# Agents and Multiagent Systems

In this section, we give one of the definitions for term agent as it is understood by [1] and [2] in the field of Artificial Intelligence. Understanding what (intelligent) agent is in terms of AI, has particular importance for us because our goal is to create one. More precisely an agent that would be able to play StarCraft and learn some of the decision making involved from professional players. Due to the complexity of StarCraft domain and the fact that there is no exclusive way how to build such an agent (actually many techniques are used together to do the job as is shown in section {link}) we designed our agent as Multiagent system (MAS). Therefore we also give some theory useful for our work regarding Multiagent Systems in this chapter.

## Description of (intelligent) agent

Currently, there is no widely accepted definition in AI community what agent is, so we use the definition from [1] and [2] which defines an agent as an **entity** that perceives its **environment** through **sensors** and **acting** upon that environment through **actuators**. This idea is illustrated on figure {link}. Despite the vagueness of this definition, it can frame the thing to which we refer as game playing agent well. In the case of a human agent playing StarCraft, we can consider player’s eyes as sensors and his hands as actuators giving commands to the game. On the contrary, given this definition, one can consider even the game program as an agent as it receives commands as sensory inputs and acts on the environment by displaying the game.

{figure of simple agent architecture}

One of the differences between those two agents is **autonomy** – one agent decides on commands to send, and another one executes them. Another key difference is that the first agent mentioned can be considered as **rational**. One of the definitions for rational agents based on [1] can be stated in the following manner. For each possible sequence of sensory input, a rational agent should select an action that is expected to maximize desirability of the resulting situation of the environment for the agent (design objective), given the evidence provided by sensory input and built-in knowledge agent has. A rational agent autonomously acting according to its best interest in every situation can be to some extent considered **intelligent**. Environment properties play a major role in the design of intelligent agent. As was shown in section {link} designing a rational agent to play StarCraft still imposes significant challenge. Current agents are not flexible enough in every situation. They very often lack **reactivity** - the ability to respond in timely fashion to changes that occurred.

### Types of agents

According to [1], there are four basic types of agent models embody the principles of intelligent agents:

* **Simple reflex agents** are also the simple kind of agents. Any action is taken only based on current sensory inputs. The action picking is based on if-then rules. A good example of such agents in StarCraft is a unit controllers programmed only in reactive fashion.
* **Model-based agents** keep some internal state of affairs of the environment that depends on the history of their sensory inputs using a model. The model represents knowledge how the world works. The action is then selected not just according to current sensory inputs, but state plays an important role too. Many agents in StarCraft use this approach as they are employing variants of finite state machines.
* **Goal-based agents** are based on the idea that knowing about the present situation of the environment may not be enough. Agents need goals to describe situations which are desirable. They are combining way how model-based agents choose actions with emphasis on actions which may lead to a goal. {give example, with bot following build order, Steamhammer?}
* **Utility-based agents** try to solve the problem of the goal-based agents that defining goals may still not be enough in complex environments like StarCraft. So they work with utility as a measure of desirability (preferences) for particular states of the environment. The architecture of this agent is on figure {link}. {give example, the bot picking build order against opponent, kraisi0?}

{figure of utility-based agent architecture}

## Description of Multiagent Systems

As was mentioned in previous section agent is in an environment. It is common that many agents share the same environment and there is a subset of agents where each agent must interact with agent different from itself. We refer to those kinds of environments with interconnected agents as Multiagent systems (MAS).