Software Design and Architecture

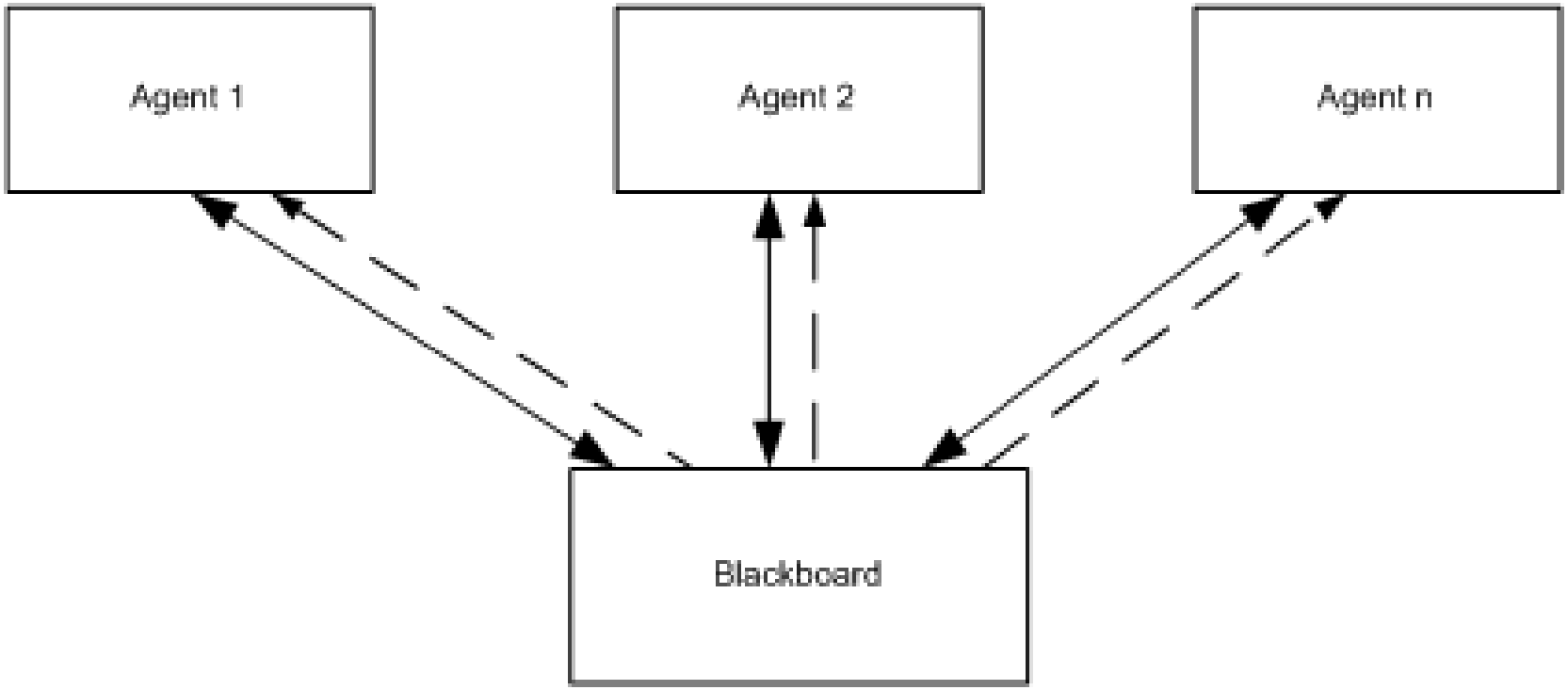
Blackboard Architecture

Blackboard Architecture

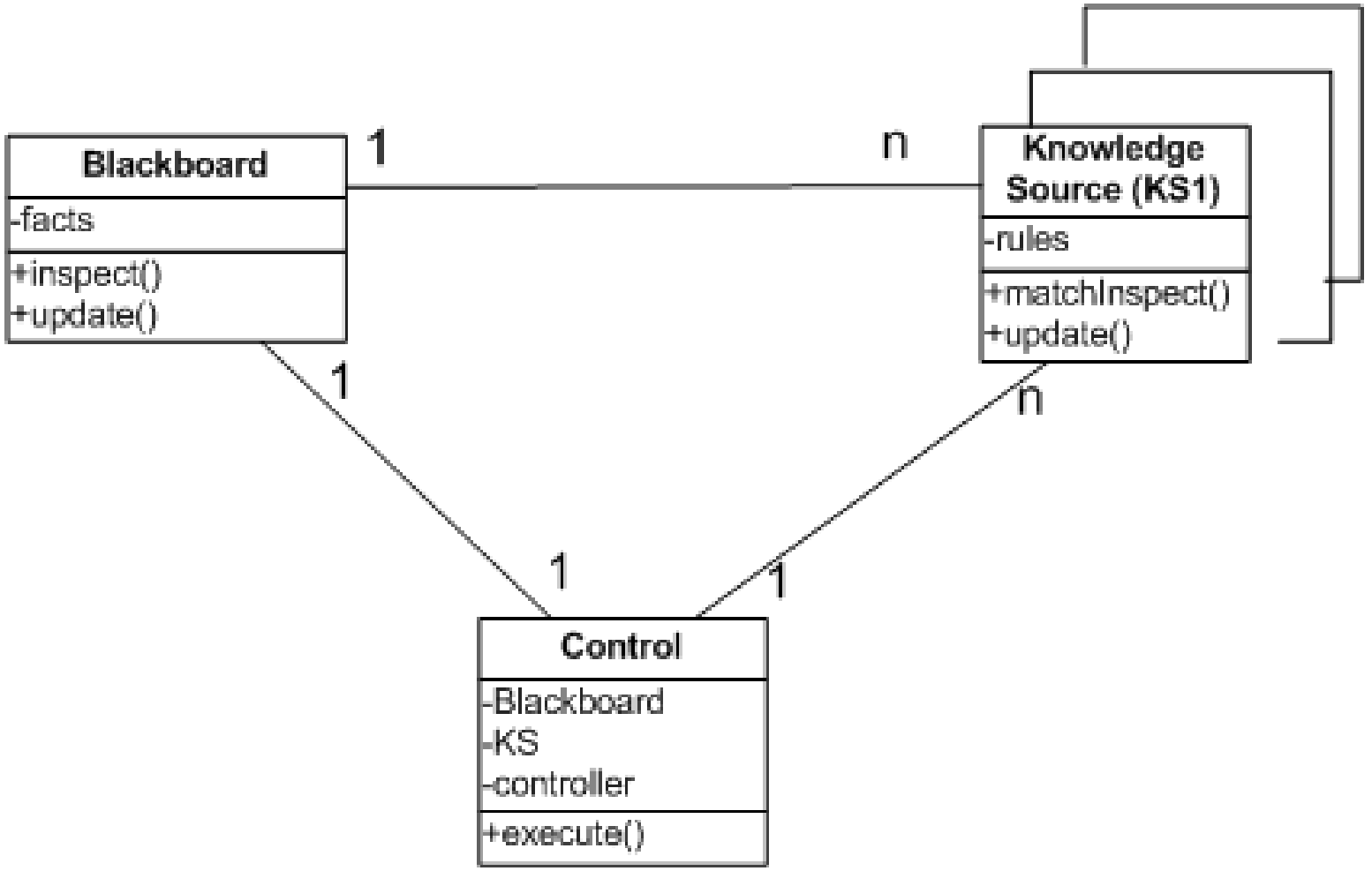
- The word blackboard comes from classroom teaching and learning.
- Teacher and students can share data in solving a problem on the classroom via a blackboard.
- Students and Teacher play the role of agents to make their contributions to the problem solving.
- They all can work in parallel and very independently.
- They will try their best to work together to find the best solution.

- The idea of blackboard architecture is similar to the classroom blackboard setting used in solving problems without deterministic outcome.
- It is a data-directed and a partially data-driven architecture.
- The whole system is decomposed into two major partitions.
- One partition, called the *blackboard*, is used to store data (hypotheses and facts), while the other partition, called *knowledge sources*, is where the domain-specific knowledge is stored.
- There may be a third partition called the *controller* that is used to initiate the blackboard and knowledge sources, takes a bootstrap role and overall supervision control.

- The connections between the blackboard sub-system and knowledge sources are basically implicit invocations from the blackboard to specific knowledge sources, which are registered with the blackboard in advance.
- Data changes in the blackboard trigger one or more matched knowledge source to continue processing.
- Data changes may be caused by new deduced information or hypotheses results by some knowledge sources.
- This connection can be implemented in *publish/subscribe* mode.


Agent 1
Agent 2
Agent n
Blackboard

- Many domain specific knowledge sources collaborate together to solve a complex problem such as pattern recognition or authentication in information security.
- Each knowledge source is relatively independent from other knowledge sources.
- They don't need to interact with each other, which is very similar to a repository system.
- They only need to interact and respond to the blackboard sub-system.
- Each source only works on a specific aspect of the problem, and contributes a partial solution to the ultimate solution.


Blackboard
-facts
+inspect()
+update()
1
n
Knowledge Source (KS1)
-rules
+matchInspect()
+update()
1
n
1
1
Control
-Blackboard
-KS
-controller
+execute()

- Let's take a look at a well-known animal identification knowledge based system (KBS).
- The knowledge is represented in the format of production rules with condition and action parts.
- For each rule, if the condition is true then the action is taken.
- The action is to put new conclusion data in the data store, which is the blackboard.

Here is a set of rules:

R1: IF animal gives milk THEN animal is mammal

R2: IF animal eats meat THEN animal is carnivore

R3: IF animal is mammal AND animal is carnivore AND animal has tawny color AND animal has black stripes THEN animal is tiger

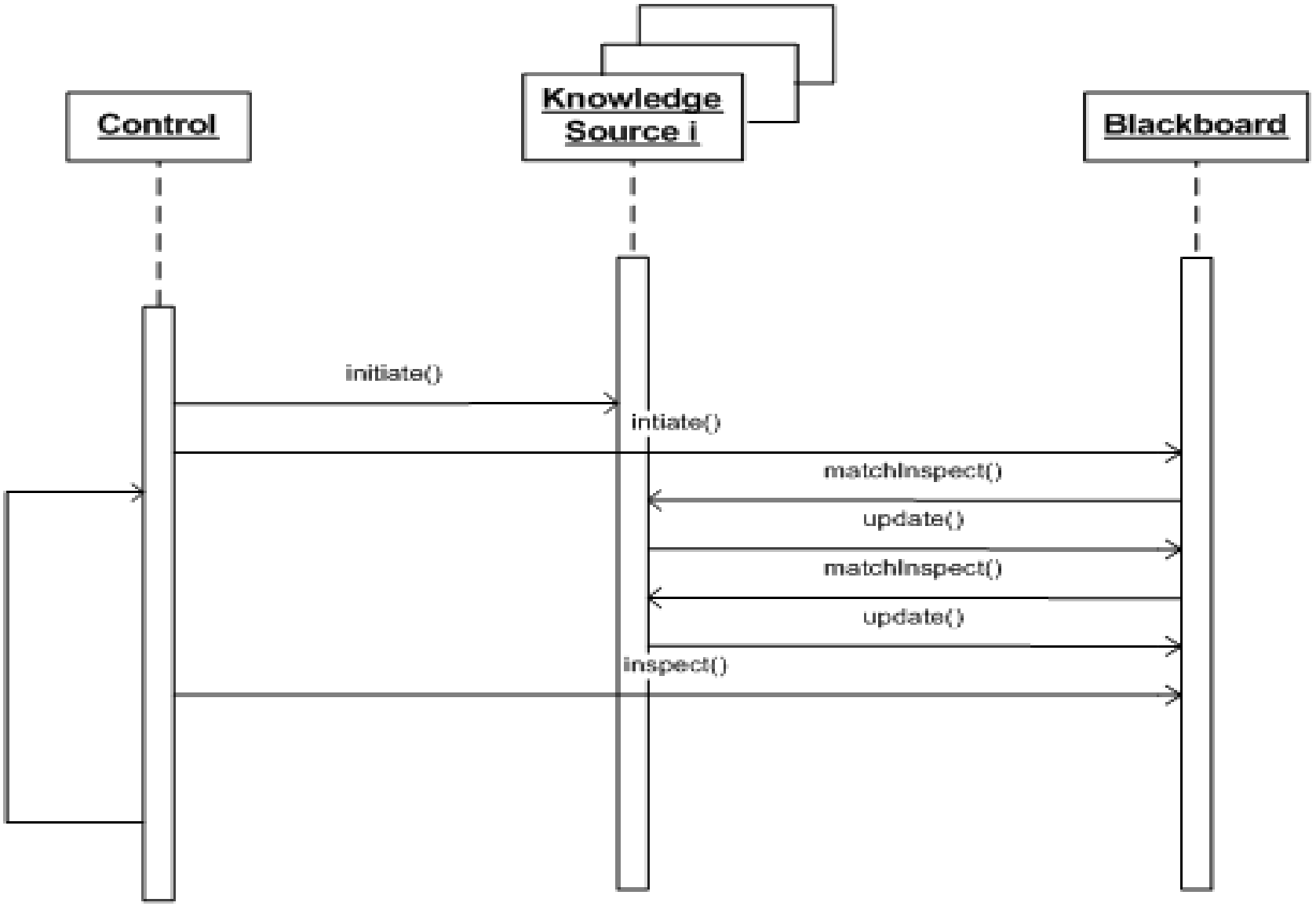
Forward Reasoning or Backward Reasoning.

- The forward reasoning starts with the initial state of data and proceeds towards a goal.
- It may reach the given goal or it may fail to reach the goal.
- If the goal is not given, the reasoning will come to a point where no more new facts can be derived that indicated that is the best result we can get.
- The backward reasoning works in the opposite direction than the forward reasoning.

- Let's follow the forward reasoning sequence at this time. F1 matches the condition of R2 and "animal is carnivore" is derived and stored in the blackboard.

- If the algorithm checks the newest generated data first, the new fact does not match any rules, and then checks F2 which matches the condition of R1, and "animal is mammal" is derived and stored in the blackboard.

- After this point there is no single fact that matches any rule and if combined data are used the R3 is matched, and "animal is tiger" is derived and put back in the data store.


Control
Knowledge Source i
Blackboard
initiate()
intiate()
matchInspect()
update()
matchInspect()
update()
inspect()

Applicable Domain of Blackboard Architecture:

- Suitable for solving open-ended and complex problems such that artificial intelligence (AI) problems where no deterministic solutions exist.
- The problem spans multiple disciplines, each of them has complete different knowledge expertise and problem solving paradigms that co-operation is a must.

- Partial, or approximate solution is acceptable to the problems.
- Exhaustive searching is impossible and impractical since it may take forever because available knowledge and even data and hypotheses may not complete or precisely accurate.

Benefits

- Scalability: easy to add new knowledge source or update existing knowledge source.
- Concurrency: all knowledge sources can work in parallel since they are very independent of each other.
- Supports experimentation for hypotheses.
- Reusability of knowledge source agents.

Limitations

- Tight dependency between the blackboard and knowledge source, the structure change of the blackboard may have a significant impact on all its agents.
- Difficult to make a decision when to terminate reasoning, since only partial or approximated solutions are expected
- Synchronization of multiple agents is an issue. Since multiple agents are working and updating the shared data in the blackboard simultaneously, the preference or priority of executions of multiple agents must be coordinated.
- Debugging and testing of the system is a challenge.

Summary

- Introduction to Blackboard Architecture.
- Benefits of blackboard architecture.
- Limitations of blackboard architecture.