Traffic Simulator

Revised Requirements Definition Document

Khalid AlHokail
Luke Bay
James Grady
Michael Murphy

Version 2.0
03/07/2007

Table of Contents

1. The Core Idea

The core idea is a model and simulation to study and improve traffic flow. The purpose of the traffic simulator is to avoid possible causes of traffic congestion. Using the simulator, a traffic engineer can study how traffic behaves in various environments by observing a visual representation of the traffic on a 2D map. The simulation is built upon a graph model of nodes and edges where the nodes represent intersections and destinations and the edges represent the roads. The model is visually represented by a 2D map which displays the state of the model as it progresses in time.

The simulation will allow the traffic engineer to study traffic flow by changing variables within the model. She can observe the effects of environmental alterations like adding or removing roads, redistributing populations, changing the timing of traffic lights, or altering the travel patterns of a population. In addition to a broad overview of the area, the engineer will be able to zoom in on a specific intersection. The zoomed view will center on an intersection and display the adjacent connecting intersections and their roadways.

2. Overview of How the Traffic Simulator Fits with Other Projects

The map model is similar to the Building Security Team's blueprints. Their building's blueprints and our graph of the Traffic Simulation are both based on the node and edge model. In both models movement is restricted to edges between nodes. For example, a node in the traffic simulation is an intersection or building, and in the building security simulation nodes are intersecting hallways or rooms. Objects traveling within the model must be on a node or edge at all times. The map model is essential but not dependent on the Building Security Team's model.

The driving model is similar to the Island Team's animal migration patterns. Both models have population distributions and populations with predictable behavior. The populations will migrate at predictable times because of feeding patterns or travel destinations. The weather is another commonality. Heavy snow will impede the speed of animals and automobile alike. The driving model is essential, but not dependent on the Building Security Team's model.

The traffic light control model is similar to the sensor control in the Building Security project. In both simulations the movement of traveling objects, either people or cars, can be regulated by signals in the environment. Specifically, cars obey traffic lights and robbers avoid security cameras. The traffic light control model is not dependent on the Security Team's model and is essential.

The decision model is similar to the guard behavior in the Building Security project. The cars and robbers will try to find the best possible route to their destinations. But this best route does not necessarily mean the shortest route, for the cars will try to avoid traffic jams or frequent traffic lights, while the robbers will try to avoid detection. The decision model is not dependent on the Security Team's model and is essential.

The edge of the map behavior is similar to the floor transitions of the Security team. Guards and robbers must be able to enter or leave a floor of a building through specific points of entry or egress much like vehicles can appear on a road by entering or exiting the roadways bounding the edges of the map. The map model is not dependent on the Security Team's model and is essential.

The hall to office behavior of the Building Security team is similar to our model's intersection behavior. In both simulations it is necessary to keep objects from colliding or occupying the same space. Whether one considers cars on a road or people in a hallway, the area with highest probability for a collision is an intersection. To avoid such collisions, rules for right-of-way are established for intersections. Again, the decision model is not dependent on the Security Team's model and is essential.

3. Requirements for Essential Features of Our Project

3.1. Map Model

3.1.1. Graph

3.1.1.1. Functional

- All the nodes must be reachable.

3.1.1.2. Non Functional

- The graph must be a connected graph that can be traversed from any node to any node.
- The properties of outgoing edges (that represents roads) and nodes (that represents intersections) must be readable, at each node, and must be assigned valid values, and cannot be assigned invalid values.
- The graph shall be a directed graph.

3.1.2. Rules for Intersections

3.1.2.1. Functional

- The intersection must include a configuration of allowed turns (intersections, roundabouts, dead ends, one way streets, and highway onramps).
- The intersection must only allow legal turns.

3.1.2.2. Non Functional

- There are no non-functional requirements.

3.1.3. Number of Lanes

3.1.3.1. Functional

- The lane capacity shall be represented by an integer greater than or equal to 1.
- Each lane shall be divided into car-length segments.

3.1.3.2. Non functional

- There are no non-functional requirements.

3.1.4. Status of Lanes and Intersections

3.1.4.1. Functional

- The number of drivable lanes shall be represented by an integer greater than or equal to 0, and shall never exceed the maximum allowed capacity.

3.1.4.2. Non Functional

- The number of drivable lanes shall be updated in real time when a lane is closed or opened.

3.1.5. 2D Map - Visualize Flow (over time)

3.1.5.1. Functional

- The 2D map must display roadways, intersections, and traffic distributions (the number of cars at a specific position and time) or individual cars.

3.1.5.2. Non Functional

- The map must provide real-time updates (the simulation display keeps up with actual time).

3.2.Driving Model

3.2.1. Time of Day

3.2.1.1. Functional

- The time of day shall be represented.

3.2.1.2. Non Functional

- The update time shall be done according to simulation time.
- The time of day shall be displayed to the user.

3.2.2. Population Distributions

3.2.2.1. Functional

- There are no functional requirements.

3.2.2.2. Non Functional

- Static population distributions shall be provided.

3.2.3. Traffic generator

3.2.3.1. Functional

- The traffic generator shall populate the map with vehicles.

3.2.3.2. Non Functional

- There are no non-functional requirements.

3.3. Traffic Light Control

3.3.1. Rules for one concrete location

3.3.1.1. Functional

- The rules of the location shall dictate the order in which the lights will cycle.

3.3.1.2. Non functional

- There are no non-functional requirements.

3.3.2. Traffic light state and timing

3.3.2.1. Functional

- The lights shall be represented by enumerations and guide the drivers at intersections.
- Each traffic light shall have a separate timing.

3.3.2.2. Non Functional

- The current status of the light shall be displayed.

3.4. Decision Model

3.4.1. One concrete strategy for vehicle speeds and speed limits

3.4.1.1. Functional

- The drivers shall follow edges, road signs and speed limits.

3.4.1.2. Non Functional

- There are no non-functional requirements.

3.4.2. One concrete strategy for vehicle status and capability

3.4.2.1. Functional

- The number of traffic incidents shall be determined by a probability for each vehicle.
- When a traffic incident occurs, the map model is updated with the proper road segment or intersection closures.

3.4.2.2. Non Functional

- A driver will try to drive around a blocked road segment.
- The vehicle must never break down or run out of fuel.

- The status of the vehicle shall be displayed (start node, current node, next node, end node, edge position, speed).
- Traffic incidents shall be represented on the 2D map.

3.4.3. Driving speeds for individual cars

3.4.3.1. Functional

- Car speeds must be updated according to the road conditions.

3.4.3.2. Non Functional

- There are no non-functional requirements.

3.4.4. Route

3.4.4.1. Functional

- For each driver a list of waypoint nodes shall be generated. This list will detail a path that the driver will follow from his source to his destination.

3.4.4.2. Non Functional

- There are no non-functional requirements.

4. Requirements for Peripheral Features of Our Project

4.1.Map Model

- A first person driving experience may be added.
- Zoom view
    1. The 2D map shall allow intersections to be selected so that the map is rescaled to show a close up view of the intersection selected, the adjacent intersections, and the roadways connecting them.
    2. Selecting an adjacent intersection in the zoomed view shall center on the new intersection selected.
    3. If the centered intersection is selected the view shall zoom out to the world map.
    4. The map must provide real-time updates (simulation display keeps up with actual time).
    5. With each real time update, for every object with a changed attribute, if that attribute is visually represented (i.e. position, color, creation, or elimination) the refreshed screen will display this change.

4.2.Driving Model

- The list of waypoint nodes that define the entire path a driver will follow will be the shortest path.

4.3. Traffic Light Control

- Other traffic light rules (us/non-us) will be considered.
- A synchronization protocol for traffic lights will be considered.
- "Smart" sensing/adapting traffic lights may be added.
- A protocol for automatically choosing traffic light cycles will be considered.

4.4. Decision Model

- The driver's experience level may be taken into account.
- Driver's status may eventually affect a driver's ability to control the vehicle.
- Additional driving strategies may be added.
- Additional vehicle status and capability strategies may be added.

5. Requirement of Features Needed from Other Teams

The following are all essential features needed from the infrastructure team:

5.1. Geometry & Topology

The geometry and topology model provides the modeling for the map, by representing intersections as nodes and roads as edges.

5.2. Entity Hierarchy

The entity hierarchy provides the movable objects in the simulation, it specifically provides the cars needed for the modeling.

5.3. Kernel

The kernel builds the environment that the object from the Entity Hierarchy will interact with.

5.4. Data

The data is information such as traffic flow.

5.5. Encounters

Encounters provide the behavior of the objects in the model.

5.6. Motion Protocol

Motion Protocol will control the motion of the entities in the model.

6. External Interfaces

6.1. User interface

Our system will have a user interface that he interacts with. This interface will display the 2D map along with all the roadways and traffic. The interface will also display the status of each roadway whether it is congested or not. Furthermore, the time of day will be displayed as part of a legend to indicate the simulation time. The legend will also include a color key for the status of roadways. This color can tell us how many cars are traveling on that roadway and whether there is a blockage or an accident or not.

By clicking on a roadway, an information box will be displayed that shows statistics about traffic going on that part of the roadway. This box will include:

1. Number of cars in the clicked roadway.
2. Current average speed of the roadway.
3. Speed limit of the roadway.
4. Number of accidents.
5. Number of lanes.
6. Number of drivable lanes.

On the other side, when we click on a car, the interface will also display an information box that contains:

1. Current Node.
2. Next Node.
3. Start Node.
4. Destination Node.
5. Current Speed.
6. Expected Arrival Time.
7. Status of the car (Moving/Stopped/In an accident).

When a user clicks on an intersection, an information box that contains the current status of all the lights on that intersection (Red/Yellow/Green/Flashing Red/Flashing Yellow/Off) will be displayed. Population distribution statistics will be displayed when a user clicks on a section on the map. The information box shows how many people are living in that section of the map.

6.2.Framework Interface

The framework interface consists of the following four parts:

1. 2D map

This will provide the map model the ability to display and draw the map along with all the streets and intersections.

2. Event Handler

This part will provide the map model the ability to trigger an event when a construction or an accident occurs or clears so that a lane will be closed or opened.

3. Colors

This part will provide services to the map model so that when the map is refreshed; all the colors are refreshed also. Colors also provide services to the traffic light control so that all traffic lights can be displayed with their status. Furthermore, colors enable the decision model to display road segment closures in the event of an accident.

4. Sprites

Sprites will enable the map model to display all the sprites on the map when the map is refreshed. Sprites also enable the traffic light control to display the traffic lights on the intersections.

6.3.Platform Interface

The platform interface consists of the following three parts:

1. Detect Object Collisions

This part will provide the decision model the ability to detect a collision between two cars. If a collision is detected then it is considered an accident.

2. Trigger Object Specific Behavior

This part will provide the decision model the ability to control the car’s behavior such as where to go and how to speed up or slow down.

3. Time Passing

Both the traffic light control and the driving model need to observe time passing. The traffic light control needs time control the order each light cycle. And for the driving model, it needs to keep track of the time of day to calculate the simulation time and to display the time of the day to the user.

7. Glossary and Definitions

Additional driving strategies

Additional driving strategies define whether a driver is an aggressive driver, a non-aggressive driver, a driver who tends to speed, or a driver who tends not to speed.

Driver's experience level

A driver’s experience level defines how expert that driver is. Each driver is classified as beginner, intermediate, or advanced.

Edge

An Edge is part of the graph that represents a roadway, street, highway or basically anything that a car uses to move from a point to another.

Graph

A graph is a kind of a data structure that (in our project) represents the map to the computer. A graph consists of a set of nodes and a set of edges.

Lane Capacity

Lane capacity is the number of lanes that a road consists of.

Light Cycles

The way the traffic light changes its state. For example, some countries use the following cycle (Red, Yellow, Green, Red) and some other countries use (Red, Green, Yellow, Red).

Light Enumerations

A light will have a value that defines its state. The values can be one of the following: {red, yellow, green, flashing red, flashing yellow, flashing green, right red, right yellow, right green, left red, left yellow, left green}.

Node

A node is part of the graph that represents an intersection, roundabout or basically any point that connects two or more edges.

Speed limit

No car may exceed the speed limit that is stored with the edge.

"Smart" sensing/adapting traffic lights

Consider the option of installing sensors on traffic lights that sense if a car is waiting and turn green instead of waiting for all other traffic lights to cycle normally.

Another sensing option adapts to sparse traffic by changing the normal cycling of a traffic light into one that is flashing red or flashing yellow.

Time of Day

The time of day the simulation is simulating.