SustainCity Project

Software Engineering for HPC: Requirement Engineering and Design Document

Version 1.0

Release Date: April 3, 2025

Table of Contents

1. The Project and Project Goals

The SustainCity project aims to address two urgent global concerns: environmental sustainability and climate change. Urban commuting significantly contributes to air pollution and greenhouse gas emissions, worsening these issues. Our goal is to develop a comprehensive software system that helps manage and optimize urban traffic through the following types of actions:

Type1: Dynamic Traffic Light Management

The system will dynamically modify the duration of traffic lights on main city roads based on real-time traffic conditions. For example, if a significant increase in traffic flow is detected on road A compared to its crossing roads, the system will extend the duration of green lights on road A (and consequently extend red lights on crossing roads) for a specified period.

Type2: Traffic Pattern Analysis and Optimization

The system will analyze daily traffic patterns to identify potential optimizations in terms of one-way roads, traffic light configurations, and public transport schedules.

Type3: Event-Specific Traffic Management

The system will collect information about planned events that attract large crowds (sports events, concerts, fairs) and define event-specific configurations for traffic lights, roads, and public transport schedules.

The SustainCity system will integrate data from existing infrastructure and services:

- A sensor network that measures traffic flow at intersections, publishing data on a message bus
- A microservice with public transport schedule information
- A news channel that provides information about planned events in the city

The system will automatically implement Type1 actions and provide suggestions to urban area managers for Type2 and Type3 actions. Additionally, it will log all Type1 actions and publish reports for citizens, including:

- Daily reports on average traffic flow and Type1 actions taken
- Yearly reports on Type2 and Type3 suggestions, both accepted and not accepted by urban area managers

2. Requirement Analysis

2.1 Relevant Human and Non-Human Actors

Human Actors:

- **Citizens**: End users who benefit from improved traffic flow and can access public reports generated by the system.
- **Urban Area Managers**: Key stakeholders who review and approve or reject Type2 and Type3 action suggestions.
- **System Administrators**: Technical personnel responsible for maintaining and monitoring the system.

Non-Human Actors:

- **Traffic Sensors**: Devices that measure and provide data about the time vehicles need to cross intersections.
- **Message Bus**: The communication infrastructure that facilitates the exchange of information between system components.
- **Public Transport Microservice**: External service providing information about public transport schedules.
- **News Channel**: External source of information about planned events in the city.
- **Traffic Light Control System**: The hardware and software components that control traffic lights.

2.2 Use Cases

The primary use cases for the SustainCity system are illustrated in the following use case diagram:

[Use Case Diagram would be here in final document]

Detailed Description of Critical Use Case: Dynamic Traffic Light Adjustment

Use Case Name: Dynamic Traffic Light Adjustment

Actors: Traffic Sensors, Traffic Light Control System, System Administrator

Description: This use case describes how the system automatically adjusts traffic light durations based on real-time traffic conditions.

Preconditions:

1. Traffic sensors are operational and continuously publishing data to the message bus.
2. The system has access to current traffic light configurations.
3. The system has established connection to the traffic light control system.

Main Flow:

1. Traffic sensors continuously monitor and publish intersection crossing times to the message bus.
2. The system subscribes to and processes the sensor data from the message bus.
3. The system identifies significant disparities in traffic flow between intersecting roads.
4. The system calculates optimal traffic light durations for affected intersections.
5. The system sends commands to the traffic light control system to adjust light durations.
6. The traffic light control system implements the adjustments.
7. The system logs the Type1 action taken, including the affected intersections, the adjusted durations, and the timestamp.

Alternative Flows:

- If communication with the traffic light control system fails, the system logs the failure and alerts the system administrator.
- If abnormal sensor data is detected, the system ignores the potentially faulty data and relies on data from other sensors or historical patterns.

Postconditions:

1. Traffic light durations are adjusted to optimize traffic flow.
2. The Type1 action is logged in the system database.
3. The action will be included in the daily report.

2.3 Domain Assumptions

1. **Sensor Reliability**: The traffic sensors are sufficiently reliable and provide accurate data about intersection crossing times.
2. **Data Timeliness**: The data provided by traffic sensors is near real-time with minimal delay.
3. **Control Authority**: The system has the authority to directly modify traffic light durations without human intervention for Type1 actions.
4. **Data Coverage**: Sensors provide adequate coverage of all major intersections in the urban area.
5. **Public Transport Data Accuracy**: The microservice providing public transport schedules contains up-to-date and accurate information.
6. **News Channel Completeness**: The news channel provides comprehensive information about all significant events that might impact traffic patterns.
7. **Urban Manager Engagement**: Urban area managers regularly review and make decisions on Type2 and Type3 action suggestions.
8. **Traffic Light System Compatibility**: The traffic light control systems throughout the city can be remotely configured by the SustainCity system.
9. **Historical Data Availability**: For pattern analysis, historical traffic data is available or can be collected over time.
10. **Legal Compliance**: The system operates within legal frameworks governing traffic management.

2.4 Requirements

2.4.1 Functional Requirements

FR1: Traffic Data Collection and Processing

- FR1.1: The system shall continuously collect traffic data from sensors via the message bus.
- FR1.2: The system shall process and aggregate traffic data to identify traffic flow patterns.
- FR1.3: The system shall store historical traffic data for analysis and reporting purposes.

FR2: Dynamic Traffic Light Management (Type1)

- FR2.1: The system shall automatically detect significant traffic flow disparities between intersecting roads.
- FR2.2: The system shall calculate optimal traffic light durations based on current traffic conditions.
- FR2.3: The system shall send adjustment commands to the traffic light control system.
- FR2.4: The system shall log all Type1 actions, including affected intersections, adjusted durations, and timestamps.

FR3: Traffic Pattern Analysis (Type2)

- FR3.1: The system shall analyze historical traffic data to identify recurring patterns.
- FR3.2: The system shall generate optimization suggestions for one-way road configurations.
- FR3.3: The system shall generate optimization suggestions for traffic light configurations.
- FR3.4: The system shall analyze public transport schedules in relation to traffic patterns.
- FR3.5: The system shall generate optimization suggestions for public transport schedules.

FR4: Event-Specific Traffic Management (Type3)

- FR4.1: The system shall monitor the news channel for information about planned events.
- FR4.2: The system shall assess the potential traffic impact of planned events.
- FR4.3: The system shall generate event-specific configurations for traffic lights and road usage.
- FR4.4: The system shall generate event-specific suggestions for public transport adjustments.

FR5: Suggestion Management

- FR5.1: The system shall present Type2 and Type3 suggestions to urban area managers for review.
- FR5.2: The system shall record the acceptance or rejection of suggestions by urban area managers.
- FR5.3: The system shall maintain a history of all suggestions and their outcomes.

FR6: Reporting

- FR6.1: The system shall generate daily reports on average traffic flow and Type1 actions.
- FR6.2: The system shall generate yearly reports on Type2 and Type3 suggestions and their outcomes.
- FR6.3: The system shall publish reports for public access.

2.4.2 Non-Functional Requirements

NFR1: Performance

- NFR1.1: The system shall process sensor data within 5 seconds of receipt.
- NFR1.2: The system shall implement Type1 traffic light adjustments within 30 seconds of detecting significant traffic flow disparities.
- NFR1.3: The system shall generate daily reports within 1 hour of the end of each day.

NFR2: Reliability

- NFR2.1: The system shall maintain 99.9% uptime for critical components related to Type1 actions.
- NFR2.2: The system shall implement fault tolerance mechanisms to handle sensor failures.
- NFR2.3: The system shall maintain data integrity during communication with external systems.

NFR3: Scalability

- NFR3.1: The system shall handle data from at least 1,000 traffic sensors simultaneously.
- NFR3.2: The system shall support at least 100 concurrent users accessing reports.
- NFR3.3: The system shall be extendable to incorporate additional data sources in the future.

NFR4: Security

- NFR4.1: The system shall implement role-based access control for different user types.
- NFR4.2: The system shall encrypt all communications with the traffic light control system.
- NFR4.3: The system shall maintain an audit trail of all administrative actions.

NFR5: Usability

- NFR5.1: The system shall provide intuitive interfaces for urban area managers to review and respond to suggestions.
- NFR5.2: The system shall present reports in a clear, easily understandable format for citizens.
- NFR5.3: The system shall provide visual representations of traffic patterns and adjustments.

3. Design

3.1 General Description of the Architecture

The SustainCity system follows a microservices architecture pattern, combining event-driven components with RESTful services. The architecture consists of the following major components:

[Component Diagram would be here in final document]

Components:

1. **Data Acquisition Service**
   - Subscribes to the message bus to collect traffic sensor data
   - Processes and normalizes raw sensor data
   - Stores processed data in the Traffic Database
2. **Traffic Analysis Engine**
   - Analyzes current and historical traffic data
   - Identifies traffic flow disparities and patterns
   - Triggers Type1 actions when necessary
   - Generates insights for Type2 and Type3 suggestions
3. **Action Management Service**
   - Implements Type1 actions by communicating with the Traffic Light Control System
   - Formulates Type2 and Type3 suggestions
   - Presents suggestions to urban area managers
   - Tracks the status and outcomes of all suggestions
4. **Event Monitoring Service**
   - Monitors the news channel for information about planned events
   - Extracts relevant event details (location, time, expected attendance)
   - Forwards event information to the Traffic Analysis Engine
5. **Public Transport Integration Service**
   - Interacts with the external Public Transport Microservice
   - Retrieves and caches public transport schedules
   - Provides schedule information to the Traffic Analysis Engine
6. **Reporting Service**
   - Generates daily and yearly reports
   - Formats reports for public consumption
   - Publishes reports through the Public Portal
7. **Public Portal**
   - Provides a web interface for citizens to access reports
   - Visualizes traffic data and actions taken

8. **Administration Interface**
    - Provides an interface for urban area managers to review and respond to suggestions
    - Allows system administrators to monitor and configure the system
9. **Traffic Database**
    - Stores historical traffic data
    - Maintains records of all actions and suggestions
    - Stores report data

3.2 Sequence Diagrams

Sequence Diagram 1: Dynamic Traffic Light Adjustment (Type1)

[Sequence Diagram 1 would be here in final document]

Description:
This diagram illustrates the process of dynamically adjusting traffic light durations based on real-time traffic conditions. The process begins with traffic sensors publishing data to the message bus. The Data Acquisition Service processes this data and stores it in the Traffic Database. The Traffic Analysis Engine detects a significant traffic flow disparity and determines optimal light durations. The Action Management Service then communicates with the Traffic Light Control System to implement the adjustments and logs the action. Finally, the action information is stored for inclusion in daily reports.

Sequence Diagram 2: Traffic Pattern Analysis and Suggestion Generation (Type2)

[Sequence Diagram 2 would be here in final document]

Description:
This diagram shows how the system analyzes traffic patterns and generates optimization suggestions. The Traffic Analysis Engine retrieves historical data from the Traffic Database and analyzes it to identify patterns. The Public Transport Integration Service provides schedule information from the external microservice. The Analysis Engine generates optimization suggestions, which the Action Management Service formats and presents to urban area managers through the Administration Interface. The managers' responses are recorded, and the suggestion outcomes are stored for reporting.

Sequence Diagram 3: Event-Specific Configuration Planning (Type3)

[Sequence Diagram 3 would be here in final document]

Description:
This diagram illustrates how the system handles planned events that may impact traffic. The Event Monitoring Service detects event information from the news channel and forwards it to the Traffic Analysis Engine. The Analysis Engine assesses the potential traffic impact using historical data and generates event-specific configuration suggestions. These suggestions are presented to urban area managers, who can approve or reject them. Approved suggestions may result in scheduled Type1 actions for the event day.

3.3 Critical Points and Design Decisions

1. Real-time Processing

- **Challenge**: The system must process large volumes of sensor data in near real-time to make timely traffic light adjustments.
- **Solution**: We've adopted an event-driven architecture with a message bus to handle high-throughput data processing. The Data Acquisition Service uses stream processing techniques to efficiently handle the continuous flow of sensor data.

2. System Reliability

- **Challenge**: Traffic management is a critical urban function, and system failures could lead to traffic chaos.
- **Solution**: The architecture incorporates redundancy and failover mechanisms for critical components. If the automatic adjustment system fails, the Traffic Light Control System defaults to pre-programmed schedules. Additionally, the system includes monitoring and alerting to quickly notify administrators of issues.

3. Data Quality and Sensor Failures

- **Challenge**: Sensor data might be inconsistent, incomplete, or unavailable due to sensor failures.
- **Solution**: The Traffic Analysis Engine implements data validation and anomaly detection algorithms to identify potentially faulty sensor data. When suspicious data is detected, the system relies on data from nearby sensors and historical patterns to maintain operation.

4. Integration with Legacy Systems

- **Challenge**: Traffic light control systems may vary across the city, with different interfaces and capabilities.
- **Solution**: The Action Management Service implements an adapter pattern with specific adapters for different traffic light control systems. This abstraction layer allows the core system to interact uniformly with diverse control systems.

5. Scalability

- **Challenge**: As the city grows, the system must handle an increasing number of sensors, intersections, and users.
- **Solution**: The microservices architecture allows each component to scale independently based on its specific load. The Traffic Database uses a sharding strategy to distribute data across multiple servers, optimizing for both write-heavy operations (sensor data collection) and read-heavy operations (pattern analysis).

6. Suggestion Quality

- **Challenge**: Urban area managers rely on the quality of system suggestions for decision-making.
- **Solution**: The Traffic Analysis Engine employs machine learning models trained on historical data to generate high-quality suggestions. These models continuously improve as more data is collected and as managers provide feedback on suggestions.

7. Privacy and Compliance

- **Challenge**: Traffic management systems may inadvertently collect data that could be used to track individual movement patterns.
- **Solution**: The system is designed to work with aggregated data rather than individual vehicle tracking. All data processing complies with relevant data protection regulations, and the system only retains personally identifiable information when explicitly necessary for its operation.