

Traffic Simulation

Document:	System Specification
Version:	2.0
Date:	20 November 2022
Author:	John Businge
Status:	Delivered

1 Summary

This document contains the specification for a computer system that can execute a traffic simulation. It has been written for the course “472 - Software Product Design and Development I” (Bachelor of computer science - UNLV).

2 Context

On 25 June 2018, the political steering committee of Las Vegas selected 18 projects that will cover the Las Vegas Strip roads. The first works started in the summer of 2019 and have caused a lot of traffic disruption. In order to limit this nuisance as much as possible during the further course of the project, the Department of Public Works has opted to develop a simulation model that can simulate the traffic.

Department of Public Works has asked UNLV to develop this simulation. The computer science bachelor students will be tasked to work on this project in the “Software Product Design and Development I” and “Computer Graphics” courses.

3 Legend

The requirements specification has been drafted on the basis of use-cases. Each use-case describes a small part of the desired functionality. The intention is that during each phase of the project different use-cases are implemented. A typical use-case contains the following components:

- **Reference number & title:**
Used to identify or refer to a use-case.
- **Priority:**
The specification of the system demands more than what can be delivered within the foreseen time. That is why we use the priority of a use-case to indicate to what extent its functionality is important. The priority can be (in order of importance): REQUIRED (this use case must be completed), IMPORTANT (not essential but preferably deliver), USEFUL (interesting but can be omitted).
- **Goal:**
Brief description of the goal of the use case, i.e. what the use case contributes to the entire functionality.
- **Precondition:**
Brief description of the required properties at the start of the use-case.
- **Postcondition:**
Brief description of the required properties at the end of the use-case.
- **Steps:**
A sequential description of how the use-case executes if everything goes well (the so-called "happy day scenario"). The steps are numbered and may include control instructions (WHILE, IF, ...).
- **Exceptions:**
A list of possible deviations from the happy day scenario and how they should be treated. An exception (a) refers to the number of the step where the exception may occur, (b) contains a condition that indicates when the exception occurs, and (c) describes very briefly how the exception will be treated.
- **Example:**
An example of the input or output.

Sometimes a use-case is an extension of another use-case. Then the following components are relevant:

- **Extension:**

A reference to the use-case that is being extended.

- **Steps:**

A list of additional and / or modified steps with regard to the use-case that is being extended.

An extension (a) refers to the step number being extended, (b) states whether the extension is before, after, or during the step, and (c) describes what exactly will happen in the extension.

4 Overzicht

Use-Case	Priority
<i>1: Input</i>	
1.1 Read traffic network	REQUIRED
1.2 Read in vehicle generator	IMPORTANT
1.3 Read in vehicle with type	REQUIRED
1.4 Read bus stops	IMPORTANT
1.5 Read intersections	IMPORTANT
<i>2: Output</i>	
2.2 Simple ouput	REQUIRED
2.2 Graphical representation	IMPORTANT
2.3 Integration with Graphics	IMPORTANT
<i>3: Simulation</i>	
3.1 Moving vehicle	REQUIRED
3.2 Traffic light simulation	REQUIRED
3.3 Automatic simulation	REQUIRED
3.4 Simulation with vehicle generator	IMPORTANT
3.5 Driving vehicles with type	IMPORTANT
3.6 Simulation of bus stops	IMPORTANT
3.7 Simulation of intersections	IMPORTANT
3.8 Simulation of intersections with traffic light	USEFUL
3.9 Smart traffic lights	USEFUL
3.10 Simulation of orange traffic light	USEFUL
<i>4: Userinterface</i>	
4.1 GUI for simulation	USEFUL
4.2 GUI for traffic lights	USEFUL
4.3 GUI for vehicle generator	USEFUL

1.1. Read traffic network

Priority:

REQUIRED

Goal:

Read of the schedule of the traffic network: the different roads and the different vehicles.

Precondition:

An ASCII file containing a description of the roads and vehicles. (See Appendix 4 for more information about the XML format)

Post condition:

The system contains a schedule with the different roads, and information about all vehicles.

Steps:

1. Open input file
2. WHILE Not at end of file
 - 2.1. Detect the type of element (VEHICLE, ROAD, TRAFFIC LIGHT)
 - 2.2. Read data of the element
 - 2.3. IF Verify data is valid
 - 2.3.1. THEN Add the element to the virtual traffic network
 - 2.3.1. ELSE Error message + go to next element in the file
3. Verify consistency of the traffic network
4. Close input file

Exceptions:

- 2.1. [Unrecognized element] Error message + go to next element in the file \Rightarrow continue from step 2
- 2.2. [Invalid data] Error message + go to next element in the file \Rightarrow continue from step 2
3. [Inconsistent subway network] Error message \Rightarrow continue from step 4

Example:

A road with two cars and one traffic light:

```
<ROAD>
  <name>Rochelle</name>
  <length>500</length>
</ROAD>
<TRAFFIC LIGHT>
  <road>Rochelle</road>
  <position>400</position>
  <cycle>30</cycle>
</TRAFFIC LIGHT>
<VEHICLE>
  <road>Rochelle</road>
  <position>20</position>
</VEHICLE>
<VEHICLE>
  <road>Rochelle</road>
  <position>0</position>
</VEHICLE>
```

1.2. Read in vehicle generator

Priority:

IMPORTANT

Goal:

Read in how vehicles should be generated.

Precondition:

An ASCII file containing a description of a vehicle generator. (See Appendix A for more information about the XML format)

Post condition:

The system includes a schema of a vehicle generator .

Extension:

Use Case 1.1

Steps:

[2.1, during] Take into account extra element VEHICLE GENERATOR

Exceptions:

None

Example:

```
<VEHICLEGENERATOR>  
  <name>Rochelle</name>  
  <frequency>5</frequency>  
</VEHICLEGENERATOR>
```

1.3. Read in vehicle with type

Priority:
REQUIRED

Goal:

A specific vehicle will behave differently depending on its type. To be able to include this in the simulation, this data must also be read. See Appendix C for more information on the different types of vehicles.

Extension:

Use Case 1.1 and Use Case 1.2

Steps:

[2, during] Take into account extra possibilities for 'type'

Comments:

If you implemented Use Case 1.2, then each vehicle generator will also contain a 'type' attribute. This determines what type of vehicles will be generated.

Example:

Example of vehicle and vehicle generator with different types (car, bus, fire engine, ambulance, and police van).

```
<VEHICLEGENERATOR>
  <name>Rochelle</name>
  <frequency>5</frequency>
  <type>auto</type>
</VEHICLEGENERATOR>
<VEHICLE>
  <road>Rochelle</road>
  <position>20</position>
  <type>bus</type>
</VEHICLE>
<VEHICLE>
  <road>Rochelle</road>
  <position>40</position>
  <type>fire truck</type>
</VEHICLE>
<VEHICLE>
  <road>Rochelle</road>
  <position>60</position>
  <type>ambulance</type>
</VEHICLE>
<VEHICLE>
  <road>Rochelle</road>
  <position>80</position>
  <type>police van</type>
</VEHICLE>
```


1.4 Read bus stops

Priority:

IMPORTANT

Goal:

A road can contain bus stops. To be able to include this in the simulation, this data must also be read.

Extension:

Use Case 1.1

Steps:

[2, during] Take extra element into account

Exceptions:

None **Example:**

Example of a bus stop

```
<ROAD>
  <name>Rochelle</name>
  <length>500</length>
</ROAD>
<BUSSTOP>
  <name>Rochelle</name>
  <position>250</position>
  <waitingtime>20</waitingtime>
</BUSSTOP>
```

1.5 Read intersections

Priority:

IMPORTANT

Goal:

Roads can intersect at an intersection. To be able to include this in the simulation, this data must also be read.

Extension:

Use Case 1.1

Steps:

[2, during] Take extra element into account

Exceptions:

None **Example:**

Example of intersections.

```
<ROAD>
  <name>Rochelle</name>
  <length>500</length>
</ROAD>
<ROAD>
  <name>Maryland</name>
  <length>750</length>
</ROAD>
<ROAD>
  <name>Tamarus</name>
  <length>1000</length>
</ROAD>
<CROSSROADS>
  <road position="500">Rochelle</road>
  <road position="500">Maryland</road>
</CROSSROADS>
```

2.1. Simple output

Priority:
REQUIRED

Goal:
Output all data from the virtual traffic system.

Precondition:
The system contains a diagram of virtual traffic network.

Postcondition:
The system generated an ASCII file that contains all data from the virtual traffic network.

Steps:

1. Write out current simulation time
2. WHILE Vehicles still in simulation
 - 2.1. Write out vehicle data

Exceptions:
None

Example:
Given the input from 1.1

```
Time 0
Vehicle 1
  -> road: Rochelle
  -> position: 20
  -> speed: 16.6
```

```
Vehicle 2
  -> road: Rochelle
  -> position: 0
  -> speed: 16.6
```

2.2. Grafical representation

Priority:

IMPORTANT

Goal:

The state of the traffic system is displayed graphically.

Precondition:

The system has been initialized correctly.

Postcondition:

The system has generated a text file (ASCII) in which the state of the traffic system is described.

Variant:

The system has output a text file (HTML) describing the state of the traffic situation.

Steps:

1. Open the outputfile
2. Draw the current state of the traffic network
3. Close the outputfile

Exceptions:

None

Example:

If a road with two traffic lights, two bus stops, two cars and a bus.

```
Rochelle      | =====A=====B=====A=
> traffic lights |           |           G           |           R
> bus stops     |           B           B
```

2.3 Integration with Graphics

Priority:

IMPORTANT

Goal:

To convince the city authorities of Las Vegas, our client would like to have a 3D visualization of the simulation. For this you can use a graphics engine of choice.

Precondition:

The system has been initialized correctly.

Postcondition:

Every movement of vehicles is displayed in a 3D environment.

3.1. Moving vehicle

Priority:

REQUIRED

Goal:

Simulate driving a vehicle. See Appendix B for more information about the formulas.

Precondition:

The system contains a diagram of the virtual traffic network. There is a vehicle on a road.

Postcondition:

The position, speed, and acceleration of the vehicle have been recalculated.

Steps:

1. Calculate new speed and position of vehicle
2. Calculate new acceleration of vehicle
3. IF new position is outside current road
 - 3.1. Remove vehicle from simulation

Exceptions:

None

3.2. Traffic light simulation

Priority:

REQUIRED

Goal:

Simulating traffic lights

Precondition:

The system contains a diagram of the virtual road network. There is a traffic light on a road.

Postcondition:

Vehicles adapt depending on the state of the traffic light.

Steps:

1. IF time since last change > cycle
 - 1.1 THEN change the color of the light (green \iff red)
2. IF traffic light is green
 - 2.1 THEN vehicles in front of the traffic light may accelerate back up
- 3.1 IF traffic light is red
 - 3.1.1 THEN IF the first vehicle in front of the light is in the deceleration distance
 - 3.1.1.1 THEN apply the deceleration factor to the vehicle
 - 3.1.1.2 ELSE IF the first vehicle in front of the light is in the first half of the stopping distance
 - 3.1.1.2.1 THEN stop the vehicle

3.3. Automatic simulation

Priority:

REQUIRED

Goal:

Run simulation automatically.

Precondition:

The system contains a diagram of the virtual road network.

Postcondition:

The traffic in the road network is simulated.

Steps:

1. FOR any vehicle in the road network
 - 1.1 Execute use-case 3.1 out on the vehicle
2. FOR any traffic light in the road network
 - 2.1 Execute use-case 3.2 out at the traffic light

3.4. Simulation with vehicle generator

Priority:

IMPORTANT

Goal:

Simulation with vehicle generator.

Precondition:

The system contains a diagram of the virtual road network. There is a vehicle generator on a road.

Postcondition:

Vehicles are added automatically during the simulation.

Uitbreiding:

Use Case 3.3

Steps:

3. FOR any vehicle generator
- 3.1 IF time since last vehicle > frequency
- 3.1.1 IF no vehicle on road between positions 0 and $2l$
- 3.1.1.1 THEN add vehicle to road at position 0

3.5. Driving vehicles with type

Priority:
REQUIRED

Goal:
A specific vehicle will behave differently depending on its type. See Appendix C for more information on the different types of vehicles.

Extension:
Use case 3.1

Steps:
[1-2, during] Take into account the values of the vehicle type

Extension:
Use case 3.2

Steps:
[3, during] IF the first vehicle before the traffic light is an emergency vehicle
(3, during) THEN vehicles need not slow down or stop

Exceptions:
None

3.6. Simulation of bus stops

Priority:

IMPORTANT

Goal:

A track can contain bus stops. Bus-type vehicles must stop at each bus stop, then remain there until the waiting time has elapsed. You also need to implement Use case 1.4 for this.

Precondition:

The system contains a diagram of the virtual traffic situation. This schedule contains at least 1 bus stop and 1 bus.

Postcondition:The bus stops at the bus stop.

Steps:

- 1 IF a bus type vehicle is in the deceleration distance
 - 1.1 THEN apply the deceleration factor to the vehicle
- 2 ELSE IF a bus type vehicle is in the stopping distance
 - 2.1 THEN causes the vehicle to stop
- 3 IF time since vehicle with bus type stopped > waiting time
 - 3.1 THEN the vehicle may depart

Notes:

The steps are only applied to the first bus before the bus stop. If there are several buses on the track in front of the stop, the other buses will automatically slow down because the vehicle in front also slows down. Only when the first bus leaves it is necessary to apply the steps to the next bus.

3.7. Simulation of intersections

Priority:

IMPORTANT

Goal:

A road can contain intersections. Vehicles must choose which road to follow at an intersection. You also need to implement use case 1.5 for this.

Precondition:

The system contains a scene of the virtual traffic network. This scheme contains at least 2 roads and 1 intersection.

Postcondition: Vehicles choose a road at each intersection.

Steps:

1 IF a vehicle is approaching an intersection

1.1 THEN choose any road from the intersection to continue on

Notes:

If the position of an intersection on a road is equal to the length of the road, then the road ends at that intersection. Vehicles will therefore not be able to choose this road at this intersection. If the position of an intersection on a road is zero, then the road starts at that intersection.

3.8. Simulation of intersections with traffic light

Priority:

USEFUL

Goal:

Take into account the traffic lights at an intersection to avoid collisions.

Precondition:

The system contains a scheme of the virtual traffic network. This scheme contains at least 2 roads and 1 intersection with a traffic light at the intersection.

Postcondition:

Traffic lights at an intersection take each other into account.

Extension:

Use case 3.7

Note:

If each road of an intersection has a traffic light whose position is the same as that of the intersection, then you will have to take that into account. In this case, choose 1 of the traffic lights to follow its fixed cycle. The other traffic light will always be the opposite color regardless of its cycle.

3.9. Smart traffic lights

Priority:

USEFUL

Goal:

Traffic lights take into account vehicles on the track to improve traffic flow.

Precondition:

The system contains a scheme of the virtual traffic network. This scheme contains at least 1 intersection with traffic lights.

Postcondition:

Traffic lights take vehicles into account.

Extension:

Use case 3.2

Use case 3.8

Notes:

Instead of following a fixed pattern, traffic lights at an intersection will try to give vehicles green as much as possible. You can choose how you implement this algorithm. However, it must adhere to the following conditions:

- The traffic light must retain its color (green or red) for at least 10 seconds before it is allowed to change back.
- The traffic light may retain its color (green or red) for a maximum of 60 seconds before it has to change back.
- Traffic lights at an intersection always have an opposite colour.

3.10. Simulation of orange traffic light

Priority:

USEFUL

Goal:

Be able to simulate traffic lights with three colors.

Precondition:

The system contains a scheme of the virtual traffic network. This scheme contains at least 1 traffic light.

Postcondition:

Traffic lights turn orange before they turn red.

Extension:

Use case 3.2

Notes:

To simulate a three-color traffic lights, use the following rules:

- A traffic light that changes from green to red will first turn orange.
- A traffic light that changes from red to green will not turn orange.
- The time that a traffic light remains orange is equal to 10% of its cycle.
- When a traffic light is orange, vehicles within the stopping distance may continue at maximum speed. The first vehicle in the deceleration distance will begin to decelerate.

Example:

A traffic light with a cycle of 20 seconds follows the following pattern:

- 20 seconds green
- 2 seconds amber
- 20 seconds red
- 20 seconds green

4.1. GUI for simulation

Priority:

USEFUL

Goal:

Have a user interface for controlling the simulation. This includes a pause, play, next step, and previous step button.

Precondition:

The system contains a virtual subway network with the different stations and data on all trams.

Postcondition:

The simulation can be controlled manually using on a graphical user interface.

4.2. GUI for traffic lights

Priority:

USEFUL

Goal:

Have a friendly user interface for managing traffic lights: between simulation steps you can adjust the state of traffic lights.

Precondition:

The system contains a virtual traffic network.

Postcondition:

Traffic lights can be manually adjusted using a graphical user interface.

4.3 GUI for vehicle generator

Priority:

USEFUL

Goal:

Have a friendly user interface for managing vehicle generators: between simulation steps you can have a vehicle generator add vehicles.

Precondition:

The system contains a scheme of the virtual traffic situation.

Postcondition:

Vehicle generators can manually add vehicles using a graphical user interface.

Appendix A - Input format

The input format for the virtual traffic network has been chosen in such a way that new attributes and elements can easily be added.

```
TrafficSystem = { Element }
Element = "<" ElementType ">" AttributeList "</" ElementType ">"
ElementType = "VEHICLE" | "TRAFFIC LIGHT" | "ROAD" | "VEHICLE GENERATOR" |
"BUS STOP" | "INTERSECTION"
AttributeList = Attribute { Attribute }
Attribute = "<" AttributeType ">" AttributeValue "</" AttributeType ">"
AttributeType = "name" | "length" | "road" | "position" | "cycle" | "frequency" |
"type" | "waiting time"
AttributeValue = Primitive
Primitive = Integer | String
Integer = Digit { Digit }
Digit = "0" ... "9"
String = Character { Character }
Character = "a" ... "z" | "A" ... "Z"
```

Note that the AttributeList has a relatively free format which will strongly depend on the type of element defined. The following table shows the attributes for each element:

ElementType	Attribute
ROAD	name, length
TRAFFIC LIGHT	road, position, cycle
VEHICLE	road, position
VEHICLE GENERATOR	road, frequency

In addition, depending on the AttributeType, only one specific AttributeValue is allowed:

AttributeType	AttributeValue
name, road	String
length, position, cycle, frequency	Integer

In addition, the opening tag must always correspond to the closing tag. This is why it is necessary to check whether or not the input is valid during parsing.

The inputfile containing the traffic network is written by hand. In order to simulate the subway system, the information must be consistent.

The traffic system is consistent if:

- Each vehicle is on an existing road.
- Each traffic light is on an existing road.
- Each vehicle generator is on an existing road.
- Each bus stop is on an existing road.
- The position of each vehicle is less than the length of the road.
- The position of each traffic light is less than the length of the road.
- There is a maximum of one vehicle generator on each road.
- The position of each bus stop is less than the length of the road.
- A traffic light must not be within the deceleration distance of another traffic light (see Appendix B).

Comments:

- Length, position, cycle, and frequency must always be positive.
- The name is used to uniquely identify a road.

Appendix B - Simulation model

B.1 - Variables

The table below contains an overview of the variables used in the simulation model.

Abbreviation	Name	Description
l	length	Length of a vehicle.
x	position	Current position of a vehicle, measured from the front bumper.
v	speed	Current speed of a vehicle.
v_{max}	maximum speed	The desired maximum speed of a vehicle. Can change due to external factors (such as a traffic light).
V_{max}	maximum speed	The absolute maximum speed that a vehicle can travel.
a	acceleration	Current acceleration of a vehicle.
a_{max}	maximum acceleration	The maximum acceleration of a vehicle.
b_{max}	maximum braking factor	The maximum braking factor of a vehicle.
f_{min}	minimum following distance	The minimum desired following distance of a vehicle.
Δt	simulation time	The time between two steps of the simulation.
Δx_s	deceleration distance	The distance to a traffic light in which a vehicle must slow down, including the stopping distance.
Δx_{s0}	stopping distance	The distance from a traffic light in which a vehicle must stop.
s	delay factor	Factor by which a vehicle must slow down at a traffic light.

B.2 - Position, speed, and formulas

We distinguish between two networks when adjusting the position and speed of a vehicle:

- If $v + a\Delta t$ is less than zero, the velocity would become negative. This is not allowed in our model. In this case we adjust the position as follows:

$$x = x - \frac{v^2}{2a} \quad (1)$$

Then we set the speed equal to zero

- If this is not the case, we first adjust the speed:

$$v = v + a\Delta t \quad (2)$$

Then we adjust the position:

$$x = x + v\Delta t + a\frac{\Delta t^2}{2} \quad (3)$$

B.3 - Acceleration Formulas

To adjust the acceleration of a vehicle i , we first calculate δ , the interaction term with vehicle $i - 1$ (the vehicle driving in front of vehicle i).

We first calculate the following distance $x\Delta$:

$$\Delta x = x_{i-1} - x_i - l_{i-1} \quad (4)$$

We then calculate the speed difference Δv :

$$\Delta v = v_i - v_{i-1} \quad (5)$$

With this we calculate δ :

$$\delta = \frac{f_{min} + \max(0, v + \frac{v\Delta v}{2\sqrt{a_{max}b_{max}}})}{\Delta x} \quad (6)$$

If there is no vehicle in front, so vehicle i is the first vehicle on the road, then we set δ equal to zero.

We can then calculate the acceleration as follows:

$$a = a_{max}(1 - \left(\frac{v}{v_{max}}\right)^4 - \delta^2) \quad (7)$$

B.4 - Slowing down and accelerating

To slow down the first vehicle on the road, we set v_{max} equal to sV_{max} , with s the deceleration factor. Vehicles behind this vehicle will also automatically slow down.

If the first vehicle on the road is allowed to accelerate back up, then we set v_{max} equal to V_{max} . Vehicles behind this vehicle will automatically accelerate as well.

B.5 Stopping

If the first vehicle on the road has to come to a stop, we adjust the acceleration in each simulation step as follows.

$$a = -\frac{b_{max}v}{v_{max}} \quad (8)$$

Vehicles behind this vehicle will also automatically come to a stop

Appendix C - Default values

The table below contains an overview of the general default values used in the simulation model.

General		
Abbreviation	Name	Value
Δt	simulation time	0.0166
Δx_s	deceleration distance	50
Δx_{s0}	stopping distance	15
s	delay factor	0.4

The table below contains an overview of the default values used for the different types of vehicles: cars and buses.

Car		
Abbreviation	Name	Value
l	length	4
V_{max}	maximum speed	16.6
a_{max}	maximum acceleration	1.44
b_{max}	maximum braking factor	4.61
f_{min}	minimum following distance	4
Bus		
Abbreviation	Name	Value
l	length	12
V_{max}	maximum speed	11.4
a_{max}	maximum acceleration	1.22
b_{max}	maximum braking factor	4.29
f_{min}	minimum following distance	12

The table below contains an overview of the default values used for the different types of priority vehicles: fire truck, ambulance, and police station wagon.

Fire truck		
Abbreviation	Name	Value
l	length	10
V_{max}	maximum speed	14.6
a_{max}	maximum acceleration	1.33
b_{max}	maximum braking factor	4.56
f_{min}	minimum following distance	10
Ambulance		
Abbreviation	Name	Value
l	length	8
V_{max}	maximum speed	15.5
a_{max}	maximum acceleration	1.44
b_{max}	maximum braking factor	4.47
f_{min}	minimum following distance	8
Police van		
Abbreviation	Name	Value
l	length	6
V_{max}	maximum speed	17.2
a_{max}	maximum acceleration	1.55
b_{max}	maximum braking factor	4.92
f_{min}	minimum following distance	6