VanetMobiSim – Vehicular Ad hoc Network mobility extension to the CanuMobiSim framework

Contact

Jérôme Härri

Institut Eurécom
Department of Mobile Communications
06904 SophiaAntipolis, France

E-mail: haerri@eurecom.fr

Marco Fiore

Politecnico di Torino
Dipartimento di Elettronica
Corso Duca degli Abruzzi 24, Torino, Italy

E-mail: marco.fiore@polito.it

Contents

1. Introduction

The Vehicular Ad Hoc Networks Mobility Simulator (VanetMobiSim) is a set of extensions to CanuMobiSim, a framework for user mobility modeling used by the CANU (Communication in Ad Hoc Networks for Ubiquitous Computing) Research Group [1], University of Stuttgart. The framework includes a number of mobility models, as well as parsers for geographic data sources in various formats, and a visualization module. The framework is easily extensible. It is based on the concept of pluggable modules.
The set of extensions provided by VanetMobiSim consists mainly on a *vehicular spatial model* using GDF-compliant data structures, and a set of *vehicular-oriented mobility models*. The vehicular spatial model is composed of spatial elements, their attributes and the relationships linking these spatial elements in order to describe vehicular areas. The spatial model is created from topological data obtained in four different ways:

- *User-defined* – The user defines a set of vertices and edges composing the backbone of the vehicular spatial model.
- *Random* – The backbone is randomly generated using the Voronoi tessellations.
- *Geographic Data Files (GDF)* – Backbone data is obtained from GDF files.
- *TIGER/line Files* – Similar to the previous one, but based on the TIGER/line files from the US Census Bureau.

Any one of those methods MUST be loaded AFTER the Spatial Model, as it controls all data describing the topology. Then, it adds vehicular specific spatial elements such as multi-lane and multi-flow roads, stop signs and traffic lights.
The main component of the vehicular oriented model is the support of a microscopic level mobility model named "Intelligent Driving Model with Intersection Management (IDM_IM)" describing perfectly car-to-car and intersection managements. In the Intelligent Driving Model with Lane Changing (IDM_LC), we also included an overtaking model (MOBIL), which interacts with IDM_IM to manage lane changes and vehicle accelerations and decelerations. A *Spatial Environment* MUST be loaded for *user-oriented* and *vehicular oriented* mobility models to work. A spatial Environment MAY be loaded for all other mobility models specified in CanuMobiSim.

2. Installation

The framework's binary files are compressed to a jar-archive. To start the application, you need the JAVA Runtime Environment (v1.3 or later) [10] and the Xerces2 Java Parser library (v2.4 or later) [19]. If you work with geographic data in AWML or GDF format, you also need the GeoTransform package [7].

To launch the framework, change to the directory with framework's files and type:

```
java -jar VanetMobiSim.jar [scenario.xml]
```

The framework simulates user mobility according to a *simulation scenario*. On exit, it returns one of the following codes:

- 0 – successful execution,
- 1 – simulation aborted, error message is written to System.err

3. Format of Simulation Scenario

The simulation scenario for VanetMobiSim is similar to CanuMobiSim's. It is defined in XML format. In the notation below, tags or attributes appearing in square brackets (e.g., [<sample>] are optional.

3.1 Specifying a Simulation Area

A simulation area is specified using the *<universe>* tag.

```
<universe>
	[<dimx>dimension</dimx>]
	[<dimy>dimension</dimy>]
	[<step>step</step>]
	[<seed>seed</seed>]
	[<extension>extension_parameters</extension>]
	[<node>node_parameter</node>]
	[<nodegroup>nodegroup_parameters</nodegroup>]
</universe>
```

- dimx – specifies the x-dimension of the simulation are (in meters). Only used in the scenarios with rectangular-bounded simulation areas.
- dimy – specifies the y-dimension of the simulation are (in meters). Only used in the scenarios with rectangular-bounded simulation areas.
- step – specifies the duration of single simulation time-step (in s). If omitted, the value of 1 ms is used.
- seed – specifies the seed of the random number generation used by VanetMobiSim.
- extension – adds an instance of a global extension to the simulation.
- node – adds a node to the simulation.
- nodegroup – adds a group of nodes to the simulation.

3.2 Adding a Global Extension to Simulation

An instance of global extension is added using the <extension> tag.

```
<extension class="class_name" [name="instance_name"]>
		[extension_parameters]
</extension>
```

- class – specifies the name of class to be instantiated. The class must be derived from *de.uni_stuttgart.informatik.canu.mobisim.core.ExtensionModule* and be accessible by JVM.
- name – specifies the name of class instance. Used to uniquely identify and reference the instance in simulation. Most of extensions have their default name predefined.

Example:

```
<extension class="de.uni_stuttgart.informatik.canu.spatialmodel.simulations.TimeSimulation" param="3600.0"/>
```

3.3 Adding a Node to Simulation

A node is added to simulation using the <node> tag.

```
<node [class="class_name"] id="node_id">
        [<position>position_parameters</position>]
        [<type>type_of_node</position>]
        [<extension>extension_parameters</extension>]
</node>
```

- class – specifies the node's class name. The class must be derived from *de.uni_stuttgart.informatik.canu.mobisim.core.Node* and be accessible by the JVM. If omitted, *de.uni_stuttgart.informatik.canu.mobisim.core.Node* is used\
- id – specifies the node's id. Used to uniquely identify and reference the node in simulation
- position – specifies the node's initial position
- type – specifies the node's type. The user can choose among four different types of nodes: **ped-car-truck-bus**. If omitted, the value "*any*" is taken by default.
- extension – adds an extension to the node (e.g., instance of mobility model).

Example:

```
< node id="#0">
  <position random="true"/>
  <type="ped"/>
  <extension class="de.uni_stuttgart.informatik.canu.mobisim.mobilitymodels.RandomWaypointWalk">
    <minspeed>0.56</minspeed>
    <maxspeed>1.39</maxspeed>
    <minstay>120</minstay>
    <maxstay>600</maxstay >
  </extension>
</node>
```

3.3.1 Specifying the Node's Initial Position

The node's initial position is specified using the <position> tag.

```
<position [graph="graph_instance_name"] [random="is_random"]>
        [<x>x_value</x>]
        [<y>y_value</y>]
</position>
```

- graph – if the position belongs to the graph, specifies the name of graph instance (class *de.uni_stuttgart.informatik.canu.mobisim.extensions.Graph*). Used by the graph-based mobility model.
- random – specifies that the position must be chosen randomly. The value is of Boo-lean type. If the position belongs to a graph, it will be chosen randomly from the graph vertices.

- x – specifies the position's x-coordinate (in m).
- y – specifies the position's y-coordinate (in m).

Example:

```
<position>
   <x>12.0</x>
   <y>25.0</y>
</position >
```

3.4 Adding a Group of Nodes to Simulation

Multiple nodes (a group of nodes) are added to the simulation using the <nodegroup> tag.

```
<nodegroup n="number_of_nodes" [class="class_name"] id="group_id">
        [<position>position_parameters</position>]
        [<type>type_of_nodes</position>]
        [<extension>extension_parameters</extension>]
</node>
```

- n – specifies the number of nodes in the group.
- class – specifies the node's class name. The class must be derived from *de.uni_stuttgart.informatik.canu.mobisim.core.Node* and be accessible by the JVM. If omitted, *de.uni_stuttgart.informatik.canu.mobisim.core.Node* is used\
- id – specifies the group id. Used for choosing nodes' identifiers by concatenating the group id with the node's sequence number.
- position – specifies the initial position for all nodes in the group.
- type – specifies the node's type assigned to all member of this group. The user can choose among four different types of nodes: **ped-car-truck-bus**. If omitted, the value "*any*" is taken by default.
- extension – adds instances of extension to each node.

Example:

```
< nodegroup n="50">
   <position random="true"/>
   <type="ped"/>
   <extension class=" de.uni_stuttgart.informatik.canu.uomm.ConstantSpeedMotion">
     initposgeneratore="PosGen" tripgenerator="TripGen">
     <minspeed>0.56</minspeed>
     <maxspeed>1.39</maxspeed>
   </extension>
</node>
```

4. Globally Specified Extensions

The following extensions can be added globally to a simulation. All of them require an instance of de.uni_stuttgart.informatik.canu.spatialmodel.core.SpatialModel.

4.1 Spatial Environment

A spatial environment is added with an instance of *de.uni_stuttgart.informatik.canu.spatialmodel .core.SpatialModel* extension. **For a correct behavior of VanetMobiSim, this extension shall not be omitted.** As the spatial environment extension controls all topological and mobility extensions, it **MUST** be declared before them. It also adds support for multilane or multi-flow roads and traffic lights at intersections. It can be initialized from three different ways. First, it can be initialized from an appropriate geographic data source (GDF or TIGER). Or, it can also be initialized from an appropriate *eurecom.usergraph.UserGraph* extension. Finally, it can also be initialized from an appropriate eurecom.spacegraph.SpaceGraph extension.

<extension name="*instance_name*" class="*de.uni_stuttgart.informatik.canu.spatialmodel.core.SpatialModel*"
    [traffic_light=" *trafficlight_instance_name*"] min_x="*value*" min_y="*value*" max_x="*value*" max_y="*value*">
        [<max_traffic_lights>*traffic_lights*</max_traffic_lights>]
        [<number_lane full="*value*" max="*value*" dir="*value*">*lanes_number*</number_lane>]
        [<reflect_directions>*value*</reflect_directions>]
</extension>

- name – specifies the name of class instance. Used to uniquely identify and reference the instance in simulation. The default name is "SpatialModel". Changing the default name is not recommended.
- traffic_light – specifies the name of the *eurecom.spatialmodel.extensions.TrafficLight* extension if different from the default value.
- min_x – specifies the leftmost x-coordinate of "clipping window" (in m). The coordinate is relative to the source's minimal x-coordinate. Used to process a part of geographic area.
- min_y – specifies the lowermost y-coordinate of "clipping window" (in m). The coordinate is relative to the source's minimal y-coordinate. Used to process a part of geographic area.
- max_x – specifies the rightmost x-coordinate of "clipping window" (in m). The coordinate is relative to the source's maximal x-coordinate. Used to process a part of geographic area.
- max_y – specifies the uppermost y-coordinate of "clipping window" (in m).
- max_traffic_lights – specifies the number of intersections managed by traffic lights. This tag will have no effect if the eurecom.spatialmodel.extensions.TrafficLight is not declared after this extension. The default value is *5*.
- number_lane – specifies the number and characteristics of multi-lane roads. The default value is *1*. The maximum value is *4*.
- reflect_directions – specifies if the spatial model physically differentiates the two traffic flows. The default value is *false*. This value MUST match the value from the Trip Generator.

Example:

```
<extension name="UserSpatialModel" class="de.uni_stuttgart.informatik.canu.spatialmodel.core.SpatialModel"
traffic_light="UserTrafficLight" min_x="0" max_x="1000" min_y="0" max_y="1000">
   <max_traffic_lights>6</max_traffic_lights>
   <reflect_directions>true</reflect_directions>
   <number_lane full="false" max="4" dir="true">2</number_lane>
</extension>
```

4.1.1 Specifying Multi-lane Roads

The characteristics of multi-lane roads in the Spatial model are specified using the <number_lane> tag. When specifying a multi-lane road, the spatial model intends to model highways and therefore will generate a multi-lane highway starting from one border and ending on a different border using the Dijkstra shortest path algorithm.

<number_lane full=”*value*” max=”*value*” dir=”*value*”>*lanes_number*</number_lane>

- full – specifies whether all roads have multiple lanes or not.
- max – if the <full> attribute is false, specifies the maximum number of roads with multi-lane capability in the graph, i.e. the size of the subset of roads modeled as highways. If omitted, the default value is 4.
- dir – specifies if the spatial model physically differentiates the two traffic flows. If the <full> attribute is true, <dir> and <reflect_directions> are equivalent. If not, <dir> allows the user to differentiate the directional flows of multi-lane roads only. If omitted, the default value is *false*.
- lanes_number – specifies the number of lanes in multi-lane roads. If omitted, the default value is 1.

Example:

```
<number_lane full="false" max="4" dir="true">2</number_lane>
```

4.2 Traffic Lights

Support for traffic lights at intersections can be added using an instance of the eurecom.spatialmodel.extensions.TrafficLight extension. In order for this extension to work, the user needs to declare it before any topological extension (eurecom.spacegraph.SpaceGraph, eurecom.spacegraph.UserGraph, eurecom.spacegraph.TIGERReader, eurecom.spacegraph.GDFReader) and be sure to have specified at least one traffic light in the *de.uni_stuttgart.informatik.canu.spatialmodel .core.SpatialModel* extension.

<extension name="*instance_name*" class="*eurecom.spatialmodel.extensions.TrafficLight*"
[spatial_model=” *spatialmodel_instance_name*”] step="*interval*"/>

- name – specifies the name of class instance. Used to uniquely identify and reference the instance in simulation. The default name is “TrafficLight”. Changing the default name is not recommended.

- spatial_model – specifies the name of the *de.uni_stuttgart.informatik.canu.spatialmodel.core.SpatialModel* extension if different from the default value.
- step – specifies the time interval between traffic light changes (in ms).

Example:

```
<extension name="UserTrafficLight" class="eurecom.spatialmodel.extensions.TrafficLight" step="10000"
spatial_model="UserSpatialModel"/>
```

4.3 Graph Representation of Movement Area

A graph representation of movement area is added to simulation with an instance of eurecom.usergraph.UserGraph. As de.uni_stuttgart.informatik.canu.mobisim.extensions.Graph does NOT support a Spatial Model, we created this new extension which also let the user define a graph with a set of vertices and edges. We encourage the user NOT to use the de.uni_stuttgart.informatik.canu.mobisim.extensions.Graph extension for any configurations, as VanetMobiSim has been designed to let the Spatial Model be the central element even for non-user specific models such as the graph-based mobility model. It plays the same role and contains the same information than de.uni_stuttgart.informatik.canu.mobisim.extensions.Graph yet with more details on the topological structure and attributes. Besides, the graph may also be obtained directly from the Spatial Model.

```
<extension class=”eurecom.usergraph.UserGraph” [name=”instance_name”] >
        <vertex>vertex_parameters</vertex>
        <edge>edge_parameters</edge>
</extension>
```

- name – specifies the name of class instance. Used to uniquely identify and refer-ence the instance in simulation. The default name is “Graph”
- k – specifies the scale factor for vertices’ coordinates. If omitted, the value of 1.0 is used
- vertex – specifies a graph’s vertex
- edge – specifies a graph’s edge

Example:

```
<extension class="eurecom.usergraph.UserGraph">
  <vertex>
    <id>0</id> <x>800</x> <y>200</y>
  </vertex>
  <vertex>
    <id>1</id> <x>500</x> <y>600</y>
  </vertex>
  <vertex>
    <id>2</id> <x>700</x> <y>500</y>
  </vertex>
  <vertex>
    <id>3</id> <x>200</x> <y>300</y>
  </vertex>
  <vertex>
    <id>4</id> <x>800</x> <y>400</y>
  </vertex>
  <vertex>
    <id>5</id> <x>300</x> <y>100</y>
  </vertex>
```

```
  <edge> <v1>0</v1> <v2>1</v2> <speed>1</speed> </edge>
  <edge> <v1>0</v1> <v2>2</v2> <speed>15</speed> </edge>
  <edge> <v1>1</v1> <v2>3</v2> <speed>7</speed> </edge>
  <edge> <v1>1</v1> <v2>4</v2> <speed>21</speed> </edge>
  <edge> <v1>2</v1> <v2>4</v2> <speed>9</speed> </edge>
  <edge> <v1>3</v1> <v2>5</v2> <speed>5</speed> </edge>
  <edge> <v1>4</v1> <v2>5</v2> <speed>6</speed> </edge>
</extension>
```

4.3.1 Specifying a Graph's Vertex

A graph's vertex is specified using the <vertex> tag.

```
<vertex>
  <id>id</id>
  <x>x_value</x>
  <y>y_value</y>
</vertex>
```

- id – specifies the vertex's id. Used to uniquely identify and reference the vertex in the graph
- x – specifies the vertex's x-coordinate (in m)
- y – specifies the vertex's y-coordinate (in m)

Example:

```
<vertex>
  <id>0</id>
  <x>800</x>
   <y>200</y
</vertex>
```

4.3.2 Specifying a Graph's Edge

A graph's edge is specified using the <edge> tag.

```
<edge>
  <v1>v_id</v1>
  <v2>v_id</v2>
  [<speed>value</speed>]
</edge>
```

- v1 – specifies the id of first edge's vertex
- v2 – specifies the id of second edge's vertex
- speed – specifies the maximum speed on that vertex. The default value is *50km/h*.

Example:

```
<edge>
 <v1>0</v1>
 <v2>1</v2>
 <speed>15</speed>
</edge>
```

4.4 Space Graph

A space graph is added with an instance of eurecom.spacegraph.SpaceGraph extension. This creates a random graph, built by applying a Voronoi tessellation to a set of randomly distributed points (*obstacles*). It is possible to define areas (*clusters*) with different obstacles densities, creating a non-homogeneous graph. The user should be made aware of the fact that the traffic lights specified by this extension are represented as road furniture, whereas the eurecom.spatialmodel.extensions.TrafficLight is in charge of managing their tasks.

```
<extension class="eurecom.spacegraph.SpaceGraph" [name="instance_name"]
    [spatial_model=" spatialmodel_instance_name "] [traffic_light=" trafficlight_instance_name "]
    [max_obstacle ="value"][cluster="value"]>
        <clusters density="value">cluster_parameters</clusters>
</extension>
```

- name – specifies the name of class instance. Used to uniquely identify and reference the instance in simulation. The default name is “SpaceGraph”. Changing the default name is not recommended.
- spatial_model – specifies the name of the *de.uni_stuttgart.informatik.canu.spatialmodel.core.SpatialModel* extension if different from the default value.
- traffic_light – specifies the name of the *eurecom.spatialmodel.extensions.TrafficLight* extension if different from the default value.
- max_obstacle – specifies the number of obstacles required to generate the homogeneous voronoi tessellation, i.e. when clustering is not used. The default value is 40.
- cluster – specifies if clustering is used to create the Space Graph. The default value is *false*.
- clusters – specifies the clusters characteristics if cluster has been set to true.

Example:

```
<extension class="eurecom.spacegraph.SpaceGraph" spatial_model="UserSpatialModel"
traffic_light="UserTrafficLight" max_obstacle="55" cluster="true">
  <clusters density="0.000004">
    <cluster id="downtown">
      <density>0.0002</density>
      <ratio>0.1</ratio>
    </cluster>
    <cluster id="residential">
      <density>0.00005</density>
      <ratio>0.4</ratio>
    </cluster>
    <cluster id="suburban">
      <density>0.00001</density>
      <ratio>0.5</ratio>
    </cluster>
  </clusters>
</extension>
```

4.4.1 Specifying the Space Graph Clustering

The characteristics of the clustering of a Space Graph are specified using the <clusters> tag. Each cluster is a rectangular portion of the simulation area, characterized by a particular obstacles density.

```
<clusters density="value">
        <cluster id="cluster_name">cluster_parameters</cluster>
</clusters>
```

- density – specifies the density of clusters (in clusters/m^2): as an example, a clusters density value of $4e^{-6}$ in a topology of 1000 m^2 means that the simulation area is divided in 4 clusters. Then, the Space Graph will dispatch the different clusters in the simulation area according to the order of description and the corresponding ratios, and fill in the possible remaining cluster slots with the lowest dense cluster.
- cluster – specifies the parameters of the single cluster.

Example:

```
<clusters density="0.000004">
  <cluster id="downtown">
    <density>0.0002</density>
    <ratio>0.1</ratio>
  </cluster>
  <cluster id="residential">
    <density>0.00005</density>
    <ratio>0.4</ratio>
  </cluster>
  <cluster id="suburban">
    <density>0.00001</density>
    <ratio>0.5</ratio>
  </cluster>
</clusters>
```

4.4.1.1 Specifying a Cluster's Characterization

The characteristics of a particular cluster specified using the <cluster> tag.

```
<cluster id="cluster_name">
        <density>obstacles_density</density>
        <ratio>cluster_type_ratio</ratio>
        [<speed>value</speed>]
</cluster>
```

- id – specifies the identifier of the cluster
- density – specifies the density of obstacles in the cluster (in obstacles/m^2): as an example, a density value of $2e^{-4}$ means that there are 2 obstacles every 100 m^2 in average.
- ratio – specifies the percentage of this kind of cluster in the simulation area: this value can range in [0,1], and must be consistent with similar values of other clusters, so that the ratios of all the clusters defined sum to 1.
- speed – specifies the maximum speed in m/s allowed on the road segments created with this cluster type. The default value is 50km/h.

Example:

```
<cluster id="downtown">
   <density>0.0002</density>
   <ratio>0.1</ratio>
   <speed>20</speed>
</cluster>
```

4.5 Producing an Image of the Spatial Model

An image of the Spatial Model can be extracted in XFIG (.fig) format using the de.uni_stuttgart.informatik.canu.spatialmodel.extensions.DumpSpatialModel extension.

<extension class="*de.uni_stuttgart.informatik.canu.spatialmodel.extensions.DumpSpatialModel*" [spatial_model=”*spatial_model*”] output="*file_name*"/>

- spatial_model – specifies the name of the *de.uni_stuttgart.informatik.canu.spatialmodel.core.SpatialModel* extension if different from the default value.
- output – specifies the name of the .fig file containing the Spatial Model screenshot

Example:

```
<extension class="de.uni_stuttgart.informatik.canu.spatialmodel.extensions.DumpSpatialModel"
 spatial_model="UserSpatialModel" output="dumped_graph.fig"/>
```

4.6 Producing Simulator-Independent Mobility Traces

Node mobility traces in a generic format (node identifier, time, position, speed) are produced with an instance of de.uni_stuttgart.informatik.canu.mobisim.extensions.ReportNodeMobility extension.

<extension class="*de.uni_stuttgart.informatik.canu.spatialmodel.extensions.ReportNodeMobility*" [spatial_model=”*spatial_model*”] output="*file_name*">
<step>*value*</step>
</extension>

- spatial_model – specifies the name of the *de.uni_stuttgart.informatik.canu.spatialmodel.core.SpatialModel* extension if different from the default value.
- output – specifies the name of the .fig file containing the Spatial Model screenshot
- step – specifies the time step for mobility recording

Example:

```
<extension class="de.uni_stuttgart.informatik.canu.spatialmodel.extensions.ReportNodeMobility "
 output="mobility.tr">
 <step>1.0</ step>
</extension>
```

4.7 GDF Writer

A GDF writer module is added with an instance of eurecom.gdfwriter.GDFWriter extension. It parses the Spatial Model data representation into GDF records, and then writes the result to a file. All characteristics supported by the Spatial Model extensions may be parsed to a GDF file. The GDF generator is only a parser. The full spatial environment needs to be configured by the instances of a topological extension AND *de.uni_stuttgart.informatik.canu.spatialmodel.core. SpatialModel*. **For a correct behavior of the GDF Generator, the Spatial Model and any Topological extensions shall not be omitted.**

<extension class="eurecom.gdfwriter.GDFWriter" [spatial_model=”*spatial_model*”] output="*file_name*"/>

- spatial_model – specifies the name of the *de.uni_stuttgart.informatik.canu.spatialmodel.core.SpatialModel* extension if different from the default value.
- output – specifies the name of the .gdf GDF dataset file.

Example:

```
<extension class="eurecom.gdfwriter.GDFWriter"  spatial_model="UserSpatialModel" output="dumped_GDF.gdf"/>
```

4.8 Parser for TIGER Data Sources

A spatial environment is loaded from a TIGER data source by an instance of eurecom.TIGERReader extension. For easy map creation, the user MUST specify the GPS coordinates of the map’s central point and then specify the size of the map. The coordinates of created elements are relative to the source’s minimal x- and y- coordinates. TIGER maps have two levels of detail. Maps may be drawn with or without “*Shape Points*”. Shape points are intermediate points in road segments that allow the display of curved road segments. As the end-points and the shape points are provided in two different TIGER files (.TR1 and .TR2 respectively), we let the user decide the level of accuracy it wishes to obtain. Therefore, the source file name SHALL NOT contain any extension. Depending on the “*shapeCoord”* tag, the eurecom.TIGERReader extension will or will not load the shape points. As the TIGER files do NOT specify maximum speed limits on street segments, as an alternative to a default initialization based on the Californian Speed Limitations, we let the user provide VanetMobiSim with a file containing the speed limit per road category. Finally, as the definition of the clipping area is performed differently from the spatial model, the clipping area defined in the eurecom.TIGERReader extension overrides the one from *de.uni_stuttgart.informatik.canu.spatialmodel.core.SpatialModel*.

<extension class=”*eurecom.TIGERReader*” source=”*file_name*” [name=”*instance_name*”] [traffic_light=” *trafficlight_instance_name* ”] [shapeCoord="value"] [center_lat=”*value*”] [center_long=”*value*”] [size_x=”*value*”][size_y=”*value*”]>
<speed>*file_name*</speed>
</extension>

- name – specifies the name of class instance. Used to uniquely identify and reference the instance in simulation. The default name is “GDFReader”

- source – specifies the name of TIGER file to parse. The name is relative to current directory, uses a *url* style format, and shall not contain the TIGER file extension, as the simulator will load on-demand (see next) the multiple files describing the desired US district area.
- shapeCoord – specifies if the simulator shall include road segments' intermediate shape points, that is, if the TIGER map shall include curbed roads or not. Default value is false.
- traffic_light – specifies the name of the *eurecom.spatialmodel.extensions.TrafficLight* extension if different from the default value.
- center_lat – specifies the latitude of the map's central point. The precision MUST be of 8 digits
- center_long – specifies the longitude of the map's central point. The precision MUST be of 8 digits.
- size_x – specifies the length of the X-axis of the "clipping window" (in m).
- size_y – specifies the length of the X-axis of the "clipping window" (in m).
- speed – specifies the reference to a file containing a different mapping between road types and speed limits than the default value. The user must provide the path to the file if it is not in the current directory.

Example:

```
<extension class="eurecom.TIGERReader" source="file://${VanetMobisim_Dir}/samples/TIGER/TGR11001"
  center_lat="+38905050" center_long="-77016160" size_x="2000" size_y="2000">
  <speed>speedLimits.txt</speed>
</extension>
```

4.8.1 Speed Limitation File

As some speed limitations in the US are specific to the state or to the county (such as Freeway speed limits), TIGER files do not contain speed constraints per road element. Instead, it contains the universal street category in the form *letter+3digits*. Therefore, the structure of a speed limitation file MUST follow these guide lines:

- First Column – *Road Category*. As TIGER street classes are large, we advice the user to employ main categories. For example, a road class *A41* represents an un-separated local, neighborhood or rural street, while most roads in the US are classified simply as *A4*, or freeways as *A1*.
- Second Column – *Speed.* The user will provide the speed in Miles per hour (mph). VanetMobiSim will parse the values into meters/second.

Example:

```
A1    65
A2    55
A3    45
A4    35
A5    15
A6    25
```

4.9 Parser for GDF Data Sources

A spatial environment is loaded from a GDF data source by an instance of de.uni_stuttgart.informatik.canu.gdfreader.GDFReader extension. Unlike the CANU version, the VanetMobiSim GDFReader uses the clipping windows of the spatial model. Therefore, it cannot accept clipping windows definitions. Unless otherwise specified in the GDF file, speed limits are set by default to *50km/h* for each road element.

<extension class=”*de.uni_stuttgart.informatik.canu.gdfreader.GDFReader*” source=”*file_name*” [name=”*instance_name*”] [scale_x=”*value*”] [scale_y=”*value*”] [min_x=”*value*”] [min_y=”*value*”] [max_x=”*value*”] [max_y=”*value*”] />

- name – specifies the name of class instance. Used to uniquely identify and refer-ence the instance in simulation. The default name is “GDFReader”
- source – specifies the name of GDF file to parse. The name is relative to current directory
- scale_x – specifies the scale factor for x-coordinates. If omitted, the value of 0.01 is used
- scale_y – specifies the scale factor for y-coordinates. If omitted, the value of 0.01 is used

Example:
```
<extension class="de.uni_stuttgart.informatik.canu.gdfreader.GDFReader" source="GDF/Schwabch.gdf"
```

5. Node-Specific Extensions (Mobility Models)

The following vehicular specific extensions can be specified for each node.

5.1 Simulating Node’s Motion using the IDM_IM model

To simulate node’s motion using the Intelligent Driver Model with Intersection Management, an instance of polito.uomm.IDM_IM extension is used. It regulates vehicles speed based on movements of neighboring vehicles (e.g., if a car in front brakes, the succeeding vehicles also slow down). **Only the vehicles moving according to the Intelligent Driver Model or Intelligent Driver Model with Intersection Management are considered!** The implementation reflects restrictions of the spatial environment. Vehicles moving according to the Intelligent Driver Model with Intersection Management model support smart intersection management: they slow down and stop at intersections, or act according to traffic lights, if present.

<extension class=”*polito.uomm.IDM_IM*” [spatial_model=”*spatialmodel_instance_name*”]
initposgenerator=”*pos_gen*” tripgenerator=”*trip_gen*” [name=”*instance_name*”]>
<minspeed>*value*</minspeed>
<maxspeed>*value*</maxspeed>
[<l>*value*</l>]
[<a>*value*]
[*value*]
[<s0>*value*</s0>]
[<t>*value*</t>]
<step>*value*</step>

<stay [random="value"]>[value]</stay>]
[<ignoreBorders>value</ignoreBorders>]
</extension>

- name – specifies the name of class instance. Used to uniquely identify and reference the instance in the list of node's extensions. The default name is "Movement". Changing the default name is not recommended.
- spatial_model – specifies the name of the *de.uni_stuttgart.informatik.canu.spatialmodel.core.SpatialModel* extension if different from the default value.
- initposgenerator – specifies the name of instance that assigns node's initial position
- tripgenerator – specifies the name of instance that generates node's trips
- minspeed – specifies the minimal speed of movement (in m/s)
- maxspeed – specifies the maximal speed of movement (in m/s)
- l – specifies the vehicle's length (in m). If omitted, the value of 5 m is used
- a – specifies the maximal acceleration of movement (in m/s^2). If omitted, the value of 0.6 m/s^2 is used
- b – specifies the "comfortable" deceleration of movement (in m/s^2). If omitted, the value of 0.9 m/s^2 is used
- s0 – specifies the minimal distance to a standing node (jam distance) (in m). if omitted, the value of 2 m is used
- t – specifies the node's safe time headway parameter (in s). If omitted, the value of 1.5 s is used
- step – specifies the step for recalculating movement parameters (in s)
- stay – specifies the node's initial stay duration (in s). If omitted, the node initiates a new movement at the beginning of simulation
- random – specifies that the initial stay duration must be chosen randomly ("true" or "false")
- ignoreBorders – specifies if intersections located at borders of the map must be ignored by the intersection management engine. This can be especially useful when simulating a city section, with vehicles entering and exiting the road map. Can be "true" or "false"

Example:

```
<extension class="polito.uomm.IDM_IM" spatial_model="UserSpatialModel"
  initposgenerator="Gen" tripgenerator="Gen">
    <minspeed>0.1</minspeed>
    <maxspeed>19.4</maxspeed>
    <b>0.5</b>
    <step>1.0</step>
</extension>
```

5.2 Simulating Node's Motion using the IDM_LC Model

To simulate node's motion using the Intelligent Driver Model with Lane Changing, an instance of polito.uomm.IDM_LC extension is used. It regulates vehicle speed based on movements of neighboring vehicles (e.g., if a car in front brakes, the succeeding vehicles also slow down). **Only the vehicles moving according to the IDM_LC, IDM_IM or IDM model are considered!** The implementation reflects restrictions of the spatial environment. Vehicles moving according to the IDM_LC model support smart intersection management: they slow down and stop at intersections, or act according to traffic lights, if present. The implementation reflects restrictions of the spatial environment. Also, vehicles are able to change lane and perform overtakings in presence of multi-lane roads.

```
<extension class=”polito.uomm.IDM_LC” [spatial_model=”spatialmodel_instance_name”] initposgenerator=”pos_gen”
    tripgenerator=”trip_gen” [name=”instance_name”]>
        <minspeed>value</minspeed>
        <maxspeed>value</maxspeed>
        [<l>value</l>]
        [<a>value</a>]
        [<b>value</b>]
        [<s0>value</s0>]
        [<t>value</t>]
        <step>value</step>
        [<stay [random=”value”]>[value]</stay>]
        [<ignoreBorders>value</ignoreBorders>]
        [<bsave>value</bsave>]
        [<p>value</p>]
        [<athr>value</athr>]
</extension>
```

- name – specifies the name of class instance. Used to uniquely identify and reference the instance in the list of node’s extensions. The default name is “Movement”. Changing the default name is not recommended.
- spatial_model – specifies the name of the *de.uni_stuttgart.informatik.canu.spatialmodel.core.SpatialModel* extension if different from the default value.
- initposgenerator – specifies the name of instance that assigns node’s initial position
- tripgenerator – specifies the name of instance that generates node’s trips
- minspeed – specifies the minimal speed of movement (in m/s)
- maxspeed – specifies the maximal speed of movement (in m/s)
- l – specifies the vehicle’s length (in m). If omitted, the value of 5 m is used
- a – specifies the maximal acceleration of movement (in m/s^2). If omitted, the value of 0.6 m/s^2 is used
- b – specifies the “comfortable” deceleration of movement (in m/s^2). If omitted, the value of 0.9 m/s^2 is used
- s0 – specifies the minimal distance to a standing node (jam distance) (in m). if omitted, the value of 2 m is used
- t – specifies the node’s safe time headway parameter (in s). If omitted, the value of 1.5 s is used
- step – specifies the step for recalculating movement parameters (in s)
- stay – specifies the node’s initial stay duration (in s). If omitted, the node initiates a new movement at the beginning of simulation
- random – specifies that the initial stay duration must be chosen randomly (“true“ or “false”)
- ignoreBorders – specifies if intersections located at borders of the map must be ignored by the intersection management engine. This can be especially useful when simulating a city section, with vehicles entering and exiting the road map. Can be “true” or “false”
- bsave – specifies the maximum “safe” deceleration (in m/s^2). If omitted, the value of 4 m/s^2 is used
- p – specifies the *politeness factor* of drivers when changing lane. If omitted, a value of 0.5 is used
- athr – specifies the threshold acceleration (in m/s^2) for lane change, must be lower than the acceleration parameter “a”. If omitted, a value of 0.2 m/s^2 is used.

Example:

```
<extension class="polito.uomm.IDM_LC" spatial_model="UserSpatialModel"  initposgenerator="Gen"
    tripgenerator="Gen">
      <minspeed>0.1</minspeed>
      <maxspeed>19.4</maxspeed>
      <step>1.0</step>
      <p>0.8</p>
</extension>
```