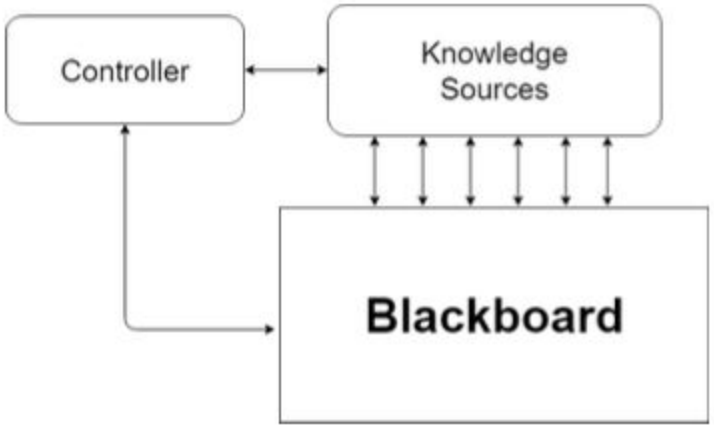
Blackboard architecture

Definition

- Blackboard is an architecture for problem-solving, especially for a complex problem which a single expert is not able to handle — for example, multilingual translation.
- In general, problem-solving in blackboard architecture begins with the announcement of a problem and writing initial data onto the blackboard. The specialists are watching the blackboard looking for an opportunity in order to make a contribution to solution development. When a specialist finds this opportunity, they record the contribution on the blackboard, in the hope that others will use his contribution for final problem-solving. This process continues until the problem is solved.
- The procedure to a solution in the blackboard architecture is assumed to be unknown in advance.

- Vocabulary for components and connectors:
    - Knowledge sources: Knowledge sources (KSs) are "independent modules that contain the knowledge needed for problem-solving" . Each language specialist in the above example of multilingual translation is a KS. KSs need to focus on the progress of solving the current problem - track the sentences to be translated. However, it is not necessary for a KS to communicate with or even know the existence of other KSs.
    - Blackboard: Blackboard is the place where language specialists work together to develop the translation. The formal definition of the blackboard is "a globally accessible database which is used for intermediate, partial results of problem-solving ". Each KS can read and write the shared blackboard.
    - Controller: The controller decides "which candidate knowledge source to execute next in runtime for optimal problem solution" [DA]. In the multilingual translation problem, if there are multiple experts in a language available for a piece of text, Controller is the coordinator to choose who should go for the translation.

Topological Constraints



There is no direct communication between KSs. Instead, all communications happen indirectly by reading and writing information to the global blackboard. The controller is in the duty of for "selecting among the candidate KSs to assign the task to one of them" .

Applicable problems

- Complex problems where information is ambiguous
- A large problem consisting of smaller sub-problems
- Non-deterministic problems
- A problem with many different solutions

Resilience to change

- Highly resilient to change
    - When new knowledge sources are added, the other components of the system are not affected
    - If new information is added to the problem, new knowledge sources can easily be added
    - There are no dependencies between knowledge sources, so they can easily be put in, taken out, or swapped.

Negative behaviours

- Complex controller:
    - The controller needs to schedule and coordinate the KSs. It may not be straightforward to implement the controller if there are many different KSs.
- The difficulty of testing:
    - Because of the non-deterministic nature of the blackboard architecture, the solution of each execution may not be consistent.
- Efficient solution not guaranteed:
    - The blackboard architecture might only partially solves a problem due to its non-deterministic nature. The controller might trigger inefficient or unnecessary KSs to solve a specific problem.
- Single point of failure:
    - The blackboard is a single point of failure. If the blackboard crashes, the whole system will stop working because none of KSs are able to access the memory to solve the problem.

Supported NFPs

- Reusability :
    - Since KS modules only have interaction with the blackboard module, and each KS module solves problems independently, a KS can be easily reused in a different Blackboard system.
- Scalability :
    - Due to the nature of low coupling design, KS modules can be easily added or removed to handle the changes in workload.
- Robustness :
    - The Blackboard system is robust because a failure in one KS module does not affect other KS modules, and other KSs could continue to process the problem to find a potential solution.
- Heterogeneity :
    - Due to the nature of low coupling design, different subjects of KS module can be integrated into a Blackboard system to solve comprehensive problems.

Inhibited NFPs

- Dependability :
    - KS modules depend entirely on the blackboard module to communicate because of the low coupling design that there is no direct communication

between KS modules. This makes the blackboard module a single point of failure. A malfunction in the blackboard module would affect the whole system's effectiveness.

- Testability :
    - Since there is randomness on the order of KS is applied to solve problems, the possible combinations of KSs are often too large to be fully tested in real life.

- Reliability :
    - The Blackboard system determines a solution at run-time. Therefore, the output may be slightly different and unpredictable for the exact same problem.

Comparison with other architectures

- In Mobile code, plug in and black board architectures, the components are not coupled.
- In Mobile code, plug in and black board architectures, the data is in one place.
- The plugin architecture aims to do different tasks, the blackboard architecture aims to accomplish one single task.

Credit:

Students:

A Group of Students (Daniel Francisco, Ellen Gao, Raiyan Alim, Charlie Yu)

Winterberry (Heli Wang, Yang Chen, Danying Gu, Bincan Wang)

Instructor and TAs:

Mei Nagappan, Achyudh Ram Keshav Ram, Aswin Vayiravan, Wenhan Zhu