

# Reflection on Agent's Sensors and Effectors

M. Morsy Cheikhrouhou  
morsy.cheikhrouhou@eurecom.fr

October 13, 1997

## 1 Preamble

The agent must be continuous in time. This means that it has not to exit as soon as it could solve a problem. This explains why a normal rule-based program in Clips cannot be an agent (may be unless using functional and object-oriented programming in Clips).

## 2 Sensors and Effectors

### 2.1 Effectors

The least one can ask from an agent is to be able to sense the environment. Sensing the environment is an asynchronous task since changes occur asynchronously. AgentClips supports well the asynchronous exchange of Clips commands. Consequently, an agent in AgentClips performs a new pattern matching as soon as a new command is received.

Of course, sensors are part of the agent. A first type of sensors allows the agent to get a piece of information on demand. This kind of sensors may be seen as functions (from a programming point of view.) They can be parameterized in order to get a specific information from the same function when applied differently. This kind of sensors can be mapped from human beings when they touch something in order to recognize if it is hot or cold.

Another kind of sensors allow the agent to perceive changes in the environment. They can be seen as a human ear that only communicates sounds which the human is interested in. Concretely, such a sensor is a thread or a process from a programming point of view, and tells the agent about the information that occurs.

This kind of persistent sensors may also be useful to collect broad or general information. This information cannot be obtained in a functional manner. As an example, "*there were a lot of noise in the bus*" is deduced from hearing the sounds during one's travel in a bus.

The behavior of this kind of persistent sensor must be fully controlled by the agent's brain. For example, the agent might ask a sensor to be more or less sensitive to some kind of changes during its lifetime.

Is a sensor itself intelligent or not? In fact, there is no problem in having a sensor intelligent or not. What should be kept in mind is that a sensor's intelligence must not hide some knowledge that the agent might require. In that, a sensor can itself perform some information filtering and event correlation if only these operations do not prevent the agent from accessing useful information.

## 2.2 Effectors

Effectors can be seen as actions to be taken on the environment. They can be mapped to procedures from a programming point of view.

Can we imagine a kind of effectors to be in the form of asynchronous threads? This question should be further addressed and the answer is helplessly delayed for later. However, an effector taking the form of a thread maps well to a reactive agent. Nevertheless, one should clearly distinguish between what would be an agent and would be an effector in an agent.

## 3 Illustration using Clips syntax

Below, we try to illustrate the above concepts using Clips syntax.

A synchronous/functional sensor can be called from within the agent as:

```
(bind ?real-user-name (sensor finger get-real-name cheikhro))
```

Here, the agent wanted to get the full name of a certain account owner, here *cheikhro*. A functional sensor, here called *finger* is used for that purpose. *get-real-name* is what the agent asks from the sensor to provide, as same as a human brain asks his hand to touch a toast.

An asynchronous/persistent sensor must be launched and controlled by the agent. For example, an agent may launch a sensor for esteron printer:

```
(launch-sensor PrinterMonitor for esteron.eurecom.fr)
```

Later, the agent might require to know the number of jobs submitted to esteron:

```
(sensor PrinterMonitor how-many-jobs?).
```

This is a kind of information that requires continuous sensing of the printer. On the other hand, the agent might ask the sensor to notify it of incoming long jobs:

```
(sensor PrinterMonitor MonitorNewJobs)
```

Therefore, when a new long job is submitted to esteron, the sensor may send an event, in the form of a Clips template fact:

```
(event (from sensor PrinterMonitor) (at 14:30:20) (is a-new-long-job-is-submitted))
```

Finally, an effector is simply called via a Clips procedure:

```
(effector PrinterMonitor PutOffLine)
```

## 4 Further Considerations

Each Sensor/Effector must be related to all the hypothesis of its normal execution. What should be avoided here, is when one does not realize that a sensor/effector can no more run, or worse, gives no more reliable results because the system administrator has changed the location of the “ping” utility, or because he has evolved from version 3.0 to 4.03 of Netscape.

This kind of hypothesis will only be used to recall the system administrator for the moment. Eventually, the agent must perform itself these hypothesis checking and has to find the information required to perform it correctly.

The agent has also to have knowledge (and even to learn this knowledge) on the sensors and effectors it has in possession. It has to know how to use them and even the consequences of their utilization on the system (cpu and memory usage by sensors) and on the network (bandwidth usage).

## 5 Conclusion

An agent is persistent in time. In order to interact with its environment, it has to use sensors and effectors. Sensors can be either functional, or persistent. Functional sensors are used to get punctual information, whereas persistent sensors are used to get asynchronous notifications or to get processed information.

Effectors can be most seen as procedural actions. However, a kind of persistent effectors may be required to implement the reactive part of the agent.

Finally, the agent should in term include knowledge about its sensors and effectors and how to use them. It should also be aware of the consequences of their usage.