Traffic Simulation with SARL

EuSarlCon 2019 - May 2nd 2019

Dr. Jocelyn BUISSON  Yazan MUALLA  Dr. Alexandre LOMBARD
Prof.Dr. Stéphane GALLAND
1. Introducing a framework for agent-based simulation

2. Driving Activity

3. Agent Environment Model

4. Agent Architecture

5. Results
1 Introducing a framework for agent-based simulation

2 Driving Activity

3 Agent Environment Model

4 Agent Architecture

5 Results
Introducing a framework for agent-based simulation

Driving Activity

Agent Environment Model

Agent Architecture

Results
Introducing a framework for agent-based simulation

**Driving Activity**

**Agent Environment Model**

**Agent Architecture**

**Results**
Introducing a framework for agent-based simulation

Driving Activity

Agent Environment Model

Agent Architecture

Results

GENERAL AGENT-BASED SIMULATION ARCHITECTURE

Agent

Direct interaction

Environment
- Resources, services, objects
- Rules, laws
- Physical structures (spatial and topological)
- Communication structures (stigmergy, implicit communication)
- Social structure

Perceptions

Actions

Simulation Controller

Observer

Rendering Software Modules
(1D, 2D or 3D)

Change events
Introducing a framework for agent-based simulation

Driving Activity
Agent Environment Model
Agent Architecture
Results
SITUATED ENVIRONMENT MODEL

Introducing a framework for agent-based simulation
Driving Activity
Agent Environment Model
Agent Architecture
Results
Introducing a framework for agent-based simulation

Driving Activity

Agent Environment Model

Agent Architecture

Results

Galland (2009)

Agent

Agent mind

Behavior

Agent memory

Agent body

Physical attributes (x,y,z), V(t), a(t)

Perception filter

Filtered perception

Action

Action filter

Influence

Environment Interface

State variables of the decisional component
Readable/modifiable only by the agent

State variables of the physical component
Readable by the agent
Modifiable by the environment

State variables of the decisional component
Readable/modifiable only by the agent
How to support simultaneous actions from agents?

1. An agent does not change the state of the environment directly.
2. Agent gives a state-change expectation to the environment: the influence.
3. Environment gathers influences, and solves conflicts among them for obtaining its reaction.
4. Environment applies reaction for changing its state.
1 Introducing a framework for agent-based simulation

2 Driving Activity

3 Agent Environment Model

4 Agent Architecture

5 Results
Each vehicle is simulated but road signs are skipped \(\Rightarrow\) mesoscopic simulation.

The roads are extracted from a Geographical Information Database.

The simulation model is composed of two parts (Galland, 2009):
1. the environment: the model of the road network, and the vehicles.
2. the driver model: the behavior of the driver linked to a single vehicle.
1 Introducing a framework for agent-based simulation

2 Driving Activity

3 Agent Environment Model

4 Agent Architecture

5 Results
Road Network

- Road polylines:  $S = \{ \langle \text{path}, \text{objects} \rangle | \text{path} = \langle (x_0, y_0) \cdots \rangle \}$
- Graph:  $G = \{ S, S \mapsto S, S \mapsto S \} = \{ \text{segments, entering, exiting} \}$

Operations

- Compute the set of objects perceived by a driver (vehicles, roads...):

  $$ P = \left\{ o \middle| \begin{array}{c} \text{distance}(d, o) \leq \Delta \land \ o \in O \land \ \forall (s_1, s_2), \text{path} = s_1 \cdot \langle p, O \rangle \cdot s_2 \end{array} \right\} $$

  where $\text{path}$ is the roads followed by a driver $d$.

- Move the vehicles, and avoid physical collisions.
The agent has the capacity to use the car.
The body supports the interactions with the environment.

```kotlin
event Perception {
    val objects : List<SituatedObject>
}

capacity EnvironmentInteraction {
    def getBody : AgentBody
    def move(a : float, path : List<RoadSegment>)
}

event CarInfluence {
    val acceleration : float
    val path : List<RoadSegment>
    new (a : float, p : List<RoadSegment>) {
        this.acceleration = a
        this.path = p.unmodifiableList
    }
}

skill PhysicBody implements EnvironmentInteraction {
    val env : UUID
    val body : AgentBody

    def getBody : AgentBody {
        this.body
    }

    def move(a : float, path : List<RoadSegment>) {
        emit(new CarInfluence(float, path))
        [UUID == this.env]
    }
}
```
1 Introducing a framework for agent-based simulation

2 Driving Activity

3 Agent Environment Model

4 Agent Architecture

5 Results
Jasim model (Galland, 2009)
Based on the A* algorithm (Dechter, 1985; Delling, 2009):
- extension of the Dijkstra’s algorithm: search shortest paths between the nodes of a graph.
- introduce the heuristic function $h$ to explore first the nodes that permits to converge to the target node.

Inspired by the D*-Lite algorithm (Koenig, 2005):
- A* family.
- supports dynamic changes in the graph topology and the values of the edges.
- **Principle**: compute the acceleration of the vehicle to avoid collisions with the other vehicles.
- **Intelligent Driver Model (Treiber, 2000)**

\[
\text{followerDriving} = \begin{cases} 
-\frac{(v\Delta v)^2}{4b\Delta p^2} & \text{if the object ahead is far} \\
-a\frac{(s + vw)^2}{\Delta p^2} & \text{if the object ahead is near}
\end{cases}
\]

- **Free driving**:

\[
\text{freeDriving} = a \left( 1 - \left( \frac{v}{v_c} \right)^4 \right)
\]
1 Introducing a framework for agent-based simulation

2 Driving Activity

3 Agent Environment Model

4 Agent Architecture

5 Results
Simulation of Lane Changing Behavior (Lombard, 2017)
Simulation of emergency situation on a french highway (Buisson, 2014)
Simulation of fog situation in Qatar (Abbas-Turki, 2017)

Traffic and V2X simulation

Comparison between scenarios in fog situation
Thank you for your attention...
Appendix


