# Implementation

For implementation we have use the JADE framework with which we have built a multi-agent system: Context Model Administering Agent (CMMA), Context Interpreting Agent (CIA), GUI Agent and Reinforcement Learning Agent (RLA). The RLA parses a world file description containing the xml description of the available sensors and ads that information to the context ontology representation, while the uses a pooling mechanism to gather sensor data at specific time intervals.

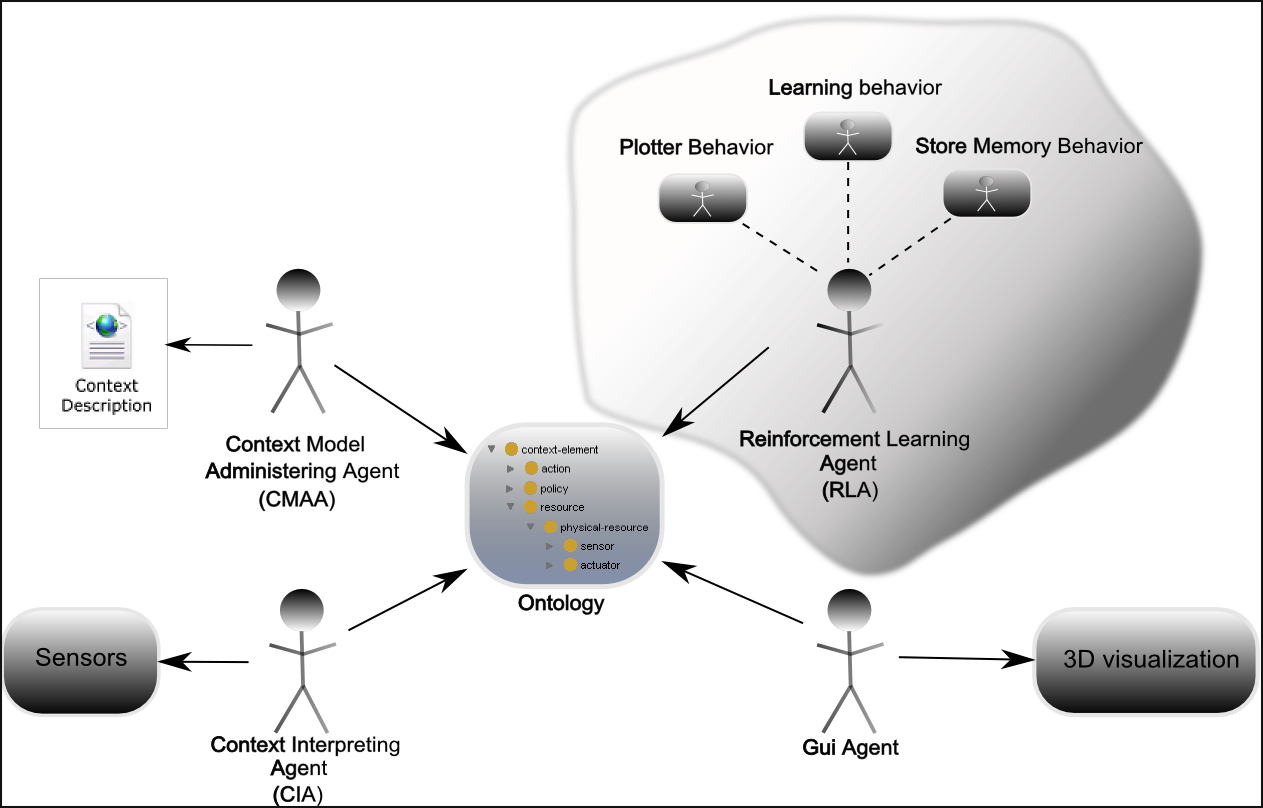


Figure 1 : Action Selection Framework Architecture

The actual action selection algorithm is implemented in the Learning Behavior attached to the RLA. The other two behaviors perform additional tasks like running time plot (Plotter Behavior) and file storage of what has been learned so far (Store Memory Behavior).

For test scenario we have chosen an environment having a computer, a camera with face recognition, a light source and an alarm. Each environment component has attached a sensor for monitoring its state and also one humidity and one temperature sensors have been added. There are three policies which our algorithm has to enforce: *light policy,* *face* *recognition policy* and *temperature and humidity* policy. The light policy specifies that the light should be on only if the room is not empty. The face recognition one specifies that if a professor is in the room the computer state must be on and if there is someone unknown in the room the alarm should go off. Last, the temperature and humidity policy enforces that the temperature should be between 18 and 23 degrees and the humidity between 20 and 30 %.

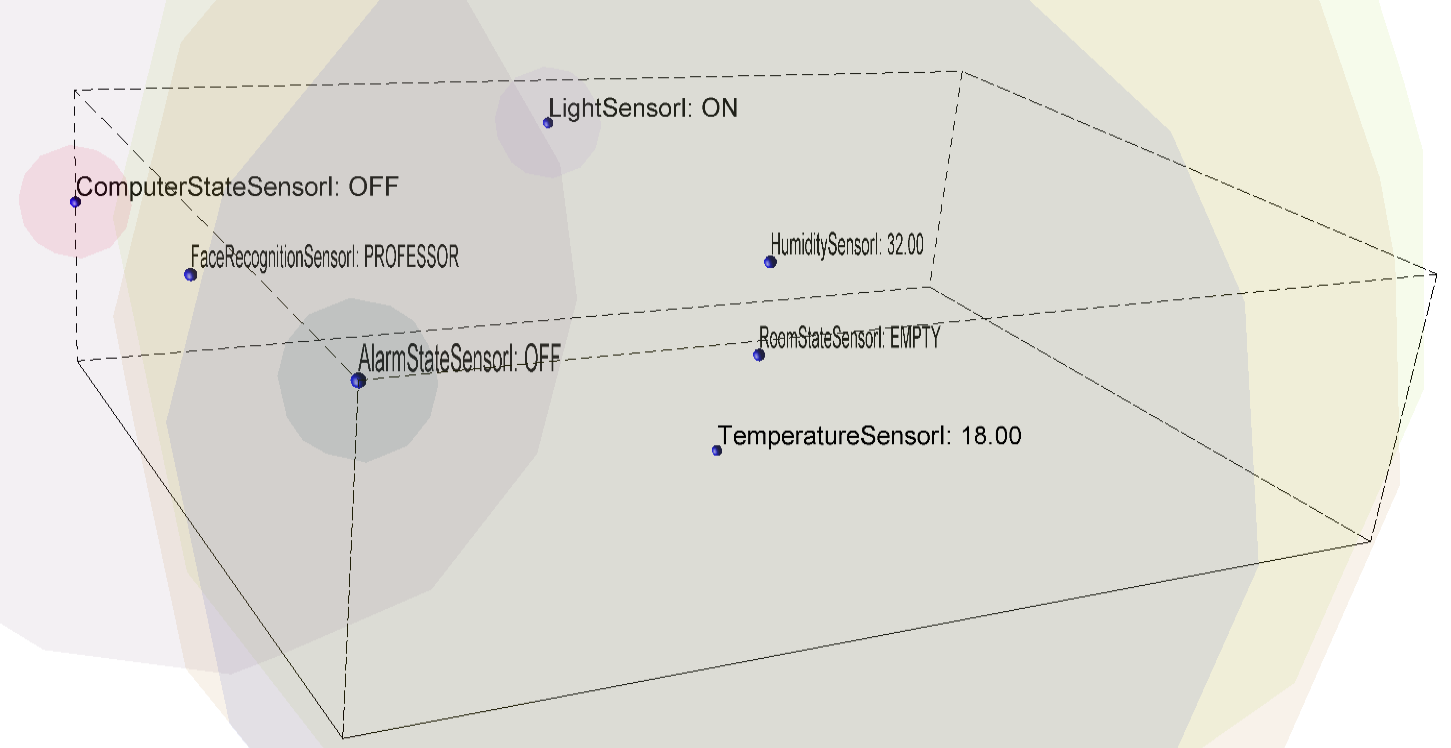
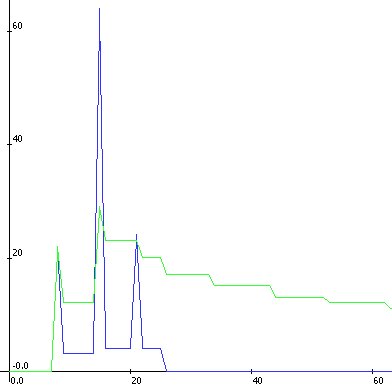


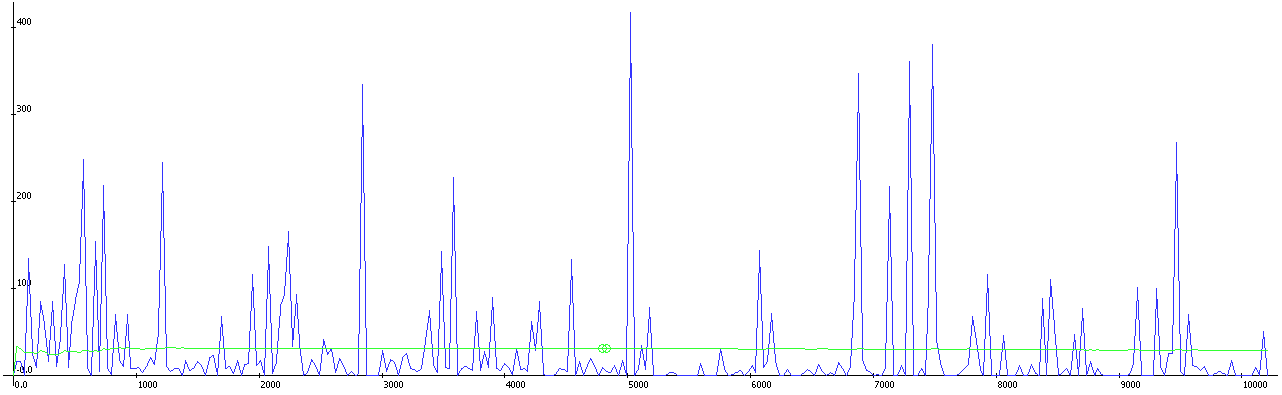
Figure 2: Test context



**Figure 3: Context disturbing agent working on a pattern**

Firstly, a pattern of four broken contexts is taken: the professor is in the room while the computer is off and light is off, the student is in the room while the alarm is off, the temperature and humidity are out of the admissible range, and an unknown person is in the room and alarm is off. We suppose we have at most a person in the room at a time.

In the above graph we can see that the first four times that the reinforcement algorithm runs, we have running times of 20, 4, 61 and 22 seconds. When the disturbing pattern is repeated, we already have in the registered in the memory the broken context with its corresponding actions, and therefore the reinforcement learning will take 0 seconds, and the self-healing mechanism will apply the actions that it already learned.

  **Figure 4: Random Disturbing of the Context**

For checking the behavior of the self-healing mechanism, we left it running 28 hours, with random values for all sensors, values set by Context Disturbing Behavior of our Agent. In the first 1000 seconds, almost all the running times of the reinforcement learning algorithm are larger than 10 seconds. After that, the self-healing mechanism begins to learn, achieving the performance of having only three running times greater than 50 seconds in the time interval [5000, 7000]. At each step of the reinforcement learning algorithm, it checks if it doesn’t already know the best sequence of actions for the context that it arrived in, and adds it to what it has discovered so far. Considering that the number of possible sensor combinations for random values of sensors is 22481940, the self-healing mechanism behaves quite well in rapidly finding and taking the needed actions for fixing the broken context.