Department of Electronics, Information, and Bioengeneering
Politecnico di Milano

Artificial Intelligence 2019-20

Marco Colombetti

2. Agents

Chapter 2 of the textbook is an introduction to the concept of *rational agents*. It is useful to classify agents according to the complexity of their controller, understood as the component that takes high-level decisions on what to do. A first distinction is between *reactive* and *proactive* agents. A purely reactive agent is an agent that decides what to do only on the basis of the state of the perceived state of its environment; in other words, such an agents simply reacts to elements of the current external (or internal) situation. A proactive agent, on the contrary, is also sensitive to the future, at least as far as this can be anticipated by the agent. An example of reactive behaviour may be running away in front of a perceived danger; a proactive behaviour may be taking a route that one believes will avoid possible dangers. Purely reactive agents are often further classified in the two categories of *simple reflex agent* and *model-based reflex agents*, and proactive agents in the two categories of *goal-based agents* and *utility-based agents* (see the **textbook, Chapter 2**).

2.1 Agent diagrams

Figures 2.1, 2.9. 2.11, 2.13 and 2.14 of the textbook (from p. 35) present diagrams of the different types of agents (see above). I think some of these figures contain misleading elements. For example, Figure 2.1 of the textbook presents a general scheme of an agent. Here are some criticisms and comments:

– Sensors do not receive "percepts" from the environment. A sensor transforms certain *physical events* (which consist of the sensor's interaction with the environment) into *sensor data* about the environment.[1] For example, the image sensor of a digital camera transforms the sensor's interaction with incoming photons into binary numbers. Often, sensor data require some kind of interpretation before they can be exploited to produce action: the process of interpreting sensor data is known as *perception*, and the term "percept" is used to refer to the representation that is produced by such a process. For example, a percept may represent part of the environment as a sleeping cat, or as half-full bottle of red wine sitting on a table. So, percepts are neither physical interactions between sensors and the environment nor data produced by the sensors, but rather representations of the environment based on the interpretation of sensor data.
– Symmetrically, actuators transform *control data* into certain physical events, which we regard as the agent's actions (and consist in the actuator's interaction with the environment).
– The box with the question mark may be viewed as the agent's *controller*: it will have sensor data in input and will produce control data in output. Sensors and actuators are part of the agent, but they do not belong to the controller: they constitute the interfaces between the controller and the environment.

Figure 2.1 of these notes presents a revision of the general scheme. Often a part of the environment, which we may call the *internal environment*, is internal to the agent. For example, a robot's battery is obviously part of the robot; but it is also part of the environment, in that it can be monitored by the robot's controller through suitable sensors, so that when the charge level goes below a certain threshold appropriate action can be taken.

[1] A device that transforms physical events into data (or vice-versa) is often called a *transducer*. Sensors and actuators are therefore input transducers and output transducers, respectively.

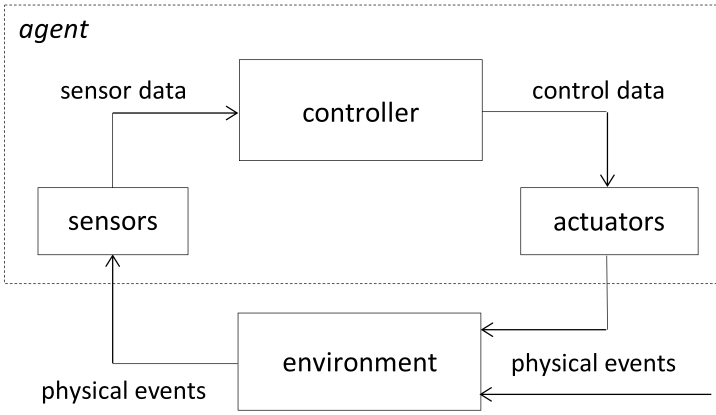

Figure 2.1. Revised general scheme of an agent.

In certain cases (in particular with simple or model-based reflex agents) the agent may be able to produce suitable control data directly from raw sensor data. With more complex types of agents, however, the production of control data will involve the ability to process information at a higher level of abstraction, for example by carrying out reasoning on *representations* of the environment and of the agent's actions. For example, a robot may need to interpret the sensor data (coming, for instance, from a camera) as representing a closed door, which will require substantial processing of the raw data (that, by themselves, are just a bit stream). In order to exit a room the robot may decide to open the closed door, and this decision will then require significant processing to be converted into suitable control data.

All this implies that the agent's controller should be broken down into different modules as shown in Figure 2.2 of these notes. The function of *perception* is to transform the sensor data into *representations of the environment* (e.g., a symbolic expression meaning "the door in front of me is currently closed"), which are then used to decide what to do. In turn, the output of the *decision* module consists of high-level *representations of actions* (e.g., a symbolic expression meaning "I now open the door"), which are then translated into control data by the *execution* module.

Perception (with real sensors) and execution (with real effectors) are often problematic in robotics, but are less so in applications where agents are software applications without a mechanical body, like for example software agents acting in the Web. In this course, in which robots are not a main concern, we shall concentrate only on the decision module.

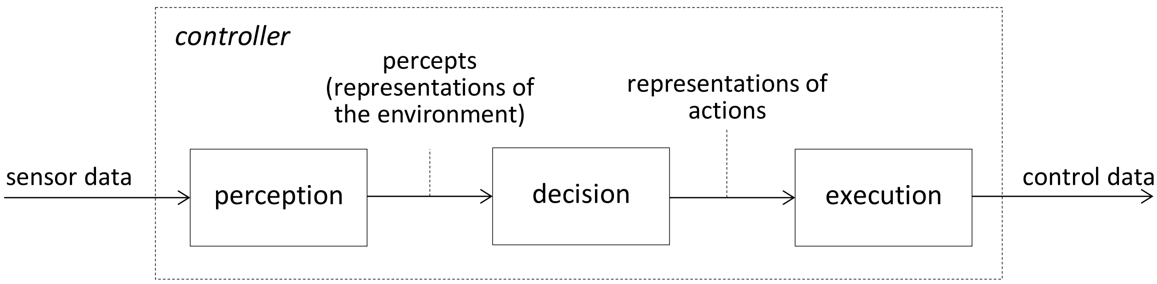

Figure 2.2. Explosion of an agent's controller.