Behaviour-based control in ARGoS: The Subsumption Architecture

– Intelligent Robotic Systems –

Andrea Roli
andrea.roli@unibo.it
Dept. of Computer Science and Engineering (DISI)
Alma Mater Studiorum Università di Bologna

The subsumption architecture

dato che abbiamo il superloop i livelli non p

The *subsumption architecture* is one the most used models for implementing behaviour-based control. In this lab session you will be asked to program a robot for a given task according to this architecture.

Task

The robot is expected to be able to find a light source and go towards it, while avoiding collisions with other objects, such as walls, boxes and other robots. The robot should reach the target as fast as possible and, once reached it, it should stay close to it (either standing or moving). For physical constraints the wheel velocity cannot exceed the value 15 (i.e., 15^{-2} m/s). The robot (a *footbot*) is equipped with both light and proximity sensors.

Exercise

Implement the robot control program in ARGoS on the basis of the subsumption architecture. Before starting to code, define the basic task-achieving behaviours (i.e. *competencies* according to Brook's terminology) and detail their relations. Design your code such that a possible further level can be added with a limited amount of changes in the previous code. Compare the behaviour of the robot in a condition without noise and another with noise (set noise level in both actuator and sensors; you may want to set the noise level at about 0.05).

Some suggestions:

- Consider the fact that you have an abstract machine in which only sequential processes can be run; therefore, some variants with respect to the subsumption architecture are in fact unavoidable.
- Try to think of the possible ways to implement a competence and do not stop at the first idea that comes to your mind. You will be surprised to discover that there are several possibilities, even for this simple task.
- Test your robot in different environments; for example, an arena with boxes placed randomly (vary number and size of boxes), another arena with one or more walls between the robot and the light—the walls may also have thin doors. Try also to set the light at different intensities and heights—this latter parameters impacts the visibility of the target.
- Try the controller also with more than one robot in the arena (just place n robots randomly by using `distribute`). Is the control software still achieving the goal?
- As you know the position of the light bulb, you can compute the distance between the robot and the target by means of the positioning sensor. Therefore, you can evaluate the performance of the robot by estimating the time needed to reach the light, e.g. in terms of the number of steps. To this aim, you may want to set a minimum radius around the target. Alternatively, you can either count how many times the robot can reach the light in a given time (just set the simulation duration in the ARGoS configuration file) or take the distance between the robot and the light at the end of the simulation.