

Class Design for Libtraci Wrapper

1. Design Goals

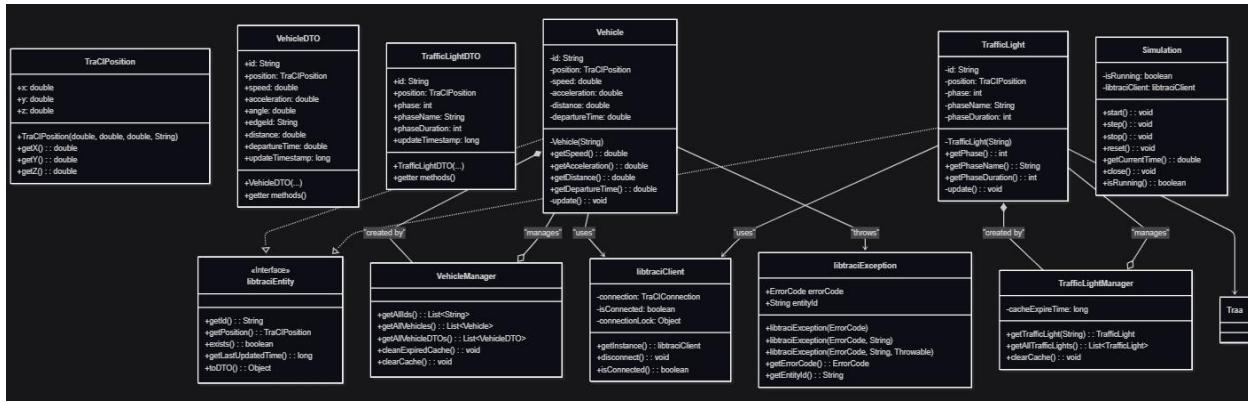
- Encapsulate the native libtraci interfaces and offer a type-safe, object-oriented API instead of using imperative calls.
- Hide TraCI protocol details.
- Provide exception handling approach.
- Ensure thread safety.

2. Relationships Between Classes

2.1 Main Classes

- SumoBusinessService is the core business logic and it relies on Vehicle, TrafficLight, and Simulation.
- SumoBusinessService is the core business logic and it relies on Vehicle, TrafficLight, and Simulation.
- SimulationData is an immutable DTO used to transfer simulation stats from SumoBusinessService to SumoMainFrame for updating the UI.
- VehicleManager and TrafficLightManager handle caching of entities and batch data retrieval.
- libtraciClient wraps TraCI protocol communication, including connecting, sending commands, and handling exceptions.

2.2 Relationship Diagram of Core Classes



3. Exception Handling

- All business exceptions are wrapped in `TraaSException`, which includes an error code and entity ID.