Real-Time Traffic Simulation System Project Overview

1. Project Background

SUMO (Simulation of Urban Mobility), as a core open-source traffic simulation tool, boasts powerful road network modeling and traffic flow simulation capabilities, widely used in research and teaching.

The goal of this project is to create an interactive traffic simulation system that links a custom Java application with the SUMO platform. Users will be able to view traffic scenarios, monitor the simulation in real time, and interact with traffic entities like vehicles and traffic lights. Using the TraaS library, the application communicates with SUMO via the TraCI protocol, allowing data retrieval and control of simulation elements.

2. Objectives

This project aims to build a "stable, user-friendly, and extensible" integrated real-time traffic simulation system based on Java and SUMO, with specific objectives including:

- Establish a bidirectional communication framework between Java and SUMO using the TraaS interface library, ensuring real-time data synchronization and accurate execution of control commands.
- Implement basic simulation functions such as road network loading, vehicle operation, and traffic light control to meet core research and teaching needs.
- Develop an intuitive GUI to simplify operation processes and reduce the usage threshold for non-professional users.
- Adopt a modular design with clear hierarchical boundaries to reserve expansion interfaces for subsequent function iterations.

- Verify the feasibility of the integration solution and form reusable technical templates and development specifications.

3. Core Functions

3.1 Full Simulation Lifecycle Control

- Support start, pause, step-by-step execution , reset, and termination operations.
- Compatible with custom road network configuration files , allowing settings of simulation duration, traffic density, and other parameters to adapt to various traffic scenarios.

3.2 Traffic Entity Management

- Vehicle Management: Support manual injection and automatic injection , with real-time collection of speed, position, and other status data.
- Traffic Light Management: Support manual switching of red/yellow/green states and cycle adjustment , synchronously displaying working status.

3.3 Visualization Display

- GUI presents road network topology maps and real-time vehicle position annotations.
- Equipped with a data dashboard that displays key indicators such as total vehicle count, average speed, simulation duration, and traffic light status, intuitively presenting simulation processes and results.

4. Technology Stack

Technology/ Tool	Version Requirement	Selection Rationale
Java	JDK 11+	Cross-platform compatibility, object-oriented design for modular development, Swing framework supporting GUI development, and multi-threading ensuring real-time processing.
SUMO	1.18+	Open-source and mature, powerful road network modeling and simulation capabilities, rich community resources, reducing early development costs.
TraaS	1.0	Official Java interface encapsulating the underlying

		TraCI protocol, simplifying communication development and ensuring compatibility with SUMO.
Maven	3.6+	Unified project dependency management, ensuring consistency across team development environments.
Git	Latest	Supports version control and team collaboration, standardizing development processes.
Draw.io	Online Version	Conveniently creates architecture diagrams and class diagrams, enhancing document professionalism.

5. Expected Outcomes

5.1 Technical Outcomes

Form a standardized Java-SUMO integration solution, delivering reusable toolkits for TraaS communication and entity interaction, as well as architecture templates and technical documentation.

5.2 Functional Outcomes

Deliver a complete simulation system with SUMO connection, vehicle/traffic light management, visualization display, and data export (CSV/PDF), meeting the needs of basic traffic experiments.

5.3 Implementation Outcomes

Provide a simplified user guide and test cases; the modular architecture reserves interfaces to directly extend advanced functions such as adaptive control.