Eco-Driving Simulator: a Multiplayer Simulation for Training Eco-Friendly Driving

Eurico Doirado **Pedro Cuba** **Arturo Nakasone** **Helmut Prendinger**
National Institute of Informatics, Japan

Motivation

Training and early education play an important role in ensuring that tomorrow's drivers will be familiar with eco-driving concepts. However this training and education might be difficult to provide. It is too expensive to provide such training with real cars and might not even be possible, for example in the case of people without a driver license. On the other hand, existent simulators do not allow for several simultaneous users, depriving users of interaction with other people, like in real life scenarios.

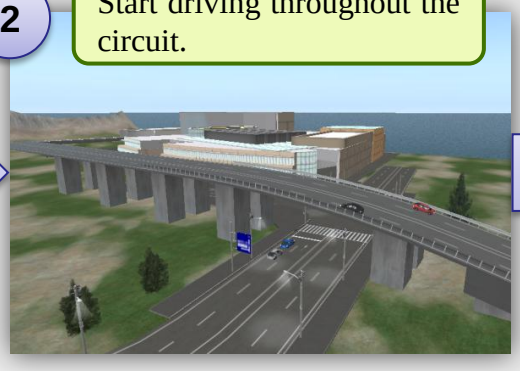
Objective

We present this eco-simulator as an inexpensive and multiuser solution to provide training in eco-driving. This system was designed to create awareness about the eco-driving topic to some extent. In this game you are challenged to care about the environment, taking the player through several real world situations that have a high impact on a car's consumption and consequently on CO2 emission. In the end, the player gets a score according to the level of "eco-friendliness".

Safe Eco-Driving In The Virtual World

1 Pick a car from the display.

2 Start driving throughout the circuit.

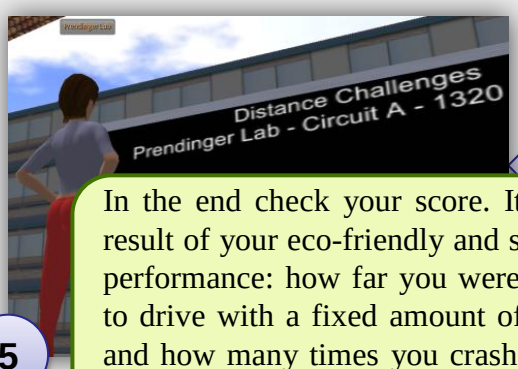


Keep an eye on the eco-indicator and try to be as much eco-friendly as possible while driving.

3

4 Drive safely, don't drive through red lights, avoid crashes and be careful in sharp turns.

5 In the end check your score. It is a result of your eco-friendly and safety performance: how far you were able to drive with a fixed amount of fuel and how many times you crashed or went through red traffic lights.

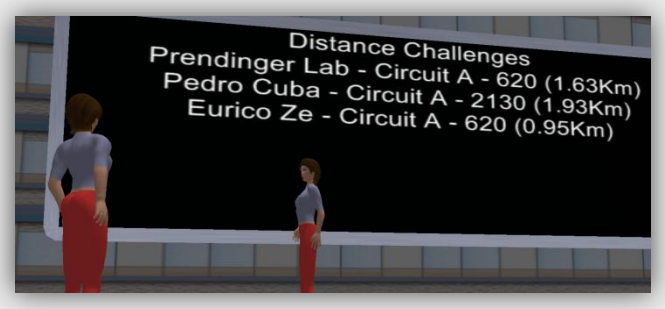


Having Fun With Your Friends

Meet your friends in the virtual world.

Drive together with your friends.

Compare scores with your friends. Try to be the most eco-friendly driver among them.



Areas With Different Challenges

Highway – Maintain a steady speed and safe distance from other cars.

City – Beware of traffic lights and pedestrians.

Accidents – Reduce your speed and avoid hitting obstacles on the road.

Mountains – Safely increase your speed before going uphill.

NII Helmut PRENDINGER (Associate Professor)／国立情報学研究所 - Prendinger Laboratory (helmut@nii.ac.jp)
Arturo NAKASONE (Project Manager)／国立情報学研究所 - Prendinger Laboratory (arturonakasone@nii.ac.jp)

Transport Scenario Authoring for Driver Behavioral Data Collection in 3D Virtual Worlds

Kugamoorthy Gajananan[1] **Helmut Prendinger [1]** **Arturo Nakasone [1]** **Marc Miska[2]**

[1]National Institute of Informatics,
The Graduate University for Advanced Studies, Japan

[2]Smart Transport Research Center,
Queensland University of Technology, Australia

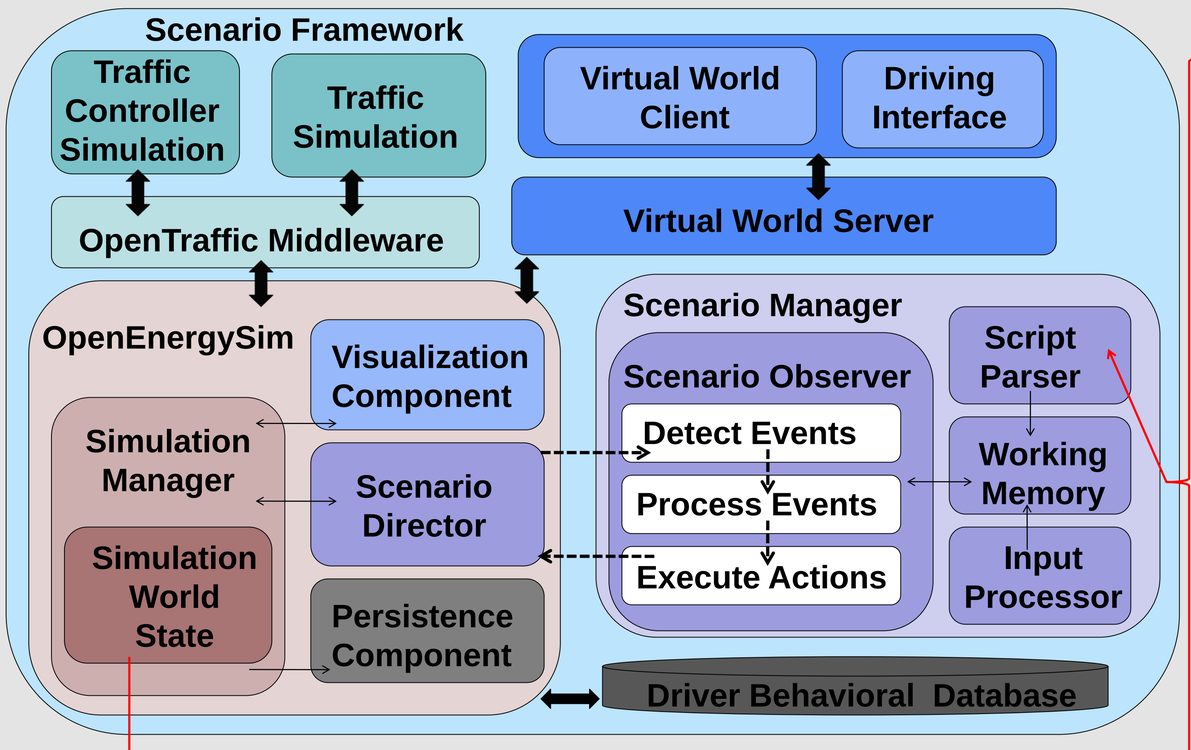
Background

Traffic engineers are interested in collecting detailed driver behavioral data of natural responses of humans exposed to specific traffic situations in virtual worlds. Data are analyzed to validate some hypotheses or develop and test theories and models that describe driving behavior under various transport conditions.

Objective

Our objective is to develop the Scenario Framework that includes (1) the Scenario Markup Language (SML), a simple yet expressive language for authoring realistic traffic situations, (2) capabilities for driver behavioral data collection. The framework extends the functionality of OpenEnergySim.

Scenario Framework Architecture



Scenario Authoring

```
<entity name  = "vehicle_one" id=  "vehicle_1" >
   <property name = “role" value = “collider"/>
    <property name = "hazard_light" value = "on"/>
</entity>

<scenario_director>
  <action id = "create_accident">
   <command id = "create_accident">
      <param name = "position" value = "100:21:20"/>
      <param name = "collider" value  = "vehicle_1"/>
      <param name = "collided-with" value  = "vehicle_2">
   </command>
  </action>
<scenario_director/>

<entity name  = "vehicle_two" id=  "vehicle_2" >
  <property name = “role" value = “collider"/>
  <property name = "hazard_light" value = "on"/>
</entity>

<event  id = “I_1">
 <condition><vars>
   <var  name = "severity" value  = "high">
    <var  name = "collider" value  = "vehicle_1"/>
    <var name = "collided-with" value  = "vehicle_2"/>
    <var  name = "position" value  = "100:21:20"/>
    <var  name = "incident_type" value  = "car_accident"/>
  </vars></condition>
</event>

<behavior  ref = "vehicle_2 "id = “SpeedControl" >
  <action id =“maintain_speed”><command>
  <param name = " aggressiveness" value = “high"/>
  <param name = “desired_speed“ value = "100km/h"/>
</command> </action></behavior>
```

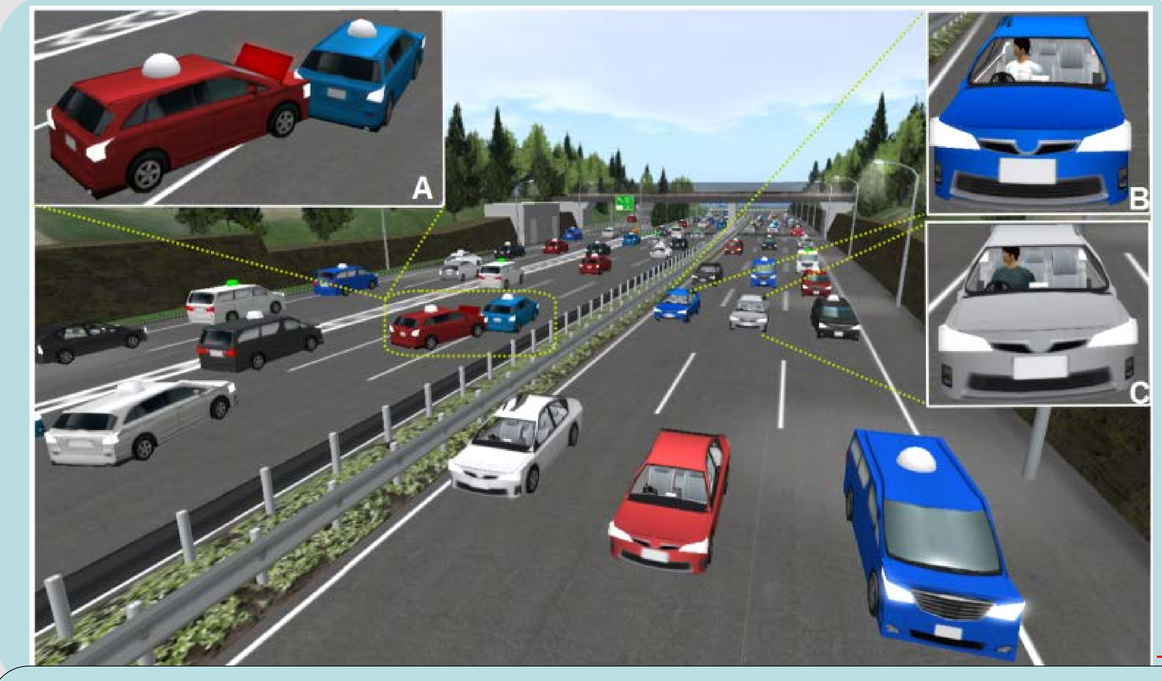
Scenario Director & Action

Entities & Properties

Event Definition

Behavior Definition

Virtual World Simulation



Rubbernecking Scenario : A shows an accident that happened on the left side of a road where vehicles 1 and 2 (specified in the script) have collided. B and C show the zoomed view of human drivers who are driving their cars on the other side (right) of the road, while they have a look at the accident – the rubbernecking effect. The picture also shows the cars from traffic simulation.

OpenEnergySim

OpenEnergySim is a virtual world based visualization application that integrates traffic simulation and immersive multi-user driving. This involves the synchronization of "ambient" traffic (i.e. traffic generated by a traffic simulator) and user-controlled cars (via a driving wheel or a game pad).

Scenario Markup Language (SML)

SML is a XML language that targets the scripting of entities that take part in complex, dynamic traffic scenarios, such as vehicles, pedestrians, traffic lights, and so on. This involves :

(1)The control of "high-level" behaviors of entities (fast driving, change of traffic light, etc), and

(2)The synchronization and orchestration of dynamic entities to induce specific traffic situations, such as an accident between two or more cars, or a car and a pedestrian.

The goal of SML is to create relevant traffic situations for behavioral driver studies.

Behavioral Driver Data Schema

User Driven (*) /Computer Controlled (^) Vehicle Data			
Attribute	**Type/Unit**	**Attribute**	**Type/Unit**
Vehicle Id	Integer	Vehicle Type	String
Current Position	X:Y:Z (Double)	Gear *	Double
Previous Position	X:Y:Z (Double)	RPM *	Double
Origin ^	X:Y:Z (Double)	Gas *	Double
Destination ^	X:Y:Z (Double)	Distance Travelled	Double
Current Speed	X:Y:Z (Double)	CO_2 Emission	Double
Previous Speed	X:Y:Z (Double)	NOx Emission	Double
Acceleration	X:Y:Z (Double)	Indicator Left	Boolean
Steering Wheel Position *	Float (-1 to 1)	Indicator Right	Boolean
Acceleration Pedal Position *	Float (0 to 1)	Break Light	Boolean
Brake Pedal Position *	Float (0 to 1)	Timestamp	HH:MM:SS

NII Helmut PRENDINGER (Associate Professor) (helmut@nii.ac.jp) , Kugamoorthy GAJANANAN (k-gajananan@nii.ac.jp)
国立情報学研究所 - Prendinger Laboratory / TEL : +81-3-4212-2650 FAX : +81-3-3556-1916