Configurable Abstractions for Agile Development of Driving Tasks in OpenDS

School of Computer Science University of Nottingham Ningbo China

Supervisor: Dr Dave Towey

Zilin Song(16522063)

1 Background and Motivation

There is an idea that human beings are not good drivers and it has been widely confirmed in many cultures [1]. Although this idea is somewhat extreme, according to a road safety report for 2015, drivers are sometimes distracted, drunk and tired which often lead to car accidents [1]. To reduce traffic accidents, more and more people focus on developing auto driving in order to provide a safer and more efficient driving [6]. In order to develop safe and reliable autopilot software, a lot of experiments are needed to carry out in laboratory before real driving experiment. When experimenting in the lab, researchers need to use simulators and build driving scenarios to simulator real experiments.

OpenDS is a mainstream tool for cognitive research to support in-lab driving simulations [5]. This simulator allows users to build scenarios and define driving behaviors by themselves. In OpenDS, every scenario is defined in five XML files and users modify the scenario by modifying the XML files. To build scenarios, people need to build several driving tasks first and then merge them together, there are also some functions in OpenDS which can be used to build driving tasks. There is a function called autopilot function which allows the car to driver automatically according to pre-set routes and speed(Figure 1). However, when I was working on building scenarios for Prof. Sun in this summer, I found the process of building scenarios is complicated and time-consuming and implementations of specific driving tasks in corresponding scenarios is inevitable [8].

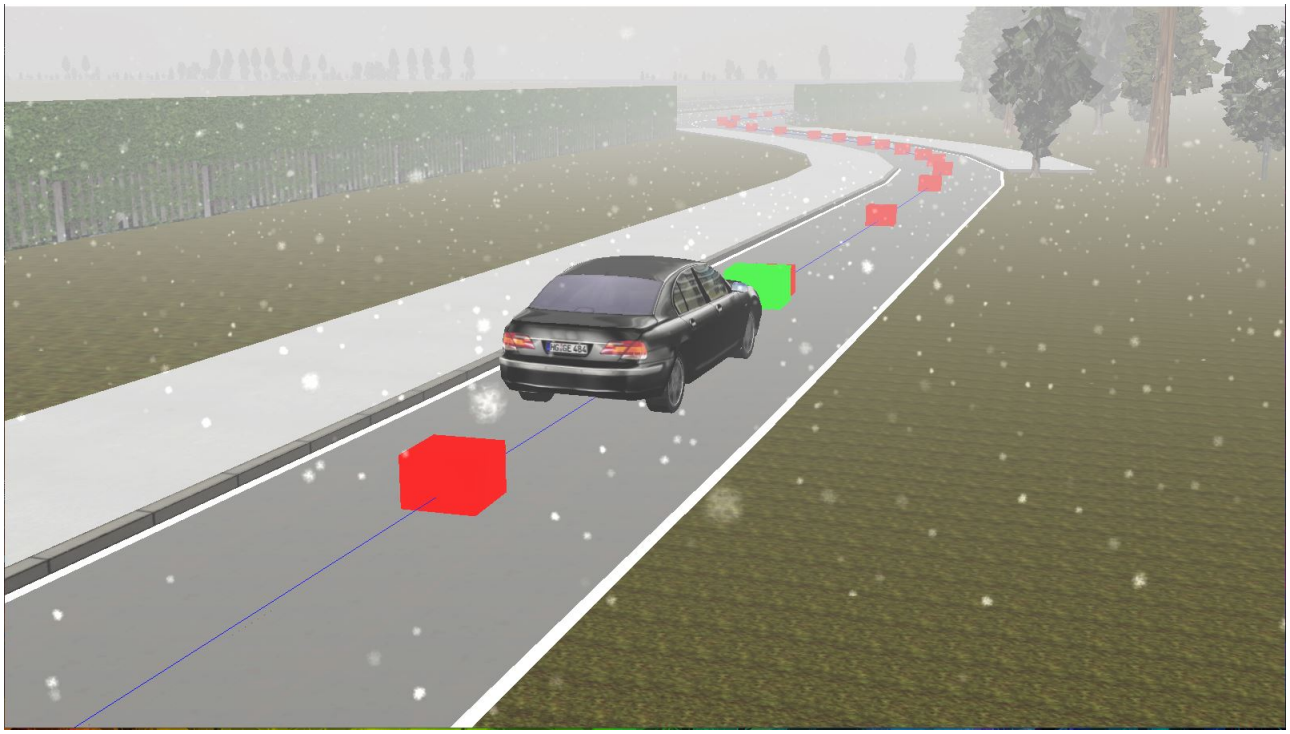

Figure 1: OpenDS pre-set route

There are several problems which lead to low efficiency when building driving scenarios. To better simulate real driving, OpenDS defines lots of variables, such as maximum/minimum vehicle speed and maximum acceleration.

When people build driving scenario, all the variables need to be considered. When one variable is changed, the whole scenario may need to be modified and it will bring great inconvenience to users. Meanwhile, different environment condition will also affect the driving behaviors, such as the minimum, maximum speed limit and the road width. For example, a change in the friction coefficient will cause a change in the distance when the vehicle brakes and this may lead to researchers change vehicle route. In addition, one driving scenario usually needs several driving tasks and some of these driving tasks may be highly similar. However, for those similar driving tasks, as long as there is only one parameter is changed, people usually need take a lot of time to re-adjust other parameters. For instance, there are three crossroads with different road widths in one scenario and every time the car passes the crossroad, it needs to turn left. However, due to the different coordinates and road widths when these behaviors occur, researchers need to re-configure the path variables.

A configurable function is the function which allow users to get different results by entering different parameters and it can reduce to the complexity of program operations and helpful to cater individual needs [7, 2]. When people study some problems, they usually intentionally ignore some trivial details to simplify the problem(abstraction) and abstraction makes the software more reliable, easy to maintain and quick to modify [3, 4]. In my opinion, almost all the driving behaviors in OpenDS can be abstracted into mathematical or physical models. For example, the trajectory of the turn can be abstracted into a part of the circle. If people can know several key variables, then they can deduce the entire driving behavior by model. As a result, they do not need to modify every variable by themselves and reduce the process of adjusting variable, which will highly improve the efficiency of building scenarios in OpenDS.

2 Objectives and Goals

The aim of this project is to collect and integrate some common driving behaviors based on OpenDS. The result of this project should be a java library. When people want to build driving behaviors, they can import this library, quickly building specific driving behaviors by calling corresponding functions and entering needed variable values.

- Background investigation of automotive driving research, learn more about how OpenDS works and the knowledge about building physical or mathematical models.
- Investigations and summary of Common Driving Behaviors and check if these driving behaviors are available in OpenDS.
- Analyze the driving behaviors and abstract these driving behaviors into physical or mathematical models.
- Implementations of Configurable Abstractions for Driving Tasks (CAfDT).
- Optimize the code structure and implement relative position calculations to improve usability.
- Examples for specific scenarios implementations based on CAfDT.

3 Project Plan

Note: Gantt Charts are attached in the appendix.

For this project, as all the CAfDT need to be supported by OpenDS, understanding the running principle of OpenDS and analyze whether it can be implemented in OpenDS need to be treated carefully. Sometimes multiple work will be carried out at the same as all the CAfDT are independent. It will help increase productivity and reduce risk.

A: Write project proposal.

B: Literature review, looking for background information, useful tools to draw models and modify scenario file (XML file).

C: Learn math and physical modeling knowledge, view OpenDS codes to understand the operating mechanism of autopilot

D: Collect some information about the most common driving behaviors when people driving (User Study)

E: Check whether these driving behaviors can be implemented in OpenDS

F: Building mathematical or physical models for driving behaviors

G: Write International report

H: Design the code framework

I: Prepare for exam

J: Chinese New Year holiday

K: Implementation the basic function of generation of driving behavior

L: Optimize code to improve the performance

M: Testing the code, fix bugs and evaluate performance (SQA)

N: Write final report

O: Prepare for final presentation

References

[1] Lex Fridman, Daniel E. Brown, Michael Glazer, William Angell, Spencer Dodd, Benedikt Jenik, Jack Terwilliger, Aleksandr Patsekin, Julia Kindelsberger, Li Ding, Sean Seaman, Alea Mehler, Andrew Sipperley, Anthony Pettinato, Bobbie D. Seppelt, Linda Angell, Bruce Mehler, and Bryan Reimer. MIT advanced vehicle technology study: Large-scale naturalistic driving study of driver behavior and interaction with automation. *IEEE Access*, 7:102021–102038, 2019.

[2] Helmut Hoyer and Ralf Hoelper. Intelligent omnidirectional wheelchair with a flexible configurable functionality. In *Proc. RESNA Annual Conference*. ERIC, 1994.

[3] Leo Joskowicz. Simplification and abstraction of kinematic behaviors. In *Readings in qualitative reasoning about physical systems*, pages 597–602. Elsevier, 1990.

[4] Barbara Liskov and John V. Guttag. *Program Development in Java - Abstraction, Specification, and Object-Oriented Design*. Addison-Wesley, 2001.

[5] Rafael Math, Angela Mahr, Mohammad M Moniri, and Christian Müller. Opends: A new open-source driving simulator for research. In *Proceedings of the International Conference on Automotive User Interfaces and Interactive Vehicular Applications, Adjunct Proceedings*, pages 7–8, 2012.

[6] Qudsia Memon, Muzamil Ahmed, Shahzeb Ali, Azam Rafique Memon, and Wajiha Shah. Self-driving and driver relaxing vehicle. In *2016 2nd International Conference on Robotics and Artificial Intelligence (ICRAI)*, pages 170–174. IEEE, 2016.

[7] Hani Naguib and George Coulouris. Towards automatically configurable multimedia applications. In *Proceedings of the 2001 international workshop on Multimedia middleware*, pages 28–31. ACM, 2001.

[8] Zilin S, Shuolei W, and Xiangjun P. Github: unnc-idl-ucc/scenario-opends, 2019.

Appendices

Appendix **Gantt Charts**

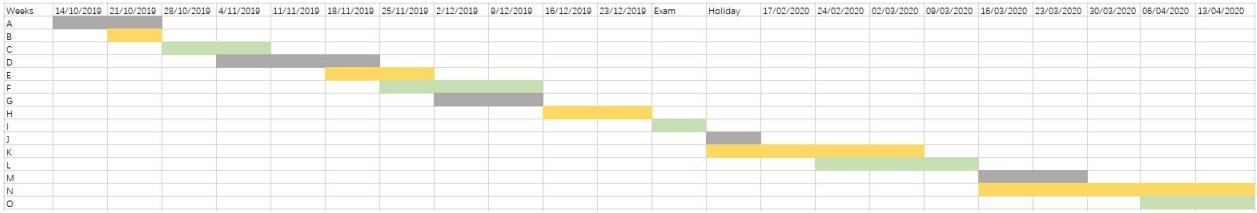

Figure 2: Gantt Chart