOR that statically detects traditional verification criteria, along with subsumption as it relates to object inheritance and monitor-rule interactions as used with active frames (Stiger et al. 1997). However, there are other research avenues and implementation considerations that must be explored in order for the tool to be comprehensive across multiple hybrid KBSs. In this extended abstract, we outline the research directions to complete KBS-DetectOR.

2. KBSDetectOR

The current implementation of KBS-DetectOR is in C/C++ on a UNIX workstation (Stiger et al. 1997). Motif is used to provide a graphical user interface. The system is currently separated into two parts as shown in Figure 1. The component labeled *Subsumption and Rule Anomalies* implements the subsumption algorithms first devised by Lee and O'Keefe (1993) and then extended by Mukherjee and Gamble (1995). These algorithms incorporate the traditional verification of rules in the knowledge base that uses an object framework for working memory where inheritance can cause additional problems. The component labeled *Monitor-Rule Interactions* implements the algorithms discussed in (Muhkerjee, Gamble, Parkinson 1997) that focus on the interference a monitor can cause during rule-based reasoning. A monitor is a method that is attached to an object attribute, such that the monitor is executed as a side-effect of accessing or changing the value of the object attribute.

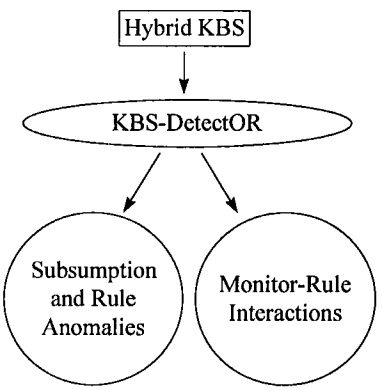

Figure 1: Current Components of KBS-DetectOR

The *Subsumption and Rule Anomalies* component is complete because the work in this area has sufficiently matured. The *Monitor-Rule Interactions* component has been implemented with some restrictions placed on the monitors. The component recognizes (1) when monitors can change matched conditions in a rule before the rule executes and (2) when monitors can alter the actions of a rule by changing an object the rule action depends on or changes itself. However, the component requires further research into more complex monitor structures. For instance, loop structures in monitors currently cannot be handled. In addition, monitors cannot be correctly analyzed using the current algorithms if they invoke rule-based reasoning. Such invocation is required

[1] Kappa-PC is a product of Intellicorp, Mountain View, CA.

[2] CLIPS is distributed by COSMIC, University of Georgia, Athens, GA.

if the hybrid KBS were to mimic a blackboard or agent infrastructure.

3. The Need for Additional Components

As shown in Figure 2, other verification components have been identified as requirements for the KBS-DetectOR to reach comprehensiveness. With the growth of object-oriented programming comes the additional burden of verifying object correctness. The component should focus on verifying the structure of the objects via static analysis. In addition, this verification criteria examines the working memory portion of the hybrid KBS, which traditionally has been overlooked. Because verifying the object structure is similar to verifying the frame structure when separated by the monitors, they are grouped into the same component.

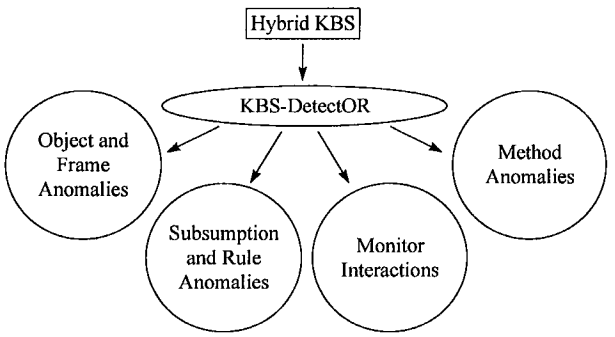

Figure 2: Additional Components for KBS-DetectOR

O'Leary (1989) has established domain-independent verification criteria for frames and semantic nets. These criteria have direct analogies to the traditional criteria established for rule-based programs. For example, *consistency* is a criteria that requires the rules in the KBS to be non-conflicting. For frames, "consistency is a concern with the names given to the frames, frame slots, and contents" (O'Leary, 1989). Algorithms to detect these criteria according to the definitions given by O'Leary are needed prior to implementation.

The *Method Anomalies* component requires the most effort to build. There are procedural code verification algorithms that may be encoded (Sommerville, 1995). However, with respect to monitors embedded as side-effect functions in object attributes, there is little to no research on what should be statically detected and how. Certainly, circularity is one verification criteria that would need to be examined in which one monitor changes an attribute that subsequently invokes another monitor to change the attribute to which the first monitor was attached. This criteria would be similar for message passing to methods. However, the manner in which redundancy, consistency, and other verification criteria for monitors are to be defined requires further research.

4. Conclusion

This extended abstract outlines the components that are needed to fully verify a hybrid KBS using KBS-DetectOR. Some components have established research and implementation details, other components have established research but not implementation, and one component has little to no research. As the research to extend KBS-DetectOR continues, additional components are likely to be needed to allow for blackboard and agent infrastructure implementations within the existing KBS shells that produce hybrid KBSs.

Acknowledgement

This research is sponsored in part by the US Department of Energy, contract #DEAC22-93BC14894.

References

(Preece & Suen 1993)
A. Preece and C. Suen (Eds.), Special Issues: Verification and Validation of Knowledge-Based Systems, *International Journal of Expert Systems,* Vol. 6, No. 2-3, 1993.

(Vermesan 1996)
A. Vermesan, A definition of subsumption anomalies in conceptual models of object-oriented KBSs. *AAAI-96 Workshop on Verification and validation of KBSs, 1995.*

(Culbert 1990)
C. Culbert (Ed.), Special Issue: Verification and Validation of Knowledge-Based Systems, *Expert Systems with Applications,* Vol. 1, No. 3, 1990.

(Gupta 1991)
U.G. Gupta (Ed.), *Validating and Verifying Knowledge-Based Systems,* IEEE Computer Society Press, Los Alamitos, CA, 1991.

(Lee & O'Keefe 1993)
S. Lee and R.M. O'Keefe. Subsumption anomalies in hybrid knowledge based systems, *International Journal of Expert Systems*, 6(3):299-320, 1993.

(Mukherjee & Gamble, 1995)
R. Mukherjee and R. Gamble. Critical examination of subsumption anomalies in hybrid KBSs, *IJCAI-95 Workshop on Verification and validation of KBSs*, 1995.

(Mukherjee, Gamble, & Parkinson 1997)
R. Mukherjee, R. Gamble, and J. Parkinson, Classifying and detecting anomalies in hybrid

knowledge-based systems, *Decision Support Systems,* forthcoming 1997.

(Murrell & Plant 1995)
S. Murrell and R. Plant, A survey of tools for validation and verification, 1985-1995, *Decision Support Systems,* forthcoming, 1997.

(O'Leary 1989)
D.E. O'Leary, Verification of Frame and Semantic Network Knowledge Bases, *5th AAAI Sponsored Knowledge Acquisition for Knowledge-Based Systems Workshop,* Banff, Canada, Nov. 1989.

(O'Leary 1994)
D. O'Leary (Ed.) Special Issue: Verification and Validation of Intelligent Systems: Five Years of AAAI Workshops, *International Journal of Intelligent Systems,* Vol. 9, No. 8-9, 1994.

(Stiger et al. 1997)
P. Stiger, R. Gamble, H. Christensen, and R. Plant, KBSDetectOR: Automating anomaly detection in hybrid KBSs, *FLAIRS-97,* May 1997.

(Sommerville 1995)
I. Sommerville, *Software Engineering, 5th Edition,* Addison-Wesley, New York, 1995.