Generalizing Model-Scale Experiments in Networked and Autonomous Driving Using a Service-Oriented Architecture

(Master Thesis)

JOHANNES ROLF

Motivation

Several model-scale laboratories offer the possibility for experiments in autonomous and cooperative driving. Two examples are the Cyber-Physical Mobility (CPM) Lab here at the i11 and the Scaled Smart City of the University of Delaware (UDSSC). Both labs provide a testbed to conduct experiments by using scaled vehicles.

An experiment consists of a scenario and one or multiple high-level controllers (HLCs). A scenario defines the traffic situation, i.e., the road geometry, where obstacles are, and how many vehicles exist, while the HLC controls the behavior of the vehicle(s) within the scenario. Since both labs base on different software, the HLCs and scenarios need to be implemented for each lab individually, even if the underlying experiment should be the same. This master thesis shall overcome this issue.

State of the Art

Another work addresses a similar purpose. It provides a generalized Java-based API for different traffic simulators. Consequently, the user is able to program controllers for traffic participants independently of the simulator and can evaluate and compare the results of the same experiments in different simulators. Although this work achieves real-time interaction with the simulators, the real-time capabilities become much more crucial in my work addressing laboratories instead of simulators. This fact is also visible in the different choices of programming languages since both labs use C++ instead of Java.

Objective

In order to generalize the experiments for model-scale labs, I shall define an API on which basis HLCs and scenarios can be programmed lab-independently. It must specify the data exchanged between the lab, the HLC, and the scenario. As part of that, it shall provide the possibility to retrieve a standard scenario for each lab, i.e., a traffic situation specifically suitable for experiments in the concrete lab. Additionally, since both labs have different scales of their streets and vehicles, some utility shall be available which can scale scenarios dependent on each lab. With that users can also write their scenarios in real-life scale and still use it in the model-scale labs.

Approach

For free choice of the programming language for HLCs and scenarios I will develop a service-oriented architecture based on version 2 of the Robot Operating System. Consequently, HLCs, scenarios, and the labs can be seen as services. The very first step is thus to compare and evaluate the currently lab-specific experiments to find a way to generalize them. As a result, I can define the message types which can be exchanged between these services. Since both labs are developed in C++, I shall develop a C++ API allowing to send these messages easily. This I can use to develop the lab-service, which I must do twice – once for the CPM Lab and once for the UDSSC. Finally, I will run an experiment based on a simple HLC and the default scenario of the UDSSC.

Supervisor: Patrick Scheffe
16.01.2023

Informatik 11
Embedded Software

RWTH AACHEN UNIVERSITY