City Dynamics: a Multi-Agents Simulation

Alice Turrini, Maud Ravanas-Deshays

Multi-Agents Systems 2023-2024

Abstract. Through this project, we aim to simulate the traffic flows in a city. The main characters of our city are cars, that drive around in the streets, and pedestrians that walk within buildings. They will have to interact with each other, especially in complex zones like crossroads or zebra crossings they have also to rely on their perception of the environment to achieve their specific goal. These agents will be autonomous and intelligent, capable of decision-making and interaction.
To implement and manage such a system, we will use the JaCaMo development platform[2], where Jason is used as the reference agent programming language to implement agents driven by their beliefs and their goals (BDI architecture). For the environment, there is Cartago [1], a framework that provides an infrastructure for creating, managing, and interacting with artifacts in a shared custom environment.

1 Description

1.1 Goals

Our main goal is to implement seamless pedestrian and car traffic flows in a city. A smart way to deal with this situation is to approach it as a multi-agent system. Our different agents will rely on their beliefs, goals, and plans to reason autonomously and behave coherently, and they will communicate with each other.

Environment Our environment, which represents a city, consists of a map displayed as a grid. Here are described in more detail the possible blocks:

- **Buildings:** They can host only pedestrian agents who are free to walk inside them. There are four different building types: a supermarket, a school, a park, and an office. They are disconnected from each other and the only possible way to connect is the zebra crossings. Pedestrians can hence walk inside a building or leave it and cross the street at the corresponding points to go to another building.
- **Streets:** They are only frequented by cars, except for the zebra-crossing parts. A block can only contain one car at a time. Each block is associated with one direction because it represents a road's lane. To make it even more realistic there are crossroads, where cars will interact with the environment to avoid incidents, indeed some lanes will have precedence over others.
- **Garage:** This specific block is where a repair Helicopter is parked. It cannot be used by any other type of agent.

Agents There are three different types of agents, that will be able to communicate with each other.

- **Cars:** They drive randomly around the city, using their percepts to follow the street lanes and be aware of their neighbors, with their associated properties (whether it is other vehicles or pedestrians). As said before, two cars can't be in the same block. A car must always follow the direction of the street block occupied. If it is possible it can change roads, and then continue in the corresponding direction, but precedence must be given to the crossed lane while turning.
  When a car approaches a zebra crossing, it has to pause and let the pedestrians cross if there are any waiting on the side of the road.
  Cars can randomly break down, and in that case, they have to call the helicopter to ask for a repair.
- **Pedestrians:** They can walk only inside buildings and between them using zebra crossings. Multiple pedestrian agents can be at the same time in the same block. They initially pop up in one of the four buildings and their goal is to head towards some specific places.
  When they use the zebra crossing, if a car stops to let them pass they have to thank them.
- **Helicopter:** This is a special vehicle parked at the garage until a car sends a message asking for help. Once the helicopter arrives at the rescue, it will go back to its initial place.
  To avoid congestion this vehicle will fly, and this allows it to pass through street-occupied blocks.

1.2 Deliverables

We aim to deliver a software which will consist of the implementation, based on JaCaMo, of our city simulation. In addition, we will also provide a final report to explain the project more in-depth, as well as a presentation for the support for the discussion.

References

[1] *Cartago*. URL: https://sourceforge.net/projects/cartago/.

[2] *JaCaMo*. URL: https://jacamo-lang.github.io/.