# Validation of Blackboard Systems: On the Order of Knowledge Sources



**Daniel E. O'Leary**
3660 Trousdale Parkway
University of Southern California
Los Angeles, California 900891421

oleary@rcf.usc.edu

213-740-4856

**Abstract**

There have been few results regarding unique issues of verification and validation of blackboard-based systems. This paper attempts to mitigate this limitation in the literature by focusing on a particular blackboard architecture.

In solution-based focusing blackboard systems, problem solving starts by specifying that one of the knowledge sources be instantiated and then iteratively different knowledge sources, typically based on order, are chosen for assistance in developing a solution. Unfortunately, depending on the nature of the particular blackboard system, the specific order in which knowledge sources are chosen can have a major impact on the quality of initial solutions developed. As a result, this paper is concerned with a priori ordering of the knowledge sources so that better initial solutions will be generated. This paper develops some analytical results that can be used to establish partial orderings of the knowledge sources and rules within the knowledge sources, using different relationships between knowledge sources, for economic based systems (i.e., systems where there are costs or revenues associated with the outcomes in the rules).

## 1. Introduction

Adrion et al. (1982) indicate that "validation is the determination of the correctness of the finalprogram or software produced from a development project with respect to the user needs and requirements." O'Keefe and O'Leary (1993) suggest that "Validation is ... concerned with the quality of the decisions made by the system."

Review of the literature (e.g., O'Keefe and O'Leary, 1993) finds little mention of verification and validation issues for blackboard systems. Since the quality of the solution plays a critical role in validation, this paper focuses on some of the unique concerns associated with validation of blackboard systems.

Blackboard systems use multiple knowledge sources in the same system. There are a number of different blackboard architectures to control the processing of knowledge sources (e.g., Hayes Roth 1985). Those approaches include the standard blackboard control with solution based focusing, standard blackboard control with sophisticated schedulers, metalevel architectures and the Blackboard Control Architecture. The purpose of this paper is to investigate the impact of knowledge source order, on the quality of the initial solution developed using a solution based focusing approach.

### Solution Based Focusing

Solution based focusing has received substantial attention in the blackboard literature over the years, including Nii and Aiello (1979), Nii et al. (1982), Hayes Roth (1985) and Jagannathan (1989). In the solution based focusing approach, the order that is instantiated is a function of the order that information appears on the blackboard and the order of the knowledge sources (Hayes Roth 1985). As noted by Jagannathan (1989, p. 87), "problem solving is started by specifying one knowledge source to run first, which is instantiated and run to create one or more events." The knowledge source that is chosen to initiate the process, may come to a solution or provide information to the blackboard so that other knowledge sources find a solution. Thus, unless the blackboard system ultimately elicits all feasible solutions (which generally is not the case and generally is not practical), control over the order of which knowledge sources are instantiated, can be a critical issue. As a result, the concern in this paper is with ordering the knowledge sources in order to generate a good initial solution.

### Importance of a Good Initial Solution

The knowledge sources and the rules in those knowledge sources ultimately will be ordered in some manner.

This paper suggests that they be ordered so that good initial solutions are found. Finding good initial solutions does not limit the incremental refinement typically associated with blackboard systems. Instead a good initial solution provides us with a good starting point for further analysis. A good initial solution can limit the extent to which we must incrementally alter the solution. In addition, in some real time systems there may be a need to stop after we obtain an initial solution. Further, in some situations the incremental refinement of solutions may be slow, ineffective or inefficient. In those two situations, it is critical to find good initial solutions.

### Multiple Processor and Single Processor Environments, and Granularity

Order is a concern whether the blackboard system is developed for a single processor or multiple processor environment (e.g., Ensor and Grabbe 1985). In both environments, systems can only execute as many knowledge sources as they are processors available. Thus, if there are more knowledge sources than processors, then there will be order effects, determining which knowledge sources are processed in what order. In either a single or multiple processor environment where there are more knowledge sources than processors there is a need to establish when to change control of the processor. The control of the generation of events in a blackboard system can be accomplished at a number of different levels of granularity, two of which are discussed here. The first model (model 1), is not interrupting the processing of knowledge sources until the knowledge source either provides a solution or finds it cannot provide a solution. A knowledge source locks up a processor until it is A knowledge source locks up a processor until it is done. Such an approach is not unusual in the use of multiple human experts. For example, one expert may be given the opportunity to solve a medical problem. If that expert does not solve the problem then another expert is pursued. Second (model 2), is the model of interrupting knowledge sources periodically after different transactions, e.g., if a rule under investigation at that knowledge source fails, we interrupt. At this level of granularity, the knowledge source may be interrupted prior to finding a solution. After the determination of the failure of a rule, a choice is made between the next knowledge source or the next rule in the current knowledge source. One basis of choosing would be a search to determine which rule(s) offers the potential for finding the best solutions. This is analogous to allowing an expert to determine if an idea does or does not work. Once it is determined that one idea does not work then choice must be made to stay with the current expert or choose another.

### Findings and Contribution of this Paper

This paper has two primary findings. First, results are presented that indicate when the processing of one knowledge source should precede some other knowledge source. Second, in those same situations, results are developed that indicate which level of granularity should be used. The results presented in this paper are analytic. Whereas most research on blackboard systems has been system construction and heuristic analysis of those systems, this paper presents results that, in theory, can be applied to any appropriate system.

### Plan of this Paper

This paper proceeds as follows. Section 2 provides an example that is used to illustrate the concepts generated in the paper. That section also discusses some of the assumptions of the research. Section 3 provides some background and notation. Section 3 summarizes some of the relevant issues in blackboard systems used in this paper. Section 4 investigates the choice between different knowledge sources. Section 5 analyzes when relationships between knowledge sources allow us to choose which level of granularity that should be used. Section 6 briefly summarizes the paper and also discusses some additional research issues and heuristic uses of the results discussed in this paper.

## 2. An Example

An example will be adopted to illustrate various concepts in the paper. The example is a version of the classic "Bartender" problem from Winston (1984), revised to accommodate the notion of multiple knowledge sources. It is assumed that the rules are grouped in three different knowledge sources: Wine Knowledge; Beer Knowledge; and Health Drink Knowledge. A selected set of rules from that example is as follows:

1. Wine Knowledge Source

*r(1,1)If expensive wine is indicated it is New Year's Eve Then Bond's Champagne*

*r(1,2) If expensive wine is indicated entree is steak Then Chateau Earl of Bartonville Red* *r(1,3) If cheap wine is indicated entree is chicken guest is not well liked Then Honest Henry's Apple Wine*

*r(1,4) If cheap wine is indicated entree is unknown Then Toe Lakes Rose*

2. Beer Knowledge Source

*r(2,1) If beer is indicated entree is Mexican Then Dos Equis*

*r(2,2)If beer is indicated Then Coors*

3. Health Drink Knowledge Source

*r(3,1)If guest is a health nut Then Glop*

*r(3,2)If guest is a health nut carrots are not served with the meal Then carrot juice*

### Order Makes a Difference

Suppose that the health drink knowledge source is chosen first. In that case, if the client is a health nut then "glop" is a feasible solution. The system would stop, unless all solutions were found. Alternatively, suppose that the wine knowledge source was chosen first. If it is also New Year's Eve and expensive wine is called for, then Bond Champagne is suggested by the system. Assuming that Bond Champagne is more profitable to the bartender than glop, it can be seen that having the wine knowledge source instantiated first would be the greatest benefit to the company. Order makes a difference.

### Scope of the Research

This paper is concerned with blackboard systems designed to solve problems of economic consequence or the equivalence of economic consequences. In particular, it is assumed that the consequences. In particular, it is assumed that the system is designed to assist in the solution of a problem with a decision that has economic impact. For example, in the bartender problem there was the choice of drink, which had economic return to the bartender (or the bar). Throughout it is assumed that the rules in the system have consequences for which a cost or benefit or both can be established. Further, it is assumed that economic information is the basis of the decision making. For example, throughout it is assumed that, a priori, if two alternatives are equally feasible, then the alternative with the largest return is more desirable than the alternative with less return. In addition, this paper assumes that the knowledge sources are rule based. This is, in part a matter of convenience. The results in this paper can be generalized to other forms of knowledge representation, as long as there is an economic consequence associated with the ultimate consequent of the equivalent rule.

## 3. Notation and Background

This section provides some notation used later in the paper, a brief summary of classic inference process, for rule bases, used in the proofs, later in the paper, and a brief summary of some of the key blackboard system concepts used in this paper,.

### Notation

Assume, for purposes of presentation, that each rule is of the type "if condition 1, ..., condition m, then consequence t(w)," where t(.) is a set of W consequences. The purpose of the system is to find a feasible solution from that set of consequences. These rules are flexible and used in a variety of applications (e.g., Greene 1987). Alternate rule forms could be investigated as extensions of this paper, but are beyond the scope of this paper. Generally, it is not practical to elicit all feasible solutions. As a result, it is assumed that the system develops a single solution. If a rule in a knowledge source is investigated by the system and each of its conditions is true then the consequence produced by that rule is the system solution. Once a solution is found, then the initial solution has been found and iterative analysis of that solution can begin. However, if the system is a real time system then processing might stop with the initial solution. Suppose there are n different knowledge sources ks(i), for i= 1, ..., n. Assume that each of the knowledge sources is a rule base with q(i) rules. Assume that each rule j in knowledge source i is referred to as r(i,j). Assume that the return associated with r(i,j) is ret(i,j). Assume that it is desired to find a solution where ret(i,j) is a maximum. Let the set of conditions used in r(i,j) be c(i,j). Let the entire set of conditions used in ks(i) be denoted s(i), so that s(i) is the union of c(i,j) over all j.

### Inference in Each Knowledge Source

It is assumed that each of the knowledge sources uses an inference engine that searches the rules in a knowledge source in the following manner. (This approach is consistent with the classic expert system approach used in MYCIN (Buchanan and Shortliffe 1985). Let A(i) be the set of active conditions in knowledge source i. Initially, A(i) equals the null set. Each rule in ks(i) is called active, false or true. If a rule has its entire set of conditions in A(i) and each condition is true then the rule is true and the computation stops. If at any time a rule has any conditions in A(i) that are not true then the rule is not true. If a rule has only a subset of its conditions in A(i) then the rule is considered active. In the case where a single knowledge source is being processed, the inference process is assumed to start with the first condition of the first rule to determine whether it is true or false. If it is false then the inference process would go to the next rule where that previous condition (and all other conditions in A(i)) was (were) found to be true. If no such rule exists then inference would go to the next rule where there were no active conditions. If no such rules existed then computation would stop.

### Blackboard Models

The blackboard model (e.g., Nii 1986) includes partitioned knowledge sources, which are kept independently. Nii (1986, 1989) indicates that often those knowledge sources are represented as collections of rules, such as those used in the above example. Knowledge sources are responsible for producing changes to the blackboard, incrementally developing a solution. Communication between different knowledge sources takes place solely through the blackboard. Nii argues that ultimately, deciding which knowledge source to apply, becomes a problem of "control." This paper is concerned with the standard blackboard architecture with so called solution based focusing (e.g., Hayes Roth 1985), as exemplified by Nii and Aiello (1979) and Nii et al. (1982). In particular, Hayes Roth (1985) notes,

*... solution based focusing relies upon a complex program that embodies all of a system's control knowledge. ... it sequentially selects specific blackboard events and executes knowledge sources triggered by each one. In some implementations all triggered knowledge sources execute in a predetermined sequence; in others, the focusing program uses other aspects of the current solution to determine which subset of triggered knowledge sources execute and in what order.*

### Impact of Ordering on Processing Time

There is no a priori reason to assume that any specific order of the knowledge sources will require greater processing time that any other order. If we assume that the information requirements of any situation are equally likely then no order will always be best or worst in terms of total processing time. Thus, it generally would be desirable to use an ordering that provides us with the good initial solution. For example, assume the ordering of the knowledge sources is as in the example, with wine knowledge source first and health drink knowledge last. If it is New Year's Eve, then the optimal strategy in the example is to serve the Bond's Champagne. Given the ordering, then that would be the initial solution found by the system. However, if the guest is a health nut and carrot are not served, then the last rule would be the one where the solution is ultimately found. A priori, it is impossible to specify which situation will occur. Thus, in this paper it is assumed that we search for an order that provides a good initial solution.

## 4. Ordering Knowledge Sources

Consider the example discussed above. If each of the wines provides a greater profit than either the beers or the health food drinks then we would expect that the company would prefer any feasible wine solution rather than either beer or a health drink. That finding is the basis of the following.

Definition 1: ks(i) will be said to dominate ks(j) if each solution found with ks(i) ordered before ks(j) is at least as good as those with ks(j) ordered before ks(i). Given this definition of knowledge source domination, it can be shown that domination is transitive. Thus, if ks(i) dominates ks(j) and ks(j) dominates ks(k) then ks(i) dominates ks(k). Transitivity can be used to reduce computational effort in the determination of domination results. Domination provides a basis on which the order of different knowledge sources can be established. In general, if knowledge source i dominates knowledge source j, we would prefer to find a solution in ks(i). One such result is summarized in the following theorem.

*Theorem 1:* If Ret (i,r) > Ret (j,s) for all r and s, then ks(i) dominates ks(j).

*Proof:* If Ret (i,r) > Ret (j,s) for all r and s, then choosing any feasible alternative in ks(i) will result in at least as good a solution as the choice of any alternative in ks(j). In the case of a single processor, this result indicates that the knowledge source i would be processed prior to knowledge source j. In the case of multiple processors, this result indicates that knowledge source i would be processed no later than knowledge source j. The existence of domination is readily apparent in many real world cases. For example, in many restaurants, the sale of virtually any wine produces a greater return than beer or health drink. Thus, waiters typically try to determine if patrons would order wine, before some mutually exclusive drink is sold. As another example, if an automotive dealer carries both Cadillac and Chevrolet, then, in general the profit associated with a Cadillac would outweigh that of the Chevrolet. Thus, we would expect the Cadillac knowledge source would dominate the Chevrolet knowledge source. As a result, it is not unusual with a visit to a new car dealership to be directed to the most expensive car in the showroom.

### A Domination Result for Model 2 Granularity

Next it will be assumed that the control granularity of model 2 is used. Periodic change of knowledge source is built into model 2. For theorem 2 the following assumptions are made. Assume that there are three knowledge sources i, j, and k (the result can be extended to arbitrarily larger sets of knowledge sources). Suppose that the conditions in some ks(i) can be partitioned into two sets 1 and 2. In addition, suppose that the rules in ks(i) have conditions in either set 1 or set 2 but not both. Assume there are two sets j and k, such that for all rules in j, all the conditions are in set 1 and all the conditions for the rules in k are in set 2. Assume that the rules are ordered so that all the rules with conditions in set 1 are ordered before those in set 2. Assume that the largest return is associated with r(i,1). As long as the first rule is from set 1 there is no loss of generality, with the inference engine described above.

*Theorem 2*: Given the structure of ks(i), ks(j) and ks(k), if those rules in ks(i) such that all conditions are in set 1 have a return greater than those in set ks(j) and if those rules in ks(i) such that all conditions are in set 2 have a return greater than those in set ks(k), then ks(i) dominates ks(j) and ks(k).

*Proof:* There are four cases to consider.

1) Success in ks(i) with conditions in 1. This means that a solution in ks(j) was not chosen, but that is optimal since the return of each rule in ks(i), with conditions in 1, exceeds that of ks(j). This means that a solution in ks(k) was not chosen. That would occur only if the return on the successful rule in ks(i) was greater than those rules in ks(k) that were not chosen. Otherwise, the control would have chosen a rule from ks(k). As a result, processing ks(i) first did not have a negative impact on the quality of the solution.

2) Success in ks(i) with conditions in 2. This means that a solution in ks(k) was not chosen, but that is optimal since the return of each rule, with conditions in 2, in ks(i) exceeds that of ks(j). This means that a solution in ks(j) was not chosen. That would occur only if the return on the successful rule in ks(i) was greater than those rules in ks(j) that were not chosen. Otherwise, the control would have chosen a rule from ks(j). As a result, processing ks(i) first did not have a negative impact on the quality of the solution.

3) Success in ks(j). This means that no rule in ks(i) with conditions in 1 led to a solution, since otherwise the solution would have come from there. As a result, processing ks(i) prior to ks(j) did not have an impact on the solution.

4) Success in ks(k). This means that no rule in ks(i) with conditions in 2 led to a solution, since otherwise the solution would have come from there. As a result, processing ks(i) prior to ks(k) did not have an impact on the solution. This theorem says nothing about the ordering of knowledge sources j or k. However, it does provide a condition for ordering knowledge source i relative to j and k. Thus, it provides a partial ordering of the knowledge sources. In the case of a single processor environment, knowledge source i would be processed before j and k. In the case of a multiple processor environment, i would be processed no later than either j or k.

## 5. Choice of Granularity Approach

The results of the previous section can be used to establish which control structure granularity model would be best used. In theorem 3 a situation for which using model 1, with no interruption of the knowledge source, is found to be optimal. Whereas in theorem 4, a situation for using periodic interruption of the knowledge source is appropriate.

*Theorem 3*: (Model 1 Uninterrupted Knowledge Source) Assume that each knowledge source has either been (A) fully examined and no solution was found or (B) has not yet been examined. If for some i, ks(i) dominates all other j in B, then use model 1 (uninterrupted knowledge source) for ks(i) will find at least as good a solution as model 2.

*Proof:* Assume that a solution is found in knowledge source i then since ks(i) dominates the other knowledge sources that solution is at least as good as any other solution. Assume that no solution is found in ks(i). In that case it would not impact the quality of the solution to examine ks(i) prior to the other knowledge sources. In most other situations a periodic review of the quality of the potential solution is required. In the following, a periodic approach is seen to be optimal for the general case.

*Theorem 4:* (Model 2 Interrupt After Each Rule Failure) In general, it is optimal to interrupt knowledge source inferencing at rule failure to determine if some other knowledge source has a better solution.

*Proof:* If we interrupt at rule failure then that allows us to choose the rule with the largest profit that has not yet been fully analyzed. If we do not interrupt at rule failure and then the inference process could continue possibly examining rules whose outcomes lead to nonoptimal solutions, since there is no aspect of the inference process designed to determine the quality of the decision being made by the system. This approach requires simply polling the This approach requires simply polling the knowledge sources for rules that are active or not yet examined for the rule with the largest return. This is easy to implement in a decentralized environment: simply require each knowledge source be ready to provide that information on inquiry.

## 6. Summary

There has been little in the verification and validation literature associated with blackboard systems. This research has been concerned with validation issues associated with unique characteristics of blackboard systems. In particular, this paper examines the importance of ordering knowledge sources in solution-based focusing. The paper finds significant order effects. The results could be extended by investigating other blackboard architectures.

## References


Adrion, W., Branstad, M., and Cherniavsky, J., "Validation, Verification and Testing of Computer Software," ACM Computing Surveys, Volume 14, number 2, pp. 159-182.

Buchanan, B. and Shortliffe, E., Rule Based Expert Systems, Addison Wesley, Reading, Massachusetts, 1985.

Ensor, J. and Grabbe, J., "Transactional Blackboards," International Joint Conference on Artificial Intelligence, 1985, pp. 340-345.

Greene, D., "Automated Knowledge Acquisition: Overcoming the Expert System Bottleneck," Proceedings of the International Conference on Information Systems, 1987.

Hayes Roth, B., "A Blackboard Architecture for Control," Artificial Intelligence Journal, Volume 26, pp. 251-321, 1985.

Jagannathan, V., Dodhiawala, R., and Baum, L., Blackboard Architectures and Applications, Academic Press, Boston, 1989

Jagannathan, V., "Realizing the Concurrent Blackboard Model," in Jagannathan, V., Dodhiawala, R., and Baum, L. (1989), , pp. 85-97, 1989.

Nii, H., "Blackboard Systems," AI Magazine, Volume 7, Number 2, 1986, pp. 38-53 and Number 3, pp. 82-106.

Nii, H., "Introduction," pp. xixxxix, in Jagannathan, V., et al. (1989).

Nii, H., and Aiello, N., "AGE (Attempt to Generalize): A Knowledge based Program for Building Knowledge based Programs," in Proceedings Sixth International Joint Conference on Artificial Intelligence, Tokyo, Japan, 1979, pp. 645655. Also published in Blackboard Systems, edited by R. Engelmore and T. Morgan, Addison Wesley, 1988, pp. 251-280.

Nii, H., Feigenbaum, E., Anton, J., and Rockmore, A., "Signal to symbol Transformation," AI Magazine, Volume 3, 1982, pp. 23-35. Also published in Blackboard Systems, edited by R. Engelmore and T. Morgan, Addison-Wesley, 1988, pp. 135-158.

O'Keefe, R. and O'Leary, D., "Expert System Verification and Validation," Artificial Intelligence Review, Volume 7, pp. 3-42, 1993.

Winston, P., Artificial Intelligence, Addison-Wesley, Reading, Massachusetts, 1984.