ÉCOLE POLYTECH'NICE SOPHIA
SI5
2020-2021

Julien Deantoni

# Behavioural Language Engineering

## Meta programming an interpreter/debugger/model checker

## Project delivery

The project must be delivered on XXXXX as a git repository. It must contain the following:

- the code of your language with a README in each ingredient folder (abstract syntax, rewriting rule, etc) to detail the key explanations.
- examples of the use of your language to control the robot + screen-shots of the development environment during simulation/debugging sessions.
- a global README that give the rational of your choices.

## Objectives: Defining a tooled executable language

Your objective here is to define, by using the meta languages provided in the GEMOC studio, the syntax and the behavioural semantics of a language dedicated to the control of an exploration robot named *polyCreate*. Your language will define a graphical concrete syntax that will be used to represent useful debugging information about the model state. You can also equip your language with a textual concrete syntax (typically to edit the model). In order to elaborate your language, it might be useful to think about existing behavioural languages like the one in UML and to browse the languages available in `http://gemoc.org/gallery.html` or `https://www.eclipse.org/sirius/gallery.html`.
Eventually, it is expected to provide a development environment with an omniscient debugger, model animation (and a model checker, if possible).

## Development steps

to develop your language, do not try to have a pure waterfall approach. Adopt an iterative development in which, for each step, you will

- define (a part of) the syntax (abstract and concrete)
- define the corresponding behavioural semantics
- provide appropriate debugging representation(s)
- make a test of your language in several execution mode (execution, debugging, omniscient debugging, model checking)

## Exploration Robot Control

Your objective is to define a full development environment based on an appropriate language to control of an advanced vacuum cleaner robot. The robot to control is based on the webots cyberbotics environment (`https://cyberbotics.com/`). This environment allows several robots and scenes but the expected language is restricted to the control of the *PolyCreate* robot defined in the repository referenced below.

Figure 1: a view of the proposed simulation environment

In `https://github.com/jdeantoni/polytechWebotsController` you will find a Java abstraction layer on top of the cyberbotics remote API. while not sufficient for all the required purpose, it can be used as a starting point. Your language should allow the realization of different missions where the robot is programmed towards specific objectives. The simulation and debugging of the program should control *polyCreate* in the simulated environment. This means that the time in your execution/debug session and in the simulator should be synchronized.
You are the boss concerning the expressiveness of your language; however it is obviously required to deal with the notion of time. During the development of your language, you should be aware of what is the *State* of your system (and document it) and if it allows for exhaustive exploration as well as V&V activities.