DESIGN AND ANALYSIS TOOLS FOR
CONCURRENT BLACKBOARD SYSTEMS

John W. McManus

Mail Stop 489
NASA Langley Research Center
Hampton, Virginia 23665

<u>Summary</u>

A blackboard system consists of a set of knowledge sources, a blackboard data structure, and a control strategy used to activate the knowledge sources. The blackboard model of problem–solving is best described by Dr. H. Penny Nii of the Stanford University AI Laboratory:

> "A Blackboard System can be viewed as a collection of intelligent agents who are gathered around a blackboard, looking at pieces of information written on it, thinking about the current state of the solution, and writing their conclusions on the blackboard as they generate them."[1]

The blackboard is a centralized global data structure, often partitioned in a hierarchical manner, used to represent the problem domain. The blackboard is also used to allow inter-knowledge source communication and acts as a shared memory visible to all of the knowledge sources. A knowledge source is a highly specialized, highly independent process that takes inputs from the blackboard data structure, performs a computation, and places the results of the computation in the blackboard data structure.

This design allows for an opportunistic control strategy. The opportunistic problem–solving technique allows a knowledge source to contribute towards the solution of the current problem without knowing which of the other knowledge sources will use the information. The use of opportunistic problem–solving allows the data transfers on the blackboard to determine which processes are active at a given time.

Designing and developing blackboard systems is a difficult process. The designer is trying to balance several conflicting goals and achieve a high degree of concurrent knowledge source execution while maintaining both knowledge and semantic consistency on the blackboard. Blackboard systems have not attained their apparent potential because there are no established tools or methods to guide in their construction or analyze their performance.

The lack of a coherent set of design tools and guidelines has resulted in blackboard systems being developed in an ad hoc fashion. Designers have been forced to make design choices in a void, in many cases leaving the designer stuck with a poorly designed system because revisions are too difficult once the system has been implemented. This lack of design and analysis tools is one of the reasons why incorporating concurrency into the blackboard problem–solving model has not been successful. The use of a centralized control mechanism and contention in accessing the blackboard have also restricted the success of previous systems.

A set of blackboard system design tools that can be used to test and refine a proposed blackboard system design before the design is implemented has been developed and validated . The set of blackboard system design tools consists of a Knowledge Source Organizer, a Knowledge Source Input/Output Connectivity Analyzer, and a validated Blackboard System

Simulation Model. This paper presents the structure and functionality of the Knowledge Source Input/Output Connectivity Analyzer. An example outlining the use of the Knowledge Source Input/Output Connectivity Analyzer to aid in the design of a concurrent tactical decision generator for air–to–air combat is presented. The blackboard system design and analysis tools are designed for generic blackboard systems and are application independent.

Introduction

Blackboard systems are a natural progression of expert systems into a more powerful problem–solving technique. They provide a way for several highly specialized knowledge sources to cooperate to solve larger and more complex problems. Blackboard systems incorporate the concepts developed by rule-based and expert systems programmers and include the ability to add conventionally coded software to cooperate in solving problems. The specialized knowledge sources are easier to develop and test and can be hosted on hardware specifically suited to the problem that they are solving. The use of opportunistic problem–solving, the data–driven control structure, and highly specialized knowledge sources allows a set of knowledge sources to cooperate to solve large, complex problems.

Designing and developing blackboard systems is a difficult process. The designer is trying to balance several conflicting goals: achieving a high degree of knowledge source parallelism, and maintaining memory and semantic consistency. Several major problems facing the designers of blackboard systems have been outlined in the current literature. [1,...,4]

1)   Speculative parallelism can be very costly and adversely affect system performance.

2)   Memory management and memory coherence can become difficult and time consuming.

3)   Process locking and concurrency control overhead can adversely affect opportunistic problem–solving.

4)   Real-time systems require predictable response times.

5)   Overly restrictive concurrency control can restrict the parallel execution of knowledge sources.

6)   Dynamically allocating and deallocating knowledge sources to achieve good processor utilization and load balancing can be time consuming, and the overhead required may outweigh the gains of parallel execution if the knowledge sources are small or execute swiftly

Blackboard systems have not attained their apparent potential because there are no established tools or methods to guide in their construction and analyze their performance. Daniel Corkill states:

> "At present (1989), intuition is the primary criterion for making an appropriate set of design choices. Application characteristics, hardware and communication characteristics, and how the application is implemented are all likely to strongly affect the performance of a parallel or distributed blackboard application.
>
> What is needed are system engineering guidelines for making appropriate design choices and improved languages and tools for constructing parallel and distributed blackboard applications. Such engineering guidelines require empirical measurements of pioneering applications. A few data points are beginning to emerge, but direct

measurement of applications designed and executed on real parallel or distributed hardware remains to be performed." [2]

Unified Approach to System Design and Analysis

A blackboard system designer requires a coherent set of blackboard system design, simulation, and analysis tools to be able to build systems that realize the full potential of the blackboard problem–solving model. A system implemented using a set of highly specialized, highly independent knowledge sources with a data transfer driven control mechanism will be faster, more efficient, and more powerful than current systems. This type of blackboard system does not require the centralized blackboard controller. Centralized control modules have been shown to be one of the major bottlenecks in existing concurrent blackboard systems.[1,...,4]

To achieve these design requirements, a set of design and analysis techniques for blackboard systems has been developed. The set of design aids consists of a Knowledge Source Organizer and a Knowledge Source Input / Output Connectivity Analyzer. A validated Blackboard System Simulation Model and a set of blackboard evaluation metrics have been developed to analyze the performance of blackboard systems.

A set of automated Blackboard System Design and Analysis Tools is being developed to support the design and analysis of concurrent blackboard systems. The first section of the design and analysis tool aids the designer in decomposing the problem domain into a set of knowledge sources and outlines the input/output pairs for each knowledge source. An outline of the structure of the blackboard data objects is also developed at this time.

The second section of the tool uses the input/output pairs for the initial knowledge source partitioning to create a data connectivity graph of the system. The data connectivity graph is then used to measure knowledge source independence, data transfers across the blackboard, and commonly shared data objects. A simplified data connectivity graph is shown in figure 1.

The knowledge source I/O connectivity analysis system then searches for tightly coupled knowledge sources (using a heuristic to measure coupling) and highlights those knowledge sources by directly connecting the knowledge sources and marking the link with a star (figure 2). The knowledge sources can then be restructured to increase the independence of the knowledge sources. This process can be repeated until the system design constraints are met, or until the knowledge source coupling metric is minimized

Preliminary Research

Preliminary research [3,..6] has shown that efficient implementation of blackboard systems requires that the knowledge sources be highly specialized and highly independent. A knowledge source is considered to be specialized if it is designed to solve a specific task or subtask. A knowledge source is considered independent if it requires data from the blackboard for execution, not from a specific knowledge source.
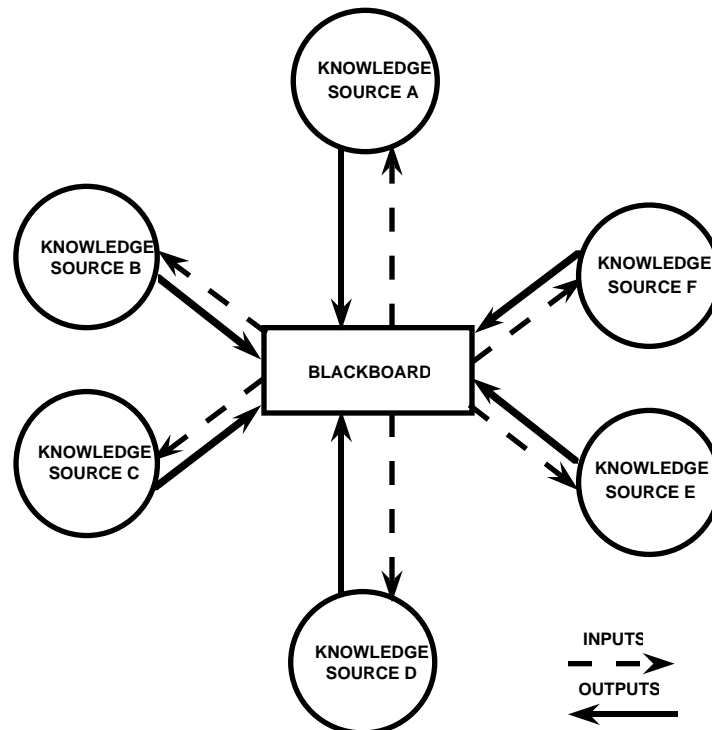
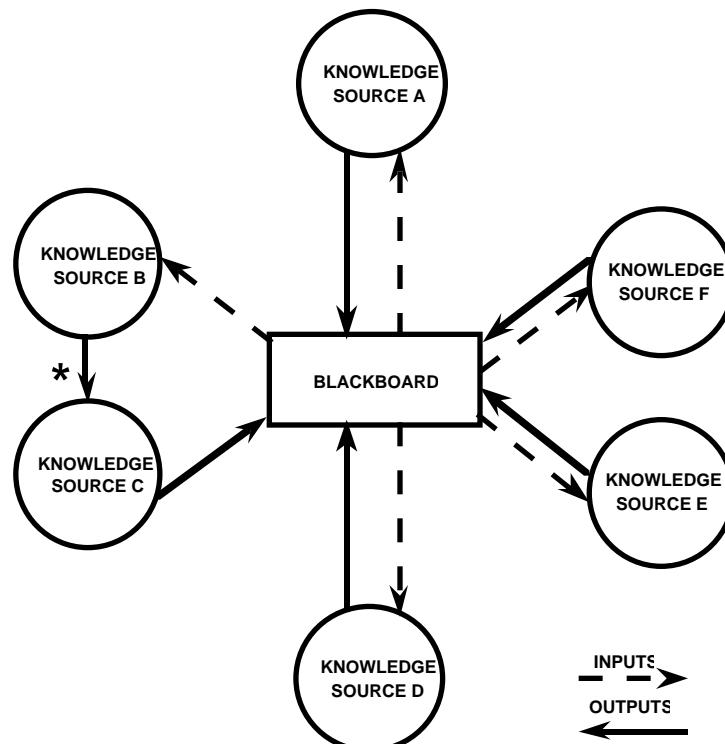**Figure 1. Initial Knowledge Source Connectivity Graph.**

Figure 2. Updated Knowledge Source Connectivity Graph.

Knowledge source specialization provides the following benefits:

- Improves performance of rule-based and expert systems. Specialization results in smaller rule bases and allows for the use of meta-knowledge and meta–rules to guide rule–partition selection, which results in fewer rules being active and faster system performance.[2]

- Provides for smaller knowledge sources that solve specific problems. The smaller, specialized knowledge sources are easier to develop and test since each knowledge source solves a small part of the total problem.

- Leads to systems that can cooperate in parallel on large problems.

- Reduces input and output overlap.

- Reduces write set conflicts and concurrency control requirements.

- Reduces the chance of semantic synchronization being violated because it reduces the chance of one knowledge source's output invalidating the output of another knowledge source.

- Reduces memory contention at the blackboard level.

Knowledge source independence provides the following benefits:

- Centralizes problem–solving functionality, making the function of each knowledge source easier to understand and provides for modular protection.

- Increases opportunistic execution of knowledge sources.

- Reduces the input/output connectivity between the knowledge sources.

- Reduces inter-knowledge source data transfers across the blackboard.

- Reduces data traffic on the communications medium.

- Reduces concurrency control requirements.

- Reduces blocking of knowledge sources.

- Reduces memory contention on the blackboard.

Knowledge Source I/O Connectivity Analysis

A static knowledge source Input/Output connectivity analysis of the preliminary design of a blackboard system's knowledge sources and blackboard structure is used to determine the data transfers between the knowledge sources, data migration across the blackboard, and knowledge source independence. Knowledge source I/O connectivity analysis measures the "connectivity" between knowledge sources and the overlap in the knowledge source output sets. The I/O connectivity is a measure of the independence of knowledge sources, and the output set overlap is a measure of the specialization of knowledge sources. Knowledge source pairs with a high level of I/O connectivity are highly dependent on each other, pairs with a low level of I/O connectivity are independent of each other. Knowledge source pairs with a high level of output set overlap compute similar functions and are not specialized. Knowledge source pairs with a low level of output set overlap are highly specialized. Knowledge source independence and specialization values are used to refine the design of blackboard systems.

To perform the static knowledge source input/output connectivity analysis for a blackboard system we treat the blackboard system β as a set of knowledge sources.

$$\beta = \{ks_1, ks_2, ..., ks_i\} \qquad (0)$$

For each knowledge source ks_j in β we can form a set Ψ_j containing all of the input variables of ks_j; and an output set Φ_j containing all of the output variables of ks_j. Ψ_j consists of both the blackboard inputs and the external inputs. If ks_j is a sensor, Ψ_j may contain only external inputs. Note that for any knowledge source ks_j if $\Phi_j = \phi$ the design is invalid since the knowledge source does not update the blackboard.

$$\Psi_j = \{iv_1, iv_2, ..., iv_n\} \qquad (1)$$
$$\Phi_j = \{ov_1, ov_2, ..., ov_m\} \qquad (2)$$

Once Ψ_j and Φ_j have been formed for all ks_j in β we can compute the sets $\Gamma_{j,k}$ and $\Theta_{j,k}$ for all knowledge source pairs $\{ks_j, ks_k\}$ in β ($j \neq k$).

$$\Gamma_{j,k} = \Phi_j \cap \Phi_k \qquad (3)$$
$$\Theta_{j,k} = \Phi_j \cap \Psi_k \qquad (4)$$

The cardinality of the set $\Gamma_{j,k}$ for each pair $\{ks_j, ks_k\}$ in β is a measure of output overlap for the pair $\{ks_j, ks_k\}$. Knowledge source pairs, $\{ks_j, ks_k\}$, with a large output overlap imply that ks_j and ks_k share a large number of output variables and perform similar functions. Knowledge source pairs, $\{ks_j, ks_k\}$, with a low overlap imply that ks_j and ks_k perform different functions. A proposed heuristic to measure knowledge source independence is to compute an independence value, $\Omega_{j,k}$ for each pair $\{ks_j, ks_k\}$ in β.

$$\Omega_{j,k} = \frac{card(\Gamma_{j,k})}{min(card\left(\Phi_j\right) card\left(\Phi_k\right))} \qquad (5)$$

This heuristic was developed to compute the percentage of overlap between the sets Φ_j and Φ_k. The cardinality of the set $\Gamma_{j,k}$ divided by the minimum of the cardinalities of the sets Φ_j and Φ_k computes a percentage of overlap between the set $\Gamma_{j,k}$ and the smaller of the sets Φ_j and Φ_k. As $\Omega_{j,k}$ approaches 1.0 the output overlap between ks_j and ks_k increases, as $\Omega_{j,k}$ approaches 0.0 the output overlap between ks_j and ks_k decreases. For the limiting cases, if $\Phi_j \supset \Phi_k$ or $\Phi_k \supset \Phi_j$, we know that $\Omega_{j,k} = 1.0$, and ks_j and ks_k compute the same outputs and are not specialized. If $\Gamma_{j,k} = \phi$, we know that $\Omega_{j,k} = 0.0$, and the two knowledge sources have no common outputs and are highly specialized.

The cardinality of the set $\Theta_{j,k}$ for each pair $\{ks_j, ks_k\}$ in β is a measure the input/output connectivity for the pair $\{ks_j, ks_k\}$. Knowledge source pairs $\{ks_j, ks_k\}$ with a high connectivity imply that ks_k is highly dependent on ks_j for its input variables, conversely knowledge source pairs $\{ks_j, ks_k\}$ with a low connectivity imply that ks_k's inputs are independent of ks_j's outputs. A proposed heuristic to measure knowledge source connectivity is to compute a connectivity value, $\Pi_{j,k}$ for each pair $\{ks_j, ks_k\}$ in β.

$$\Pi_{j,k} = \frac{card\Theta_{j,k}}{min\ (card(\Phi_k), card(\Psi_k))} \qquad (6)$$

6

This heuristic was developed to compute the percentage of overlap between the sets Φ_j and Ψ_k. The cardinality of the set $\Theta_{j,k}$ divided by the minimum of the cardinalities of the sets Φ_j and Ψ_k computes a percentage of overlap between the set $\Theta_{j,k}$ and the smaller of the sets Φ_j and Ψ_k. As $\Pi_{j,k}$ approaches 1.0 the I/O connectivity between ks_j and ks_k increases (figure 1), and as $\Pi_{j,k}$ approaches 0.0 the I/O connectivity between ks_j and ks_k decreases (figure 2). For the limiting cases, if $\Phi_j \supset \Psi_k$ or $\Psi_k \supset \Phi_j$, $\Pi_{j,k} = 1.0$ and ks_j and ks_k have direct I/O connectivity. If $\Theta_{j,k} = \phi$, $\Pi_{j,k} = 0.0$ and the two knowledge sources have no I/O connectivity.

Discussion of Results

The results of the static input/output connectivity analysis can be used to measure the specialization and interdependence of the knowledge source in β. The static knowledge source I/O connectivity analysis will produce a measure of the interdependence between knowledge sources and a measure of the data transfers across the blackboard and through the network. From this analysis the designer can determine if the knowledge sources have been partitioned correctly and if the expected level of knowledge source independence has been achieved. This type of analysis has been done by hand while developing several blackboard systems.[3..6]

These preliminary heuristics were developed during the design and testing of the Cube_CLAWS blackboard system[4]. The initial design stressed knowledge source specialization and did not consider knowledge source independence. During system testing inefficiencies in the initial design of three knowledge sources were discovered. The following example will show the original knowledge source designs and the knowledge sources I/O connectivity and independence values. The system was then redesigned to achieve better knowledge source independence.

Initial system design:

$$\beta = \{ks_1, ks_2, ks_3\}$$

$$\Psi_1 = 7, \Phi_1 = 2$$
$$\Psi_2 = 9, \Phi_2 = 4$$
$$\Psi_3 = 4, \Phi_3 = 6$$

$$\Gamma_{1,2} = \phi \quad \Gamma_{2,1} = \phi \quad \Gamma_{3,1} = \phi$$
$$\Theta_{1,2} = \phi \quad \Theta_{2,1} = 4 \quad \Theta_{3,1} = \phi$$
$$\Gamma_{1,3} = \phi \quad \Gamma_{2,3} = \phi \quad \Gamma_{3,2} = \phi$$
$$\Theta_{1,3} = \phi \quad \Theta_{2,3} = 4 \quad \Theta_{3,2} = \phi$$

$$\Omega_{1,2} = 0.0 \quad \Omega_{2,1} = 0.0 \quad \Omega_{3,1} = 0.0$$
$$\Pi_{1,2} = 0.0 \quad \Pi_{2,1} = 1.0 \quad \Pi_{3,1} = 0.0$$
$$\Omega_{1,3} = 0.0 \quad \Omega_{2,3} = 0.0 \quad \Omega_{3,2} = 0.0$$
$$\Pi_{1,3} = 0.0 \quad \Pi_{2,3} = 1.0 \quad \Pi_{3,2} = 0.0$$

Analyzing the initial design shows that ks_1 and ks_2, and ks_2 and ks_3 both have I/O connectivity values of 1.0. This high degree of I/O connectivity caused very short processing pipelines to occur and forced serial execution of these knowledge sources. The small size and fast execution rates of the knowledge sources made pipeline processing a costly approach. The cost of message passing was greater than the speedup gained by parallel execution of multiple copies of the knowledge sources. The system was redesigned to decrease I/O connectivity. The redesigned system combined the second and third knowledge source into a single knowledge source.

For the redesigned system:

$$\beta = \{ks_1, ks_2\}$$

$$\Psi_1 = 7, \Phi_1 = 2$$
$$\Psi_2 = 10, \Phi_2 = 9$$

$$\Gamma_{1,2} = \phi \qquad \Gamma_{2,1} = \phi$$
$$\Theta_{1,2} = \phi \qquad \Theta_{2,1} = 4$$

$$\Omega_{1,2} = 0.0 \qquad \Omega_{2,1} = 0.0$$
$$\Pi_{1,2} = 0.0 \qquad \Pi_{2,1} = 0.57$$

The redesigned system provided better performance than the first system with little loss in knowledge source specialization. This research clearly shows that the design process must consider both knowledge source specialization and knowledge source connectivity and find a balance between the two that will provide optimal performance.

Concluding Remarks

A concurrent object-oriented blackboard system that is faster, more efficient, and more powerful than existing systems can be developed using the design, simulation, and analysis tools. The use of the design and analysis tools provided the highly specialized, highly independent knowledge sources required for concurrent blackboard systems to achieve their design goals. The knowledge source connectivity analysis highlights potential memory hotspots and system bottlenecks, and identifies knowledge sources where the message-passing costs and other knowledge source communication costs outweigh the gains made by concurrent knowledge source execution. The system can also be used to predict the amount of time each knowledge source should require to read/write the blackboard objects, and where potential memory access conflicts can occur between knowledge sources.

The blackboard system designer can use the results of the knowledge source connectivity analysis to revise the blackboard system design. The design can be modified to gain the desired knowledge source size and independence, and the desired knowledge source I/O connectivity. These blackboard system design and analysis tools are designed to function for generic blackboard systems and are application independent.

References

1.   Nii, H. Penny: "Blackboard Systems: The Blackboard Model of Problem–solving and the Evolution of Blackboard Architectures." *AI Magazine*, **7** (2), 1986, pp. 38 - 53.

2.   Corkill, Daniel D: "Advanced Architectures: Concurrency and Parallelism." In *Blackboard Architectures and Applications*, ed. V . Jagannathan et al., Academic Press, Inc., 1989.

3.   McManus, John W: "A Parallel Distributed System for Aircraft Tactical Decision Generation." In *Proceedings of the 9th Digital Avionics Systems Conference*, 1990, pp.505 - 512.

4.   Hueschen, Richard M. and John W. McManus: "Application of AI Methods to Aircraft Guidance and Control." In *Proceedings 1988 American Control Conference,* June 15-17, 1988, pp. 195 - 201.

5.   McManus, John W. and Kenneth H. Goodrich. "Application of Artificial Intelligence (AI) Programming Techniques to Tactical Guidance For Fighter Aircraft.*"* In *Proceedings 1989 AIAA Guidance, Navigation, and Control Conference*, 1989. AIAA Paper # 89-3525, pp. 851-858.

6.   McManus, John W. and Kenneth H. Goodrich: "Artificial Intelligence (AI) Based Tactical Guidance For Fighter Aircraft." In *Proceedings 1990 AIAA Guidance, Navigation, and Control Conference*, 1990. AIAA Paper # 90-3435.