



Real Time Traffic Simulation

With SUMO - Project Overview

25.11.2025

Team Members:

Thanh Hung Phan

Quy Trang Duong

Quang Minh Khoa Le

Vo Minh Khoi Nguyen

Duc Khang Nguyen

1. Introduction

Objective

In this project, our team is implementing an Application that utilizes SUMO (Simulation of Urban MObility) to simulate the traffic, allowing the user to interact with the map and stress testing the traffic conditions. Alongside with the real-time simulation of traffic, we are going to implement statistical analysis using Multi-Threading technique.

Roles

There are 5 members in our team. Therefore, we decided to split the project responsibility into 5 main parts:

Person in charge: Thanh Hung Phan

- Project Manager and Renderer: In charge of how the project is structured, making sure that all the components work well with each other as well as satisfy all of the core features. Also, in charge of map interaction handling.

Person in charge: Quy Trang Duong

- Front-end Designer: In charge of designing the User Interface as well as implementing those the Renderer for the GUI, ensuring it can smoothly render the data fetched from SUMO through TraaS with lively visualization.

Person in charge: Quang Minh Khoa Le

- Simulation Manager: In charge of connecting the SUMO with the team Application, making sure that the data fetched is correctly, smoothly. Also, be responsible for the map Edge, Route, Lane and co-implementing the Multi-threading logic.

Person in charge: Vo Minh Khoi Nguyen

- Object Behaviours Manager: Overall manager of objects in the map. Specifically in charge of Vehicles, implementing VehicleInjection, fetch Vehicles Data and Stress Testing. Also co-implementing the Multi-threading logic.

Person in charge: Duc Khang Nguyen

- Statistical Analysis Manager: Main responsible for the Statistical Analysis, making sure that the Statistic is receiving correct data, optimizing the statistical computation and export data as of the core feature. Also, implemented a Traffic Light logic..

2. Implementation

Proposed Architecture

Our Application divided into 4 main components:

GUI: This is the interface of our map, everything has to be standardized and rendered through here.

- Renderer
- CoordinateConverter
- MainView.fxml
- MapInteractionHandler

Controller

- MainController
- MainGUI

SUMO

- SimulationManager
- MapManager
- VehicleManager
- TrafficLighManager

STATISTIC

- StatisticsManager
- ReportManager

Proposed Technology

- Programming Language: Java 17
- Eclipse IDE
- Apache Maven (Build tool)
- SUMO TraaS (TraCI as a Service) API
- JavaFX (21.0.2) for Visualization
- ExecutorService (Multi-threading)

Proposed Workflow

We planned to run our Project in 3 separate Threads:

- Main thread running the JavaFX and GUI

- One thread runs the Connection with SUMO
- One thread runs the statistical analysis

Based on the speed and smoothness of the application, we would have to adjust the threading logic accordingly.

Main Features

Live SUMO Connection: connects to a running SUMO simulation, do action like step time, change the map visualization in real-time.

Interactive Map Visualization: Render road net-work, display vehicles, traffic lights, support zooming, panning, camera rotation, filtering objects on the map.

Map control: Inject vehicles, changing Traffic Light durations, phases and states. Allow stress testing (inject an amount of vehicles to the map), control over the vehicles parameter.

Statistical Analysis and Report: tracks statistical information from the map and plots them up in real-time. Save the simulation statistics to a CSV for external analysis and generate PDF for charts and graphs (including filters).

3. Summary

The Real Time Traffic Simulation project provides an interactive application that integrates SUMO with Java, enabling users to visualize, control, and analyze traffic in real time. The system utilizes multi-threading to separate simulation, statistics, and GUI rendering, ensuring smooth and accurate data processing. The application demonstrates an object-oriented, thread-safe design suitable for scalability and maintenance for future development.