Clustering data analysis:

file:///D:/verkehr/sumo-doc-0.27.1/sumo-0.27.1/docs/userdoc/Networks/SUMO\_Road\_Networks.html

Nodes, usually named "junctions" in SUMO-context, represent intersections, and "edges" roads or streets. Note that edges are unidirectional. Specifically, the SUMO network contains the following information:

* every street (edge) as a collection of lanes, including the position, shape and speed limit of every lane,
* traffic light logics referenced by junctions,
* junctions, including their right of way regulation,
* connections between lanes at junctions (nodes).

Normal Edges

A "normal" edge is a connection between two nodes ("junctions").

<edge id="<ID>" from="<FROM\_NODE\_ID>" to="<TO\_NODE\_ID>" priority="<PRIORITY>">

... one or more lanes ...

</edge>

The attributes of an edge are:

|  |  |  |
| --- | --- | --- |
| **Name** | **Type** | **Description** |
| id | id (string) | The id of the edge |
| from | id (string) | The id of the node it starts at |
| to | id (string) | The id of the node it ends at |
| priority | integer | Indicates how important the road is (optional) |
| function | enum ( "normal", "internal", "connector" ) | An abstract edge purpose (optional with default "normal") |

* normal: The edge is a plain part of a road network, like a highway or a normal street which connects two roads
* connector: The edge is a macroscopic connector - not a part of the real world road network. Still, within the simulation, no distinction is made between "connector" roads and "normal" nodes. Only [**SUMO-GUI**](file:///D:\verkehr\sumo-doc-0.27.1\sumo-0.27.1\docs\userdoc\SUMO-GUI.html) allows to hide connector edges.
* internal: The edge is a part of an intersection (is located within the intersection), see above.
* The attributes of a lane are:

|  |  |  |
| --- | --- | --- |
| **Name** | **Type** | **Description** |
| **id** | id (string) | The id of the lane |
| **index** | running number (unsigned int) | A running number, starting with zero at the right-most lane |
| **speed** | float | The maximum speed allowed on this lane [m/s] |
| **length** | float | The length of this lane [m] |
| **shape** | position vector | The geometry of the lane, given by a polyline that describes the lane's centre line; must not be empty or have less than two positions |

Include traffic lights in clustering?

### Traffic Light Programs

A traffic light program defines the phases of a traffic light.

<tlLogic id="<ID>" type="<ALGORITHM\_ID>" programID="<PROGRAM\_ID>" offset="<TIME\_OFFSET>">

<phase duration="<DURATION#1>" state="<STATE#1>"/>

<phase duration="<DURATION#1>" state="<STATE#1>"/>

... further states ...

<phase duration="<DURATION#n>" state="<STATE#n>"/>

</tlLogic>

Junctions represent the area where different streams cross, including the right-of-way rules vehicles have to follow when crossing the intersection. An example may be:

<junction id="<ID>" type="<JUNCTION\_TYPE>" x="<X-POSITION>" y="<Y-POSITION>"

incLanes="<INCOMING\_LANES>" intLanes="<INTERNAL\_LANES>"

shape="<SHAPE>">

... requests ...

</junction>

The junction itself is described by the following attributes:

|  |  |  |
| --- | --- | --- |
| **Name** | **Type** | **Description** |
| **id** | id (string) | The id of the junction; please note, that a traffic light definition must use the same ID when controlling this intersection. |
| **x** | x-position (real) | The x-coordinate of the intersection |
| **y** | y-position (real) | The y-coordinate of the intersection |
| z | z-position (real) | The (optional) z-coordinate of the intersection |
| **incLanes** | id list | The ids of the lanes that end at the intersection; sorted by direction, clockwise, with direction up = 0 |
| **intLanes** | id list | The IDs of the lanes within the intersection |
| **shape** | position list | A polygon describing the road boundaries of the intersection |

### Connections

#### Plain Connections

Plain connections or "links" describe which outgoing lanes can be reached from an incoming lane. Additionally, the first lane to use for passing the intersection is given. Additional information describe the direction and the "state" of the connection. If the connection is controlled by a traffic light, the name of the traffic light and the index of the signal that controls this connection within the traffic light's phase definition is given. A connection is encoded as:

<connection from="<FROM\_EDGE\_ID>" to="<TO\_EDGE\_ID>" fromLane="<FROM\_LANE\_INDEX>" toLane="<TO\_LANE\_INDEX>"

via="<VIA\_LANE\_ID>" tl="<FROM\_EDGE\_ID>" linkIndex="12" dir="r" state="o"/>

The attributes are:

|  |  |  |
| --- | --- | --- |
| **Name** | **Type** | **Description** |
| **from** | edge id (string) | The ID of the incoming edge at which the connection begins |
| **to** | edge id (string) | The ID of the outgoing edge at which the connection ends |
| **fromLane** | index (unsigned int) | The lane of the incoming edge at which the connection begins |
| **toLane** | index (unsigned int) | The lane of the outgoing edge at which the connection ends |
| **via** | lane id (string) | The id of the lane to use to pass this connection across the junction |
| **tl** | traffic light id (string) | The id of the traffic light that controls this connection; the attribute is missing if the connection is not controlled by a traffic light |
| **linkIndex** | index (unsigned int) | The index of the signal responsible for the connection within the traffic light; the attribute is missing if the connection is not controlled by a traffic light |
| **dir** | enum ("s" = straight, "t" = turn, "l" = left, "r" = right, "L" = partially left, R = partially right, "invalid" = no direction) | The direction of the connection |
| **state** | enum ("-" = dead end, "=" = equal, "m" = minor link, "M" = major link, traffic light only: "O" = controller off, "o" = yellow flashing, "y" = yellow minor link, "Y" = yellow major link, "r" = red, "g" = green minor, "G" green major) |  |

Each roundabout is defined (somewhat redundantly) by its nodes and edges:

<roundabout nodes="nodeID1 nodeID2 ..." edges="edgeID1 edgeID2 ..."/>

Additional output:

# Further Outputs

## Information on Joined Junctions

The option **--junctions.join-output** [*<FILE>*](file:///D:\verkehr\sumo-doc-0.27.1\sumo-0.27.1\docs\userdoc\Basics\Notation.html#Referenced_Data_Types) causes a file to be written that specifies the junctions which were joined (usualy due to option **--junctions.join**). The resulting output file is suitable for loading with the **--node-files** option.

## Street Signs

The option **--street-sign-output** [*<FILE>*](file:///D:\verkehr\sumo-doc-0.27.1\sumo-0.27.1\docs\userdoc\Basics\Notation.html#Referenced_Data_Types) causes a file with [POIs](file:///D:\verkehr\sumo-doc-0.27.1\sumo-0.27.1\docs\userdoc\Simulation\Shapes.html#POI_.28Point_of_interest.29_Definitions) to be written. These POIs encode the type of street signs that are encountered on each edge and can be loaded as [*additional-file*](file:///D:\verkehr\sumo-doc-0.27.1\sumo-0.27.1\docs\userdoc\SUMO.html#Format_of_additional_files) in [SUMO-GUI](file:///D:\verkehr\sumo-doc-0.27.1\sumo-0.27.1\docs\userdoc\SUMO-GUI.html). Currently used sign types are:

* priority
* yield
* stop
* allway\_stop
* right\_before\_left

## Additional Information within the output file

The option **--output.street-names** [*<BOOL>*](file:///D:\verkehr\sumo-doc-0.27.1\sumo-0.27.1\docs\userdoc\Basics\Notation.html#Referenced_Data_Types) ensures that street names from suitable input networks such as [OSM](file:///D:\verkehr\sumo-doc-0.27.1\sumo-0.27.1\docs\userdoc\Networks\Import\OpenStreetMap.html) or [OpenDrive](file:///D:\verkehr\sumo-doc-0.27.1\sumo-0.27.1\docs\userdoc\Networks\Import\OpenDRIVE.html) are included in the generated *.net.xml* file.

When reading or writing OpenDrive networks, the option **--output.original-names** [*<BOOL>*](file:///D:\verkehr\sumo-doc-0.27.1\sumo-0.27.1\docs\userdoc\Basics\Notation.html#Referenced_Data_Types) [writtes additional data for mapping between sumo-ids and OpenDrive-ids into the generated networks](file:///D:\verkehr\sumo-doc-0.27.1\sumo-0.27.1\docs\userdoc\Networks\Import\OpenDRIVE.html#Referencing_original_IDs).

# Implementation phase

1. information gathering using lxml xpath and threadpools