

INSTITUTO FEDERAL
CEARÁ



UNIVERSIDADE
ESTADUAL DO CEARÁ

MInD: don't use agents as objects

Renato Lenz Costalima¹, Amauri Holanda de Souza Junior¹,
Cidcley Teixeira de Souza¹ and Gustavo Augusto Lima de
Campos²

¹Federal Institute of Ceará, Brazil ²State University of Ceará, Brazil

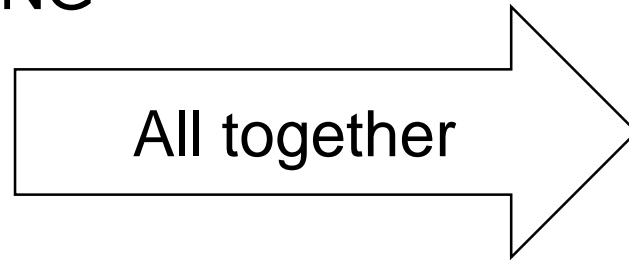
renato.lenz@ifce.edu.br

WHAT IS INTELLIGENCE?

- It is not possible to see the internal details of intelligence.
- It is described by its behaviors, including:
 - PROBLEM SOLVING
 - LEARNING
 - LANGUAGE
 - EMOTIONS

WHAT IS INTELLIGENCE?

- It is not possible to see the internal details of intelligence.
- It is described by its behaviors, including:
 - PROBLEM SOLVING
 - LEARNING
 - LANGUAGE
 - EMOTIONS



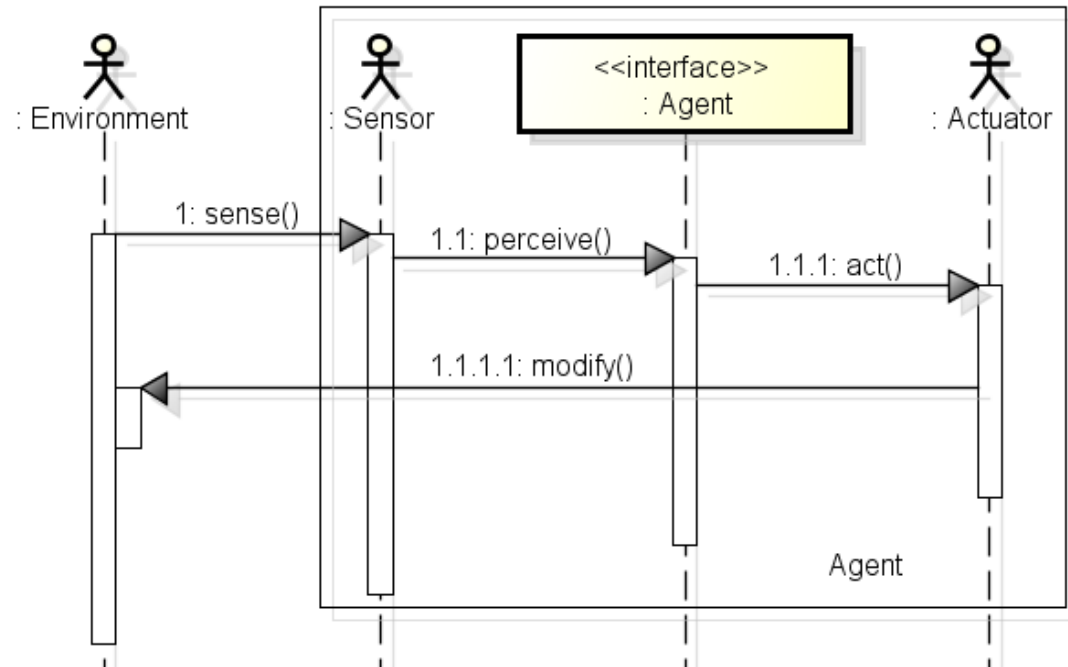
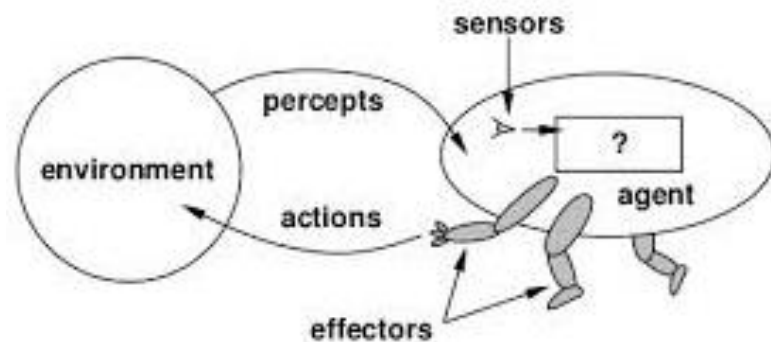
AGI?

WHAT IS INTELLIGENCE?

- These behaviors are expected outputs of intelligence, but they ARE NOT THE INTELLIGENCE ITSELF.
- Intelligence is rather what makes them possible. That could be: **“the capacity to acquire and apply knowledge”**.

Agents

- “An agent is anything that can be viewed as *perceiving* its environment through sensors and *acting* upon that environment through effectors” (Russell *et al*, 1995)



Agents

- The `perceive()` method is the only method in the `Agent` interface. Surprisingly, none of the major multi-agent framework defines it.
- All the interactions with the agents should happen through the `perceive()` method, including attempts of communication.

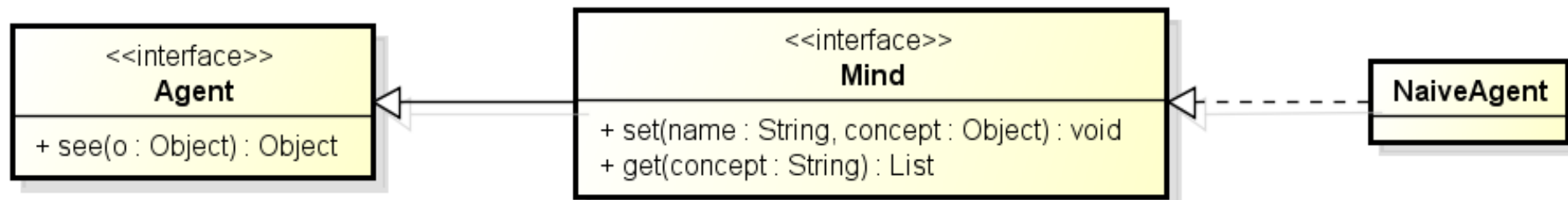
Intelligent agents

Intelligent agents must be able to “acquire and apply knowledge”.

- The agent needs a body to gather and represent the information
- The Mind interface must be able to store (`set()`) and retrieve (`get()`) it

MInD – MODEL FOR INTELLIGENCE DEVELOPMENT

- The multi-agent framework MInD, **Model for Intelligence Development**, provides the class `NaiveAgent`, a Java implementation of an all-purpose agent ready to run.
- Early development
- Beginning tests with communicating agents



DO NOT USE AGENTS AS OBJECTS

```
1  Mind agent = new NaiveAgent();
2
3  agent.see(new Symbol("write", "hi"));
4  agent.set("write", new AbstractAction() {
5      public Object act(Object object) {
6          System.out.println(object);
7          return null;
8      }
9  });
10 agent.see(new Symbol("write", "hi"));
```

DO NOT USE AGENTS AS OBJECTS

- Agents should not invoke each others methods in agent-oriented world (Wooldridge, 2009).
- A method invocation is like pushing a button: the corresponding behavior is triggered without the object's decision.
- Agents should communicate “asking” for a desired operation
- Multi-agent frameworks focus on the communication between the agents, defining specific communication methods and protocols.
 - The agent is not forced to execute the desired action, but it is forced to communicate.
 - The communication language is “hard-coded”.

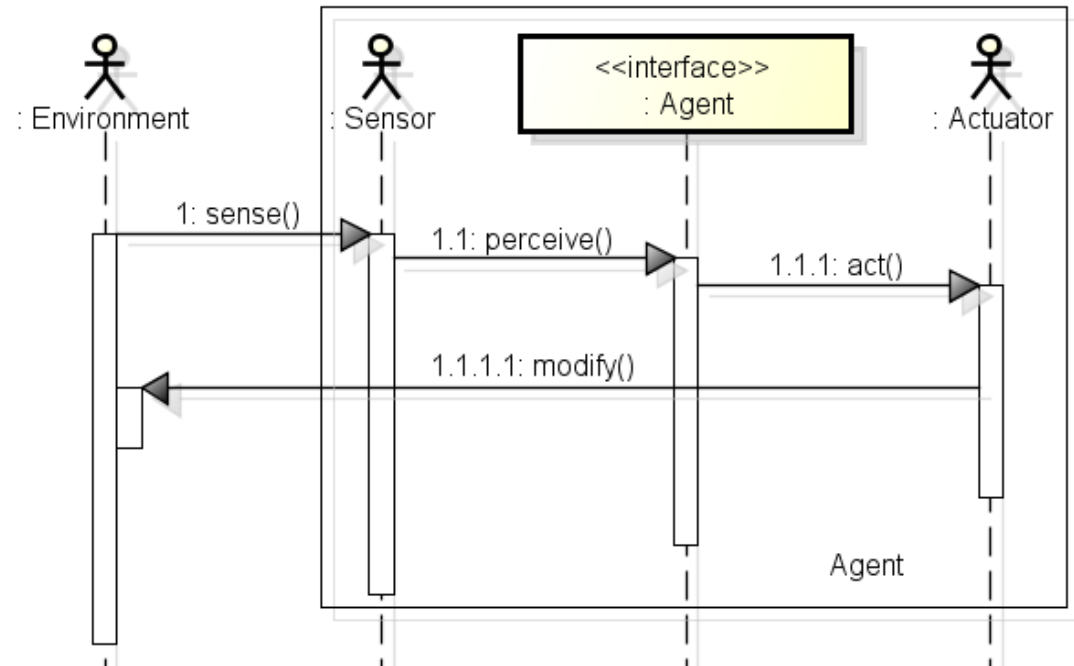
DO NOT USE AGENTS AS OBJECTS

- Can a teacher “set” knowledge inside the student's head?
- Can he get the student's attention, to make him “perceive”?
- Sensors and actuators (the agent’s body) are objects that we can see and manipulate.
- The mind is supposedly in control of the body.
- Yet, has anyone seen a mind?

DO NOT USE AGENTS AS OBJECTS

In a realistic agent-oriented world, the environment should not have access to the agent at all.

Only the agent's body (its sensors and actuators) should use the `Agent` and `Mind` interfaces (Figure 2).



LIVE EDITING

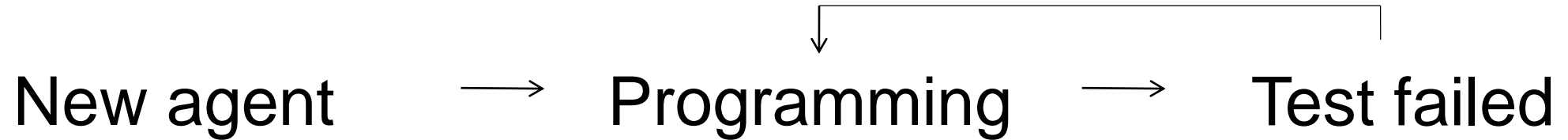
- Specially, do not build agents as objects.
- Other multi-agent framework



- Is that how it works with intelligent beings? If your kid fails a math exam do you kill him and try to make a smarter kid?

LIVE EDITING

- MInD



- Avoid the hideous cycle of “edit-refresh-save”.
- In analogy with Web 2.0, MInD allows a single method to be updated without the need to restart the entire software.