Prothonics

Agnese Salutari
agneses92@gmail.com, https://github.com/agnsal

Prothonics

- What is Prothonics?
- Environment Setup
- Proactive Agents
- Prothonics Structure
- How to Use?

What is Prothonics?

Prothonics is a Python 3 library (tested on Python 3.7).
Prothonics = Prolog + Python + Robotics.
You can use it to create proactive agents, that are able to take a decision in response to some event:

- You can provide an agent with a behaviour by writing it in Prolog files.
- You can define the event the agent has to react to in the Python program and in the Prolog files.

You can install such an agent on a robot or can run it in simulation environments.
You can download it from the following link:
https://github.com/agnsal/prothonics

What is Prothonics?
Prothonics =

- Prolog:
  SWI-Prolog (free)

What is Prothonics?
Prothonics =

- Prolog:
  SWI-Prolog (free)

  +

- Python:
  Python3.7

What is Prothonics?
Prothonics =

- Prolog:
  SWI-Prolog (free)

  +

- Python:
  Python3.7

  +

- Robotics:
  Sorry, Bender not included…

Environment Setup

1. Install Python 3.7 : https://www.python.org/downloads/
2. Install SWI-Prolog: https://www.swi-prolog.org/
3. Install Numpy: https://numpy.org/
    1. pip install numpy
4. Install Pyswip: https://github.com/yuce/pyswip
    1. Pip install pyswip
5. Download Prothonics: https://github.com/agnsal/prothonics
6. Include Prothonics inside your project: put prothonics subfolder in you project and import it.

Proactive Agents

An agent is an always running program (daemon).
It is proactive when it is able to detect changes (events) in the environment (world) it lives in and to react to them.
By detecting new events, it knows new facts about the world, so it learns:

Events can be noticed by sensors or can be communicated by others.

There are facts, different from events, that have a stable nature (they are time-independent):

These facts (rules) should be known a priori, and are mostly given to the agent in the Knowledge Base (KB) before execution.

Agent's behaviour depend on a desired situation (state) to reach (goal).

Proactive Agents:

An agent has to continuously perform the following steps:

- Sensing:
  To update its knowledge about the current state.
- Reasoning:
  To take a decision about future actions to do.
- Acting:
  To do the things it planned.

In most cases, it is important that this steps are not time-blocking:

Why?

Imagine someone who is walking and has to look around or move or think one at a time only…

Proactive Agents:
Uno Card Game Example

- The world is the play room.
- The rules are the rules of Uno game.
- Facts learned are things known while playing and change along with the state of the game (so the need to be updated):
  - Which cards the agent has now.
  - How many cards the other player has now.
  - …
- The goal is to win (to be the first player having no cards).

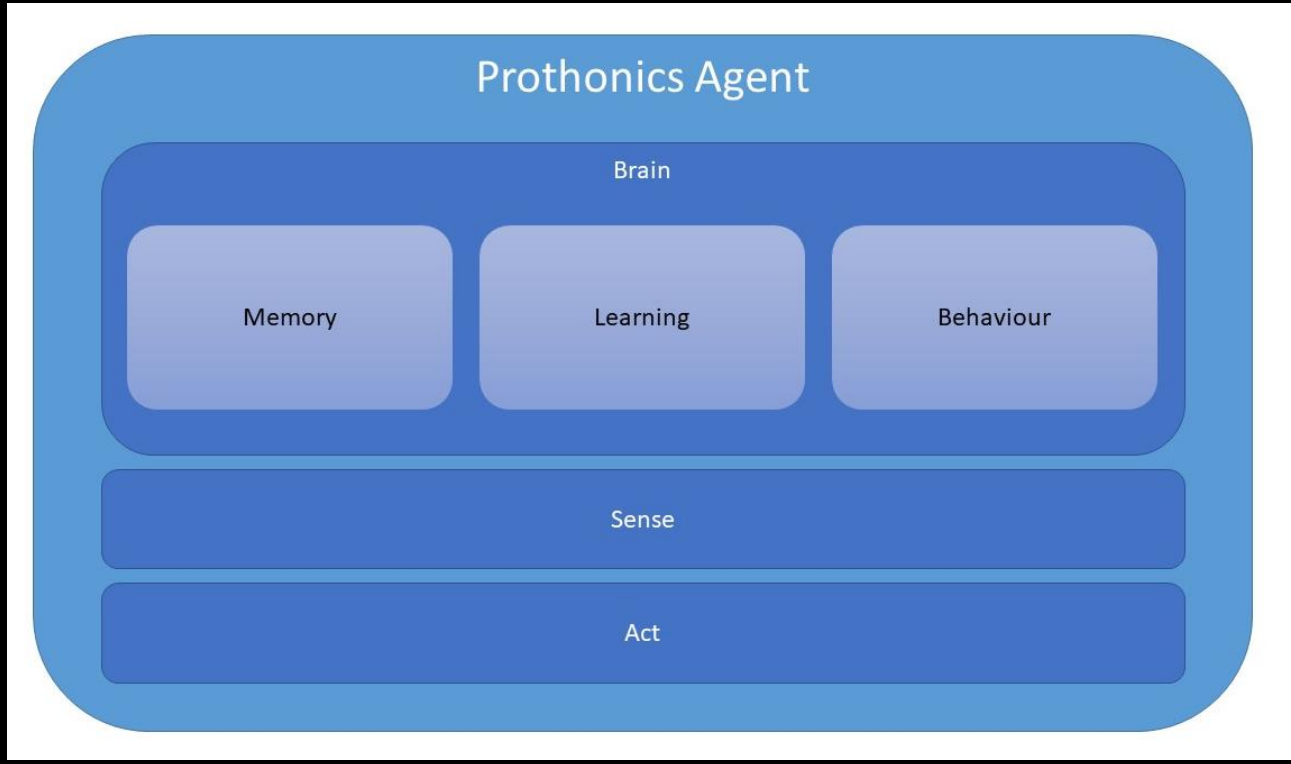
Prothonics Structure:

Prothonics is composed of several modules, each one is in charge to perform a particoular step, as you can see from its name

- Brain for reasoning:
  - Memory for storing taken decisions and learned facts.
  - Learning to learn new facts.
  - Behaviour to determine reactions.
- Sense for sensing.
- Act for acting.

Now you can use Brain Module:

Work in progress on the other ones!



How to Use?

There is a toy example you can refer to (__init__.py and behaviour.pl files in the main folder)

- It is about a square robot with 4 proximity sensors, one for each direction, to avoid obstacles.
- You can give information to the robots, simulating sensor readings.
- The robot takes decisions depending of sensor information.
- You can make queries to the robot.

PAY ATTENTION!

- Use double quoted strings for passing facts and queries to SWI-Prolog Engine!

How to Use? Remember…

Thanks for your attention