e-SMARTS: A System to Simulate Intelligent Traffic Management Solutions (Demo Paper)

ACM SIGSPATIAL 2022

Authors: Udesh Gunarathna, Dr. Renata Borovica-Gajic, Prof. Shanika Karunasekera,
And Prof. Egemen Tanin

Need for Customizable & Realistic Simulators?

- Increased connectivity and the availability of smart devices in road networks generate a vast volume of data
- Vehicles, road segments, and intersections can be thought of as intelligent agents
- AI techniques and ML have found a fertile application domain in traffic engineering by leveraging these data for novel traffic management techniques

- Need simulators that mimic real-world dynamics

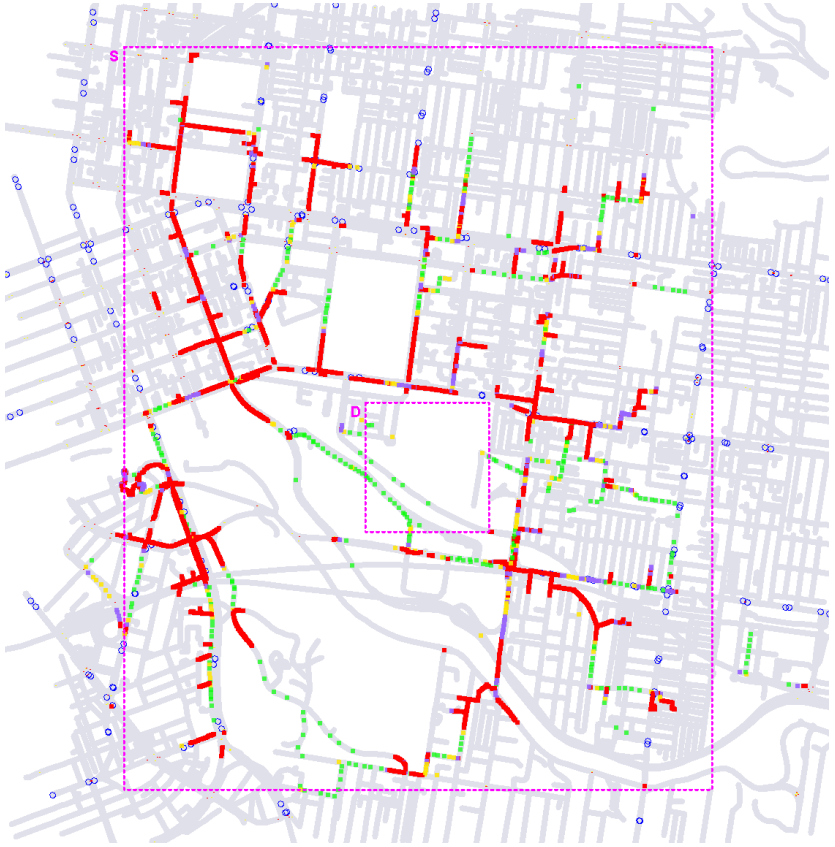
- Such simulators are less customizable i.e., most simulators need to be recompiled by changing the source code to test new algorithms, which is time-consuming

- Many intelligent agents need to be simulated in novel traffic management techniques and the existing simulators are not flexible to achieve this

We need a realistic simulation environment where new algorithms can be connected seamlessly that developed in any programming language

e-SMARTS

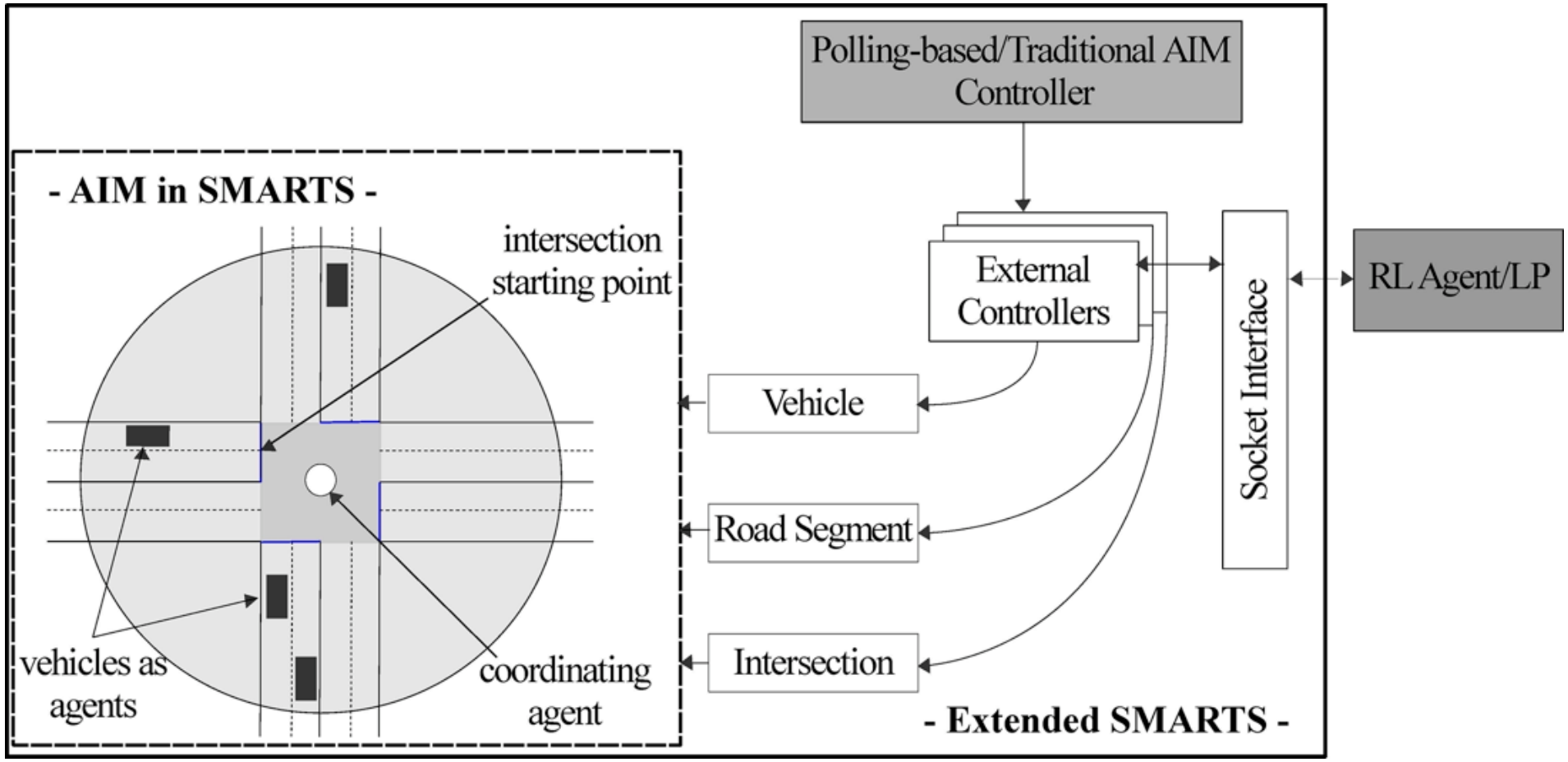
- e-SMARTS is an easily extendible simulation environment which is built on top of SMARTS to test novel traffic management solutions
- SMARTS is a distributed microscopic simulator which is capable of simulating very large traffic networks and capable of mimicking real-world dynamics
- e-SMARTS allows researchers to test their new traffic management solutions in a realistic setup which accelerate the deployment of these solutions



A simulation in SMARTS where red color indicates high traffic density

e-SMARTS Overview

Autonomous Intersection Management (AIM)



Message-Passing between e-SMARTS and an External Controller

Ease of Use Over SMARTS

- No need to understand internal implementations of SMARTS components, only the interface
- No need to be restricted to SMARTS's language JAVA (often ML algorithms are developed in other languages)
- Easy to change the external algorithm, even at runtime

```
{
  "vehicles": [
                {"index": <vehicle_id_1>,
                 "paddleCommand": <command>},
               {"index": <vehicle_id_2>,
                 "paddleCommand": <command>},
              ],
  "roads": [
                {"index": <road_id>,
                 "lane": <lane_id>,
                 "changeDirection": <command>},
              ...
              ],

   "intersections": [
                {"index": <intersection_id>,
                 "updateSchedule": <command>},
              ...
              ],
   "settings:" : {
                "demand":<value>,
                "<command_name>": <value>
                }
}
```

Example message passing