**Simmobility Geospatial Data Converter (SimGDC)**

**User’s Guide**

**1/ Installation**

SimGDC is a QGIS plugin which can convert Simmobility Geospatial XML file to shapefiles. User can then use QGis to edit those shapefiles and convert them back to XML format.

To install this plugin, following these steps:

**Step 1:** Download and install QGis from <http://www.qgis.org/en/site/forusers/download.html>

SimGDC is only compatible to *QGis 2.0 and later.*

**Step 2:** Download SimGDC source code from <https://github.com/smart-fm/dynavis/tree/tuan_fyp>

**Step 3:** Copy SimGDC source code to QGis python plugin folder.

QGis python plugin folder is usually located in the following path:

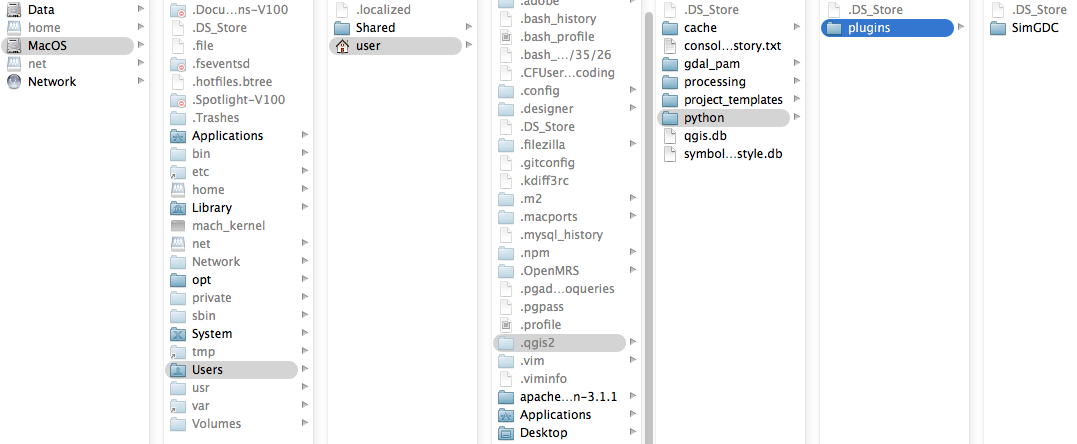
UNIX/Mac: ~/.qgis2/python/plugins or (qgis\_prefix)/share/qgis2/python/plugins

Windows: ~/.qgis2/python/plugins or (qgis\_prefix)/python/plugins

Home directory (denoted by above ~) on Windows is usually something like C:\Documents and Settings\(user).

In all platforms, “.qgis2” is usually a hidden folder, so you may have to enable the option to view hidden file/folder to see the QGis python plugin folder.

If you can find .qgis2 but not “python” or “plugins” folders, please go ahead and create them.



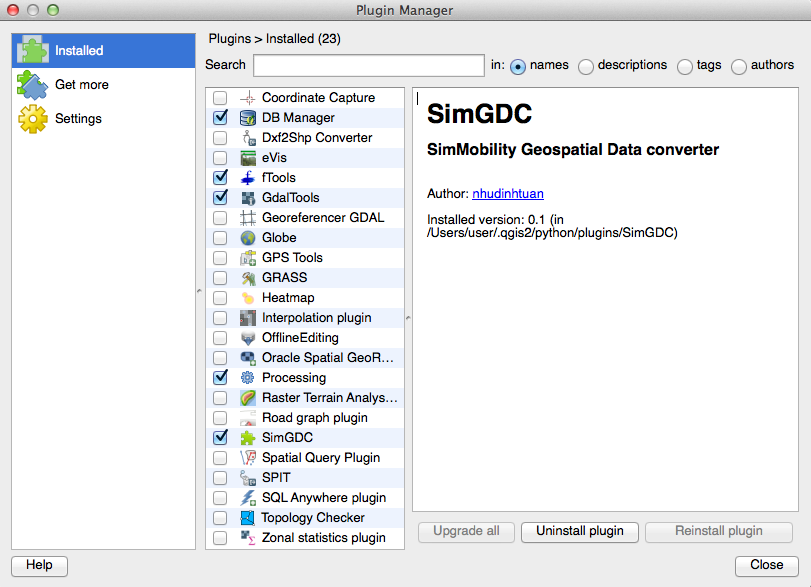
*Figure 1. Example of QGis python plugin folder in Mac OS.*

Note:

+ If you get problems to install QGis 2.0 in Ubuntu 13, you can follow this http://askubuntu.com/questions/351899/qgis-quantum-gis-install-fails-unmet-dependencies

**Step 4:** Run QGis Desktop and activate SimGDC plugin as below:

If SimGDC is copied to the correct plugin folder, SimGDC is listed in the plugin manager dialog (Plugins/Manager and Install Plugins…). Activate this plugin by clicking the checkbox to make it available to use.



*Figure 2. Plugin Manager Dialog*

**2/ Prerequisite**

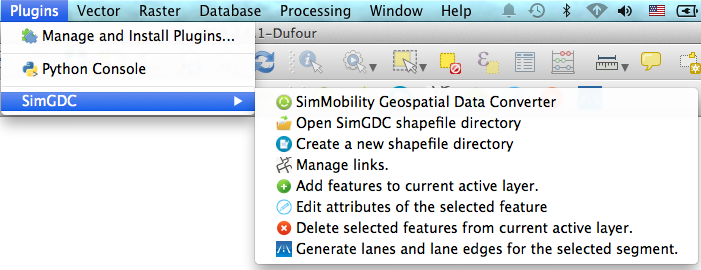
* You should have some background in QGis application before using this plugin. You can find QGis User Guide at http://www.qgis.org/en/docs/index.html
* It is good to be familiar with the SimMobility Geospatial Data XML structure in order to use this plugin properly.

**3/ Functionalities**

After the plugin is activated, the icons for SimGDC are shown on the toolbar and the SimGDC functionalities are listed in the menu bar.

Screen Shot 2014-05-04 at 5.09.16 am.png

*Figure 3a. SimGDC on the toolbar*



*Figure 3b. SimGDC functionalities on the Plugin menu bar*

In this part, we talk in detail about these functionalities.

**2.1 SimMobility Geospatial Data Converter**

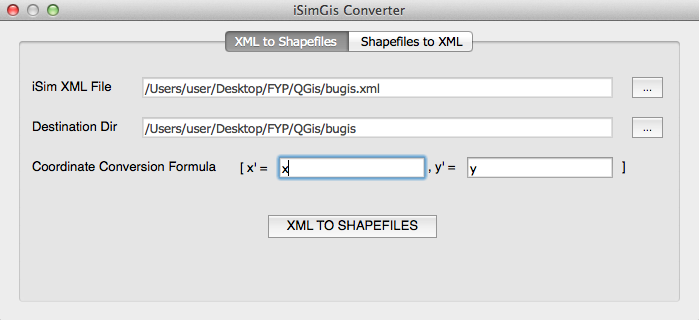
*Icon:* converter.png

*Description:* This functionality allows users to convert the SimMobility geospatial data from XML to shapefile format and from the shapefile format back to XML.

One single SimMobility Geospatial Data XML file is converted into multiple shapefiles. Each shapefile contains a particular type of geospatial data. They are uninodes, multinodes (or intersections), segments, lanes, lane edges, crossing, bus stops. All these shapefiles are located in a single folder. This folder can be converted back into a single XML file.

*How to use:* Clicks the icon, converter dialog is opened. There are two tabs in the dialog

**XML to Shapefiles**



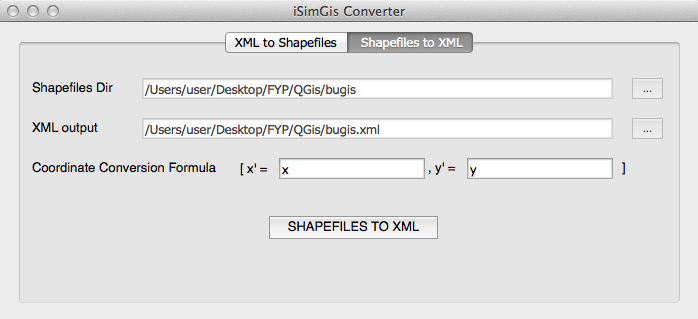
*iSim XML File:* SimMobility Geospatial XML file.

*Destination Dir:* An *empty* folder is to contain all shapefiles after converting. The folder name is used as the prefix for shapefiles and vector layers in QGis. It is important to make the folder empty before selecting it as the destination. If the folder is not empty, some files may be overwritten unpredictable.

*Coordinate Conversion formula:* This is useful when you want to change Coordinate Reference System. For example, the coordinate in XML is centimeter, you can change it to meter by the formula x’ = x/100; y’ = y/100.

After converting, all generated shapefiles are automatically loaded into QGis.

**Shapefiles to XML**



Similar to “XML to Shapefiles”.

**2.2 Open SimGDC shapefile directory**

*Icon:* open.png

*Description:* There are many shapefiles generated from a single XML file. Hence, this functionality is a fast way to load all of these shapefiles into QGis.

*How to use:* Clicks the icon, select the folder containing the shapefiles generated by SimGDC.

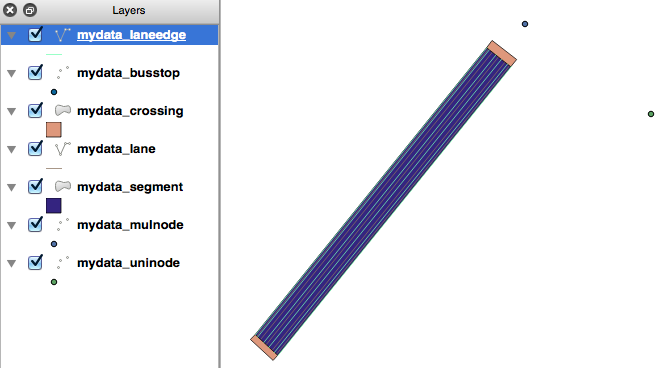
All shapefiles in the folder are loaded into QGis.

**2.3 Create a new shapefile directory**

*Icon:* notepad-icon.png

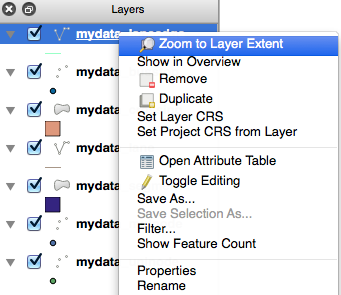
*Description:* This functionality allows users to create a simple geospatial data in shapefile format. It generates a basic geospatial data which contains one segment, one uninode and one multinode. From these basic data, you can easily add new objects to make a complete road network and convert it to XML file to become a valid input for SimMobility.

*How to use:* Clicks the icon, select an empty folder to contain the generated shapefiles. All shapefiles are loaded into QGis.



*Figure 4. New shapefiles*

Note: If you see an empty canvas after all shapefiles are loaded, you have to zoom the canvas to the location containing the data:



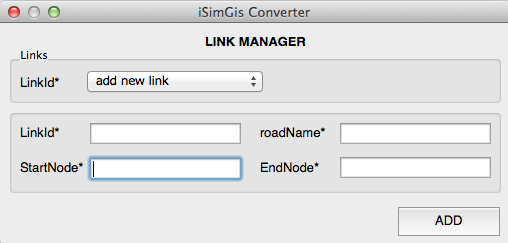
**2.4 Manage links**

*Icon:* linkmanager.png

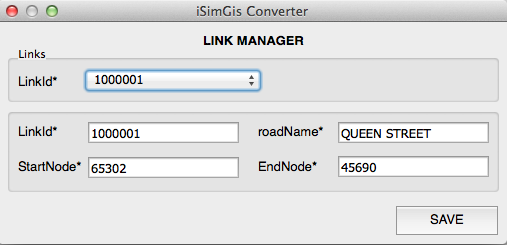
*Description:* SimMobility Geospatial Data contains links, but links do not have shape information. Hence there is no shapefile for links after converting from XML to Shapefile. SimGDC provides this functionality for users to manage the links.

*How to use:* Clicks the icon, the link manager dialog is opened.

Select “add new link” from LinkId option to add a new link.



Select a particular LinkId to edit the existed link information.



**2.5 Add features to current active layer**

*Icon:* add.png

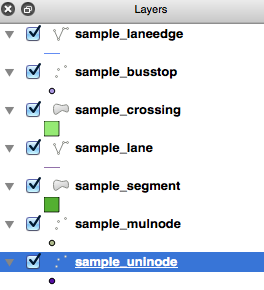
*Description:*

Components in SimMobility Geospatial Data XML does not contain only shape information, but also the attributes. Hence, if users use the normal QGis function to add a new feature into the shapefiles generated by SimGDC converter, they only add the shape but not the associated attributes. This is the reason why SimGDC provides this functionality. It allows users to add a new feature to the current active layer with the associated attributes.

*How to use:*

**To add uninode, multinode, segment:**

***Step 1:*** Select the layer which you want to add the feature

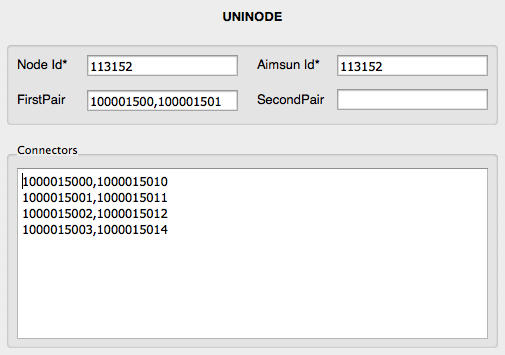


The above screenshot shows the *sample\_uninode* layer is selected to allow users to add a new uninode.

***Step 2:*** click the icon add.png, the mouse cursor becomes + when you move it on the canvas.

***Step 3:*** click a point on the canvas to place the feature. The attribute dialog is shown for you to enter the data. You should refer to the original XML file to see how the data in XML are shown in the dialog.

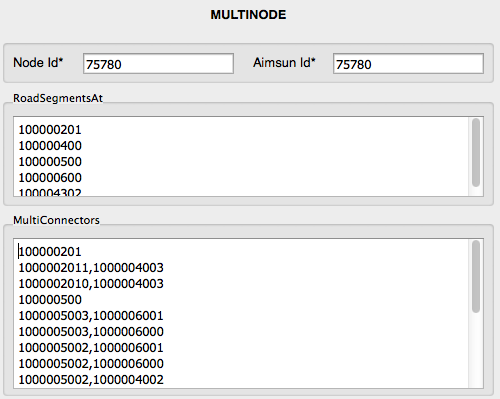
*Uninode Attribute Dialog*



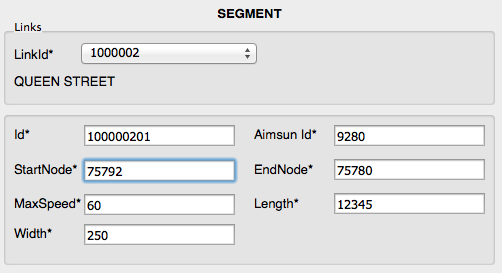
The above dialog is corresponding with the following data in XML file:



*Multinode Attribute Dialog*



*Segment Attribute Dialog*

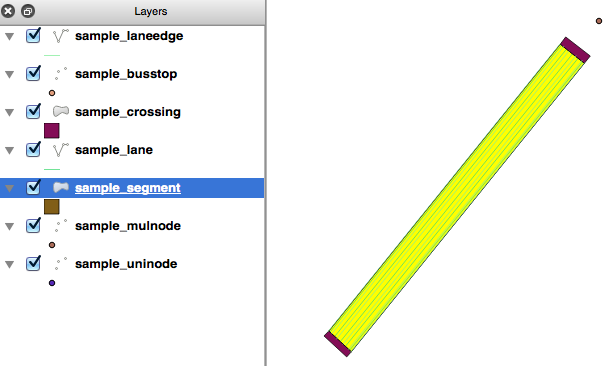


***Step 4:*** after finishing step 3, the feature is added to the layer. You may need to edit the shape (especially, for segment) by using the normal QGis function.

**To add lane, lane edge, crossing, bus stop:**

As a lane, lane edge, crossing or bus stop must belong to a segment, to add those components you must select a segment first and then follow the similar steps as above.

***Step 0:*** select the segment layer and select a segment you want to add a lane, lane edge, crossing or bus stop to that segment.



***Step 1, 2, 3, 4***: similar to add uninode, multinode, segment.

**2.6 Edit attributes of the selected feature.**

*Icon:* edit.png

*Description:* Users can use the normal QGis function to edit the shape of features. However, they can only edit the associated attributes to the features by using this functionality.

*How to use:*

***Step 1:*** select the layer and the feature you want to edit.

***Step 2:*** click edit.png icon, the attribute dialog is shown for you to edit.

*Note:* You can not undo the edited attributes after editing.

**2.7 Delete selected features from current active layer.**

*Icon:* delete.png

*Description:* Users can not use the normal QGis function to add or remove the feature from shapefile because the geospatial data do not contain only shape information, but also the associated attributes. Hence, this functionality allows users to delete the geospatial data feature properly. It will delete both shape information and associated attributes.

*How to use:*

***Step 1:*** select the layer and the feature you want to delete

***Step 2:*** click delete.pngicon to delete the feature.

*Note:*

+ You need to save all of your changes on the layer and deactivate the editing mode before deleting the feature (all editing icons are disabled)

Screen Shot 2014-05-04 at 10.14.17 pm.png

The screenshot of toolbar when the editing mode is deactivated.

+ You can not undo after deleting the feature.

**2.8 Generate lanes and lane edges for the selected segment.**

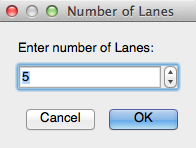
*Icon:* add_lane.png

*Description:* A segment has some lanes and lane edges. It is troublesome to add lane, lane edges to a new segment one by one. Hence, this functionality provides a fast way to generate the lanes and lane edges based on the number of lanes.

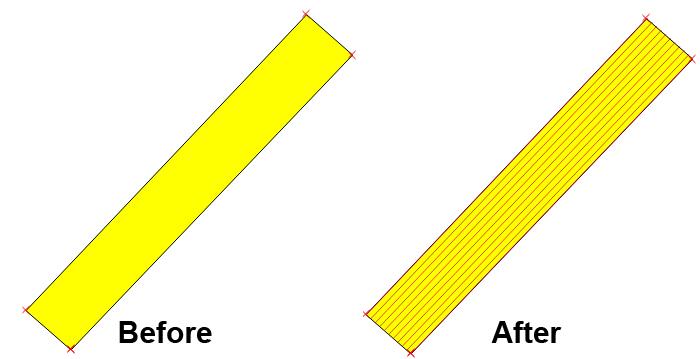
*How to use:*

***Step 1:*** select the segment layer and select the segment to add lanes and lane edges.

***Step 2:*** click add_lane.png icon, the dialog is shown



Enter the number of lanes you want to generate. The lanes and lane edges are automatically generated. The associated attributes for lanes and lane edges are set as the default, so you may need to edit them.



*Note:* If the segment has some lanes and lane edges before generating, they will be removed.

**Simmobility Geospatial Data Converter (SimGDC)**

**Developer’s Guide**

**1/ Prerequisite**

* Developers need to read and install the necessary software in the user’s guide.
* Read the QGis Developing Python Plugins <http://www.qgis.org/en/docs/pyqgis_developer_cookbook/plugins.html>
* Read the tutorial to develop a simple QGis Plugin (important!)

<http://www.qgisworkshop.org/html/workshop/plugins_tutorial.html>

* Install Python 2.7, pyqt

**2/ ESRI Shapefile**

ESRI Shapefile is one of the most popular geospatial vector data formats. It is supported in almost GIS software. QGIS uses Shapefile as the standard vector file format. Shapefile has advantages over other data sources in term of drawing speed and editing ability because it does not have the processing overhead of a topological data structure. Hence, Shapefile is used as the data format for our converter.

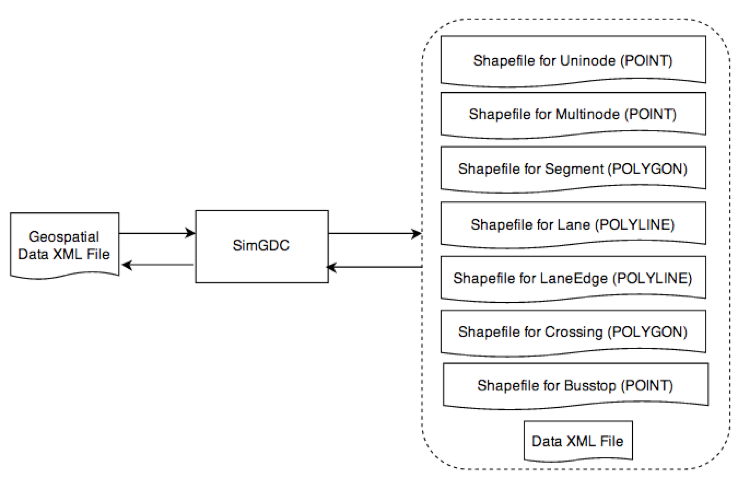
A Shapefile actually consists of several files because it does not only store nontopological geometry but also stores the attribute information. The following three files are required to make a Shapefile valid:

* “.shp” file: containing the geometries.
* “.dbf” file: containing the attributes associated with objects in “.shp” file. The attribute record size is limited to 4000 characters/object.
* “.shx” file: it is a geometry index file that allows seeking forwards and backwards quickly.

Shapefile supports different geometry types (such as polygon, point, polyline, etc.). However, Each Shapefile can contain only one geometry type, so the Shapefile must be defined as storing polygons, polylines or points.

**3/ Converting XML to Shapefile**

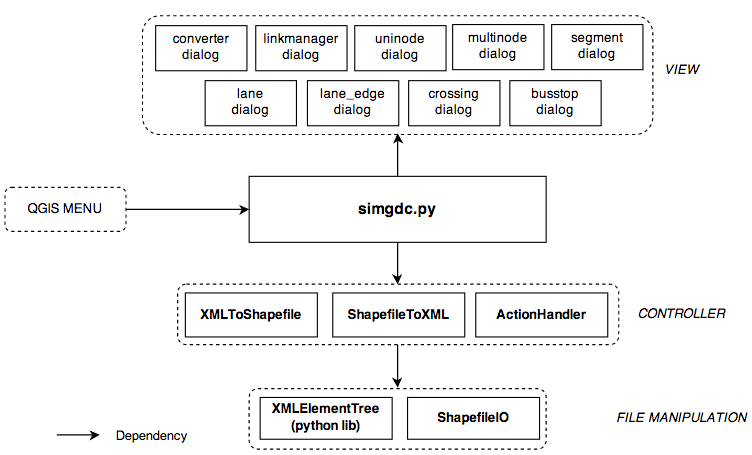
To understand how SimGDC is implemented, we first need to compare the structure of two data formats XML and Shapefile to figure out the good approach to convert data between them. The main difference between these data formats is data dependency. All objects in Shapefile are independent of each other; one object in Shapefile contains geometric location and associated attribute information. In SimMobility geospatial xml file, the hierarchical structure is used. For instance, one link may contain many segments; each segment contains lanes, bus stops, and crossings. Furthermore, not all components in geospatial data have geometric information.



*Figure 1. Converting geospatial data xml file to Shapefile*

The above diagram demonstrates our approach to convert the SimMobility geospatial data xml file to Shapefile. A single SimMobility geospatial data xml file is actually converted into seven Shapefiles and one xml file (called Data XML File). Each Shapefile is generated by extracting the location information from the input geospatial data xml file. They are classified by the type of components. The Data XML File is almost similar to the origin geospatial data xml file except the location information is removed. Seven Shapefiles are linked to the Data XML File by ids. For example, each geometric object in the Shapefile for lane has the attribute in the format (link-id, segment-id, lane-number). This attribute helps SimGDC to associate the geometric object with the Data XML File when converting Shapefiles back to SimMobility geospatial data xml file.

**4/ Architecture Diagram**



*Figure 2. The architecture diagram of SimGDC*

SimGDC is implemented as a plugin for QGis. The main controller which handles the connection between QGis menubar, toolbar, canvas and SimGDC is *simgdc.py.* This main controller calls view components and other controllers to respond to users interaction.

VIEW: The view of SimGDC are dialogs to get the user input or show the attributes of components.

CONTROLLER: Besides the main controller, there are three sub controllers to handle user’s requests.

* *XMLToShapefile*: This controls all necessary actions to convert XML to Shapefile.
* *ShapefileToXML*: this controls all necessary actions to convert Shapefile to XML.
* *ActionHandler*: This handles all necessary actions to create, edit, delete the geospatial components.

FILE MANIPULATION: SimGDC needs some components to read, write and modify XML file and Shapefiles. To handle XML file, we use the standard library provided by Python. It is *XMLElementTree*. For Shapefile, the ShapefileIO (containing ShapefileReader and ShapefileWriter) is implemented based on the QGis core module, QgsVectorFileWriter & QgsVectorLayer.

**5/ Future Work**

1. Attributes Dependency Verification.

One of the main limitation for the current SimGDC is that it does not verify the dependency inside attributes. For example, the attribute of segment contains StartNode and EndNode. SimGDC does not show the warnings or errors if these nodes are not existed. Furthermore, if users delete a node, SimGDC also does not warn users if other segments still have that node as StartNode and EndNode.

2. Geometric Information Dependency Verification

It is important that all lanes and lane edges should locate on the segment. However, the current version of SimGDC does not verify that. Users can move the lanes and lane edges to any place they want. Hence, it is good if SimGDC can check the dependency on geometric information and warn users if there is something wrong.