

Behaviour-based control in ARGoS: Subsumption architecture

Andrea Roli

andrea.roli@unibo.it

Dept. of Computer Science and Engineering (DISI)

Alma Mater Studiorum Università di Bologna

A.Y. 2018/2019

The subsumption architecture

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 it.

Task

The robot must 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 can not exceed the value 10 (i.e., 10^{-2} m/s). The robot (a *footbot*) is equipped with light and proximity sensors.

Exercise: Subsumption architecture

Implement the robot control program in ARGoS on the basis of the subsumption architecture. Before starting to code, define the basic task-achieving behaviours (competencies) and design 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 both without and with noise.

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 possible, and actually 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 they are many, even for this simple task.
- Test your robot on different environments; for example, an arena with boxes placed randomly (vary the number 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 parameter impacts the visibility of the target.
- Try the controller also with more than one robot in the arena (just place a n robots randomly by using `distribute`). Is the control software still achieving the goal?