Table of Contents

1 INTRODUCTION ........................................................................................................3
2 THE BASIC IDEA ......................................................................................................3
3 HEARSAY-II ............................................................................................................4
4 BB1 .........................................................................................................................5
5 BENEFITS OF BLACKBOARD ARCHITECTURES .............................................5
6 DRAWBACKS OF BLACKBOARD ARCHITECTURES ......................................6
7 REFERENCES ..........................................................................................................8
1 Introduction
A blackboard system is a task independent architecture for integrating multiple problem solving modules (referred to as knowledge sources). By task independent we mean that these architectures can be used for a wide range of tasks such as classification, design, diagnosis, repair etc.

Integrated problem solvers, such as systems based on blackboard systems, attempt to overcome the inherent limitations of a single heuristic expert system by employing two or more problem solving (or reasoning systems). The problems solvers may all use the same technology or they may reply on different technologies. For example, a system might integrate a heuristic rule based reasoning system with a case-based reasoning system and possibly a model based system.

However, integrated problem solving systems introduce a host of new issues. The overall problem facing such systems is that of integration of the problem solvers. Punch [1] categorises integration techniques as being fixed, programmable or task specific. However, this is possibly too simplistic a view of the integration problem. Integrated systems can possess different features in different areas, for example, one system might possess a flexible programmable structure, yet have a fixed cooperation strategy. Integration really refers to the problems of control, cooperation and communication of diverse problem solvers within a single system.

Blackboard systems (see [2]) are one (well established) approach to solving this problem of control, cooperation and communication between problem solvers.

2 The Basic Idea
In a blackboard system, a set of problem solving modules (typically called knowledge sources) share a common global database (called the blackboard). The contents of the blackboard denote and indeed are often called hypotheses. These hypothesis are often structured hierarchically. Knowledge sources respond to changes on the blackboard, and interrogate and subsequently directly modify the blackboard. This modification results form the creation, modification and solution of hypotheses. In fact only knowledge sources are allowed to make changes to the blackboard. It is therefore through the blackboard that the knowledge sources communicate and cooperate.

In a blackboard architecture, each knowledge source responds only to a certain class or classes of hypotheses. These hypotheses, that a knowledge source responds to, often reflect the different levels in the blackboard’s hierarchy. The blackboard holds the state of the problem solution, while the knowledge sources make modifications to the blackboard when appropriate.

As an example, consider the sample blackboard system taken from the Hearsay-II project (see Erman et al [3]).
The Hearsay-III architecture (see [4]) is essentially a generalisation of the Hearsay-II system. That is, it is a “bare” blackboard framework and contains no specialised knowledge.

3 Hearsay-II

In the Hearsay-II blackboard system, the flow of control occurs between the knowledge sources. These knowledge sources attempt to work in an opportunistic manner, influenced by the scheduler. By opportunistic we mean that cooperation occurs dynamically determined during the execution of the system based on the current information available.

In a blackboard system, each of the knowledge sources monitors the blackboard looking for an opportunity to aid in the emerging solution being recorded on it. If a knowledge source identifies a situation in which it can provide some information it registers its wish to be activated by posting a knowledge source activation record to the scheduler’s agenda.

The scheduler determines which of the knowledge sources wishing to be activated next is chosen for activation. It does this using information in the focus-of-control database and the likely impact of the knowledge source. The impact might be the degree to which it refines the hypothesis space. The blackboard monitor is the system component that updates the focus-of-control database. It does this by monitoring the generation and modification of hypotheses on the blackboard. Such a control regime can be described as flexible control.
The knowledge sources are constrained to cooperate only with the blackboard. Cooperation within a blackboard system is effected via the blackboard. Knowledge sources can create, modify and solve hypotheses on the blackboard, which may then receive further processing by other knowledge sources. This cooperation is described as opportunistic.

The blackboard is, in fact, a global database that is visible to all knowledge sources. The contents of the blackboard (the hypotheses) are often organised into a hierarchy, where each level represents a general abstraction of a lower level. The knowledge sources are organised into corresponding hierarchies and respond only to a particular class or classes of hypotheses reflected in the different levels in the hierarchy. Information can be transferred from one level in the hierarchy to another only by the knowledge sources.

4 BB1

BB1, described by Hayes-Roth in [5], extends the basic blackboard architecture of Hearsay-II, by adding a blackboard control architecture. A system built using the BB1 framework has two blackboards; a domain blackboard and a control blackboard. The domain blackboard is the same as the blackboard described above, the control blackboard however is concerned with generating a solution to the control problem.

The control blackboard in BB1 works in exactly the same way as a standard blackboard, however the knowledge sources are now control problem solvers and the solution on the blackboard are solutions to the problem of which domain knowledge source should be activated next.

In essence BB1 dynamically generates a solution to the problem of control. It does this using knowledge about the current state of the problem, and knowledge about control problem solving (which is implicit in Hearsay-II within the design of the scheduler).

The end result is that BB1 possesses a more flexible and powerful control strategy than that provided by Hearsay-II (at the expense of additional complexity and processing).

5 Benefits of blackboard architectures

Some of the main benefits of blackboard architectures and the systems built using a blackboard framework are as follows:

**Flexibility of configuration.** The knowledge sources within a blackboard system are not tied together in a fixed or rigidly applied manner. Instead they communicate and cooperate via the common blackboard. This means that any knowledge sources can be added to the system without having to specify its existence in any other knowledge source.

**Flexible problem solving.** A blackboard architecture is independent of any particular task and can therefore be used as the basis of very different applications, from route planning, to image processing, speech recognition and diagnosis. Cooperation is driven by the current state of the problem, rather than through
any particular control strategy, and control of this cooperation is distributed between the knowledge sources themselves. These knowledge sources then determine which they are appropriate and the scheduler determines which of the appropriate knowledge sources to actually activate.

**Selection of knowledge sources.** Because more than one knowledge source may be able to perform the same function, the scheduler can select that problem solver which will provide the most benefit to the emerging solution. This can improve both problem solving efficiency and the quality of the eventual solution.

**Multiple problem solvers.** The blackboard architecture promotes the integration and multiple problem solving modules (knowledge sources). Each of these knowledge sources can potentially have its own representation and inference mechanism. This means that the twin advantages of multiple reasoning systems are maintained, namely: resilient behaviour and efficient behaviour.

**Management of multiple levels of abstraction.** The blackboard architecture explicitly supports the management of multiple levels of abstraction. Not only can the blackboard be divided hierarchically, but the knowledge sources can reflect this hierarchy. Problems of moving between the levels of the hierarchy can be overcome by providing knowledge sources whose sole task is to move information between two particular abstraction levels.

**Opportunistic cooperation.** Cooperation in a blackboard system is explicitly opportunistic; knowledge sources can post partial solutions to the blackboard in the hoped that some other knowledge source will be able to take these partial solutions and progress further down the path to the final solution.

### 6 Drawbacks of blackboard architectures

Some of the main drawbacks of blackboard architectures for integrating multiple problem solvers in a system such as a classification expert system include:

**No communications language.** The blackboard acts as a very general communications facility. However, the communications language is the language of the hypotheses. That is, if all communications have to go via the blackboard then not only does the blackboard access mediate each decision step, it also constrains the representation of the information to be exchanged. That is, the language for passing information between problem solvers is the language of the intermediate hypotheses on the blackboard. This means that rather than an explicit request for some task to be performed, such as the generation of a simulation, the request must be phrased in the terms used to describe partial solutions on the blackboard — this may or may not be appropriate.

**Computational complexity of cooperation.** The problem of determining which knowledge sources are appropriate in each situation and which of these appropriate knowledge sources should be activated has to be solved each time a decision has to be made. In the case of BB1 this can be
computationally very expensive, even in Hearsay-II system this imposes a certain amount of computational overhead.

**No guidance for task.** Blackboard architectures do not give any guidance to a system builder on how to solve a particular problem or perform a particular task within their framework. That is, blackboard architectures do not give any support for determining how specific problems should be addressed. Indeed they were never intended to do so, they are task independent architectures.

**More complex problem solving framework.** The full facilities of a general problem solving system are rarely required for a specific task. Yet blackboard architectures provide such a generality whatever the task being performed. Therefore, systems built within the blackboard framework often possess a more complex problem solving framework than is necessary.

**No exploitation of task specific strategies.** There is no explicit support for task specific and domain specific problem solving strategies (for example, the use of useful diagnostic short cuts). Such strategies would have to be implicitly added into the scheduling system of the architecture.

**Unbound nature of the control problem.** BB1 faces the potential unbound problem of determining which problem solver to invoke. A great deal of control “problem solving” could occur before any real work is begun on solving the problem on the domain blackboard.

**Global database.** The blackboard, through which all knowledge sources communicate, by which cooperation is achieved and on which the emerging solution is stored, is actually a global database. It is directly modified by the knowledge sources and assumes that knowledge sources will behave responsibly. If they do not, chaos could ensue. This assumption goes against all the principles of information hiding which have been developed by software engineers to promote well engineered systems.

**Knowledge of problem solving scope.** Although individual knowledge sources are aware of the types of situations (problems) they can address, no knowledge of the scope of the system as a whole is maintained. Therefore blackboard systems are unable to easily detect problems on or beyond the periphery of the whole system’s ability.

**Knowledge acquisition.** Although the use of specialist knowledge sources should make it easier to develop problem solving systems, and to encode the appropriate knowledge in them, no support is given to help in the process of obtaining this knowledge in the first place. In addition, each knowledge source can potentially possess its own knowledge representation that may require its own particular techniques not only to obtain but also to codify.

**Implicit representation of management information.** Within the scheduler of the Hearsay-II system, problem management information is implicitly represented. This information focuses on the generation of a solution, and determines which knowledge source, from those capable of processing a
particular solution, should be invoked next. In order to be able to do this efficiently, knowledge about problem solving strategies, short cuts, the abilities of the knowledge sources and task specific techniques are required. The implicit nature of the scheduler hides much of this information and some of these functions.

**Problem solvers defined at system build time.** The knowledge sources need to be “connected” to the blackboard system at system build time. The abilities of the knowledge sources also need to be made available to the scheduler prior to the problem solving, in order that it can determine the “impact” of knowledge source.

7 References


