

SHared automation **O**perating models for
Worldwide adoption
SHOW

Grant Agreement Number: 875530

**Simulation Suite: Connections between Simulation
Levels**



1 Simulation Levels

Various simulation tools have been broadly applied in the transportation field for properly answering research questions and evaluating methods, measures, and strategies proposed for reducing traffic burden and for assuring mobility. Traffic simulation models can be classified into four levels, i.e. macroscopic, mesoscopic, microscopic and ego-vehicle (sometimes also denoted as sub-microscopic). The first three levels can usually consider all road users and the respective travel performance metrics, e.g. travel times, delays and stops, emitted pollutants, etc. within a given network. Ego-vehicle simulations are also microscopic, but focus on the precise replication of interactions between a specific vehicle (the ego-vehicle) and road users/objects that surround it, as well as on its own dynamics and performance, usually including sub-models for the vehicle's sub-parts behaviour. The mathematical formalizations of the applied models are at different granularity levels, corresponding to the simulation levels.

The microscopic traffic simulation suite SUMO (Simulation of Urban Mobility) is adopted to set up the simulation environment for the test sites Linköping and Monheim am Rhein within the scope of the SHOW project. The simulation work can facilitate to understand the possible influences of automated shuttles on other road users in the shared area, at the bus terminal and in the campus area. Although the built simulation environment is at microscopic level, it can be transferred to several prevalent simulation tools. In the next chapter, the possibilities to connect to other tools regarding network and demand will be explained.

2 Possibilities to connect to other tools

SUMO provides several possibilities to interact with other tools for raising research synergies. The connections can be classified into two categories: one is the interchange of data (network and demand) and the other one is tool coupling. The former one lets generated data be easily converted between different formats, as required by SUMO or other tools. The latter one is to couple SUMO with other tools or other platforms for allowing an on-line interaction between them.

2.1 Network

SUMO's tool NETCONVERT provides the possibility to transfer a network between the SUMO format and other often used formats.

- **Plain**

When using the option `--plain-output-prefix <NET-FILE>`, NETCONVERT generates four files containing data about nodes, edges, connections and traffic light logics as described in [1]. This information is stored in XML files and can thereby be easily processed and transformed to match own needs.

- **OpenStreetMap (OSM)**

Due to its open source character OSM has been extensively used in practice. NETCONVERT can easily convert the road and rail networks included in OSM data to a SUMO network that can be used for simulations. In addition, a complete synthetic scenario, including e.g. network, traffic demand and traffic lights, can be quickly established with use of the script `osmWebWizard.py`.

After double clicking the python script a web page, connected to the OSM-database, will appear. Users can set the desired boundary, traffic demand (vehicles, pedestrians, bicycles, trams, buses etc.) and a time period. An illustration of the web page is shown in Figure 1. More information about OSM-related tools can be found in [2] and [3].

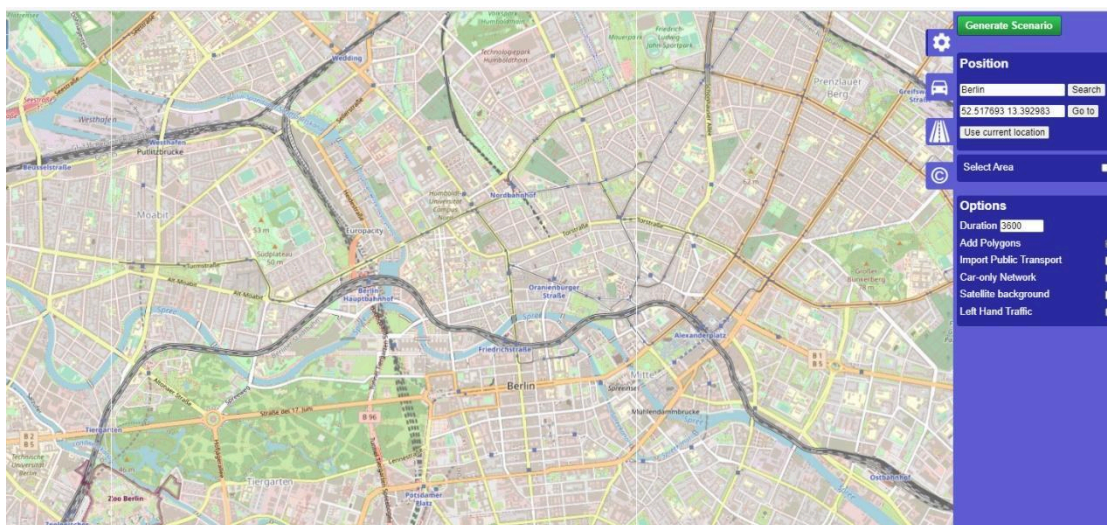


Figure 1 The illustration of the web page generated by `osmWebWizard.py`

- **OpenDRIVE**

Road networks stored in the OpenDRIVE format contain road-related information in a high resolution, including the roads' slope, width, and profiles, and are often used by driving simulators for testing vehicle functions. SUMO's NETCONVERT application can be used for both, converting OpenDRIVE networks to the network format used by SUMO or for exporting read road networks into the OpenDRIVE format.

With the following command:

```
netconvert --opendrive <OPENDRIVE-FILE> -o <NET-FILE>
```

an OpenDRIVE-based SUMO network can be quickly generated. It is applicable to the OpenDRIVE Versions 1.4, 1.5 and 1.6 [4]. Additional options can be set for defining parameters, e.g. speed, road width, and the elements, which should be considered in the network. With the option `--opendrive-output <NET-FILE>` NETCONVERT can write a given sumo network in the OpenDRIVE format (version 1.3).

- **MATSim**

MATSim is an open source agent-based transport simulation tool [5]. Given a SUMO network, a MATSim network can be generated with the option `--matsim-output <NET-FILE>` in NETCONVERT. At the same time, it is also possible to convert a MATSim Network to a SUMO network with the command `netconvert -matsim *.xml -o *.net.xml`.

- **DirNavteq**

As you can tell from the name, the DirNavteq network format is based on the Navteq format, but modified by DLR. Like converting MATSim and OpenDRIVE networks, NETCONVERT can also make the network conversion between SUMO and DirNavteq bi-directionally. The respective command examples are listed below.

```
netconvert --dlr-navteq <dlrNavteq-NET-FILE>_  
--output-file <SUMO-NET-FILE>
```

```
netconvert -s <SUMO-NET-FILE> --dlr-navteq-output  
<dlrNavteq-NET-FILE>
```

- **VISSIM/VISUM**

Both VISSIM and VISUM network files can be easily converted to SUMO networks with the following commands respectively. More information can be found in [6] and [7].

```
netconvert --vissim-file=<VISSIM-FILE>  
--output-file=<SUMO_NET-FILE>
```

```
netconvert --visum <VISUM-NET-FILE> -o <SUMO-NET-FILE>
```

- **ArcView**

Information in an ArcView-based geographic database can be extracted and used to build a network in the SUMO format. At least three files need to be given, i.e. dbf-file, shp-file and shx-file. A projection file with the extension name "proj" can help to set the network's correct coordinates, allocating it at the correct place on the

Earth globe and for facilitating possible mapping tasks. Because shape-file descriptions use more than one single file, only the file name without extension needs to be given [8]. The execution command is shown as below.

```
netconvert --shapefile-prefix <SHAPE-FILES>
```

An overview about the possible network importing and exporting with NETCOVERT is shown in Figure 2.

	Plain (XML)	OS M	OpenDRIVE	MATSim	DirNavteq	VISSIM/ VISUM	ArcView
import	X	x	x	x	x	x	x
export	X		x	x	x		

Figure 2 An overview of the network importing and exporting by NETCONVERT

2.2 Traffic demand and data

The SUMO applications suite supports different tools for importing traffic demand and detector data and converting them into the SUMO format.

- **VISSIM/VISUM**

With use of SUMO's tool `od2trips` VISUM's OD matrices can be imported and the respective trips with departure times can be generated in the SUMO format. It is also possible to extract VISUM's turn percentages at intersections and routes with the use of `visum_convertTurnPercentages.py` and `visum_convertRoutes.py`. The respective usage information can be found in [9].

Regarding VISSIM, the tool `vissim_parseRoutes.py` can process VISSIM's vehicle flows and route decisions and then parse the respective routes. The execution call is:

```
vissim_parseRoutes.py <VISSIM-NET-FILE> <OUTPUT_PREFIX>
```

Besides, bus stops in VISSIM can also be retrieved with the use of the call `vissim_parseBusStops.py <VISSIM-NET-FILE> <OUTPUT_PREFIX>`.

In addition, VISUM can store count positions, and optionally, their values. SUMO's tool `visum_parseZaehlstelle.py` can parse the respective positions and attributes, and, if necessary, fix the positions on the given network. The execution call is:

```
python visum_parseZaehlstelle.py <SUMO-NET-FILE>  
<VISUM-NET-FILE> <OUTPUT-FILE>
```

The parsed locations are saved as a list of POIs (points-of-interest), located on the rightmost lanes of matched roads. can be used as additional polygons and POIs.

- **MATSim**

MATSim uses road users' travel plans to describe the traffic demand within an area, and has tools to generate virtual population and the respective travel plans. SUMO's tool `matsim_importPlans.py` can help to import MATSim's person plans into SUMO. The execution call is:

```
python matsim_importPlans.py --plan-file  
<MATSIM-PLANS-FILE> -o <OUTPUT-FILE>
```

It is also possible only to import vehicular trips with the option `--vehicles-only`.

2.3 Tool coupling

2.3.1 Off-line coupling

Being started on the command line and due to using data defined in the extensible and exchangeable XML format, the traffic simulation SUMO can be easily integrated in loops that incorporate other applications. This is already the case for one of the standard processing loops in SUMO – the computation of a traffic equilibrium where both, SUMO, and the routing tool “duarouter” included in the suite are iteratively executed. In the past, the number of such processing workflows and loops grew continuously, included traffic demand generation (e.g. SUMOPy [10] or SAGA [11]) and calibration (e.g. Cadyts [12]). Such off-line workflows are mainly used in the scope of traffic modelling, where the performance or interactions of autonomous vehicles are not yet addressed. They will thereby not be described in a bigger detail herein. Further information can be found at [13].

2.3.2 On-Line coupling using TraCI

Since 2006, SUMO features an interface named TraCI (Traffic Control Interface) [14] which allows to exchange information with other applications or scripts on-line for both, obtaining information about the current state of the simulated scenario as well as for controlling the instances of the scenario. TraCI, which was extended in the following years, is capable to deliver information about almost all of the artefacts a simulation consists of – roads, lanes, traffic lights, and of course traffic participants – and their current states, and to change their current states.

The complete access to a running simulation offers a broad range of applications, mainly by avoiding the need to implement new algorithms directly into SUMO. This eases the development of new methods and algorithms a lot, as one does not need to get common to the internal representations of artefacts in SUMO. In addition, SUMO is written in the `c++` programming language, which is not known by many. Instead, using TraCI, one can control a running simulation using external scripts that may be written in any programming language. Standard implementations for accessing the simulation using TraCI are currently available for Python, Java, and Matlab.

In the past, TraCI has been used in the context of the development of traffic lights, new routing algorithms, reinforcement learning [15], and, above all, vehicular communications, using additional middleware applications like iCS [16] developed in the European project “COLOMBO”, TraNS, or Veins (see [17] for a more complete

list). Here, the middleware is used to connect SUMO to a communication network simulator, mainly ns-2 or ns-3, and a V2X-application to simulate.

A further specific application of the TraCI is the coupling of SUMO with a driving simulator. In such usage scenarios, an external simulator is responsible for supporting an ego vehicle's dynamics while SUMO supports the background traffic – the vehicles in the ego-vehicles surrounding.

Such applications have been extended in the past for connecting fine-grained sub-microscopic autonomous vehicle simulators with SUMO (see, e.g. [18]. Again, the major idea here is to supply such simulators with the environment and the background traffic, both of which are defined with and/or retrieved from SUMO. Within the SHOW project, the simulation work for the Karlsruhe test site is one example of such a coupling of SUMO with a simulation of an autonomous vehicle.

References

- [1] SUMO (2023). PlainXML. [PlainXML - SUMO Documentation \(dlr.de\)](https://sumo.dlr.de/docs/Tools/Import/PlainXML.html). Accessed August 29, 2023.
- [2] SUMO (2023). OSM. <https://sumo.dlr.de/docs/Tools/Import/OSM.html>. Accessed August 31, 2023.
- [3] SUMO (2023). OpenStreetMap. <https://sumo.dlr.de/docs/Networks/Import/OpenStreetMap.html>. Accessed September 15, 2023.
- [4] SUMO (2023). OpenDRIVE. <https://sumo.dlr.de/docs/Networks/Import/OpenDRIVE.html>. Accessed September 15, 2023.
- [5] MATSim (2023). Multi-Agent Transport Simulation. <https://www.matsim.org/>. Accessed August 31, 2023.
- [6] SUMO (2023). Extension tools to import features from VISSIM *.inpx files. <https://sumo.dlr.de/docs/Tools/Import/VISSIM.html> (in German). Accessed September 15, 2023.
- [7] SUMO (2023). Vissim. <https://sumo.dlr.de/docs/Networks/Import/Vissim.html>. Accessed September 15, 2023.
- [8] SUMO (2023). ArcView. <https://sumo.dlr.de/docs/Networks/Import/ArcView.html>. Accessed August 31, 2023.
- [9] SUMO (2023). Further ways to import VISUM demand definitions. https://sumo.dlr.de/docs/Demand/Further_Ways_to_import_VISUM_Demand_Definitions.html. Accessed September 19, 2023.
- [10] Joerg Schweizer (2014) *SUMOPy: An Advanced Simulation Suite for SUMO*. In: *Simulation of Urban Mobility*, p. 71-82, Springer, 10.1007/978-3-662-45079-6_6.
- [11] Lara Codeca, Jakob Erdmann, Vinny Cahill, Jérôme Härrri (2020) *SAGA: An Activity-based Multi-modal Mobility Scenario Generator for SUMO*. In: *SUMO User Conference 2020*, doi: <https://doi.org/10.52825/scp.v1i>
- [12] Yun-Pang Flötteröd (2017) *SUMO-Cadyts calibration with limited data quality*. *SUMO User Conference 2017 - Towards Simulation for Autonomous Mobility*.
- [13] SUMO (2023). Contributed. <https://sumo.dlr.de/docs/Contributed/index.html>. Accessed August 31, 2023.
- [14] Axel Wegener, Horst Hellbruck, Christian Wewetzer, Andreas Lubke (2008) *Vanet simulation environment with feedback loop and its application to traffic light assistance*. In: *GLOBECOM Workshops, 2008 IEEE*, pp. 1–7, 30.
- [15] The Mobile Sensing Lab of the UC Berkeley (2023). FLOW. <https://flow-project.github.io/>. Accessed August 31, 2023.
- [16] Daniel Krajzewicz, Robbin Blokpoel, Alessio Bonfietti, Jérôme Härrri, Stefan Hausberger, Jérémie Dubois-Lacoste (2015) *Traffic management based on vehicular communication at low equipment rates*. In: *Proceedings of the 22nd ITS World Conference*.
- [17] SUMO (2023). V2X simulations in SUMO. <https://sumo.dlr.de/docs/Topics/V2X.html>. Accessed September 19, 2023.
- [18] Eclipse MOAIC (2023) Eclipse MOAIC web pages, <https://eclipse.dev/mosaic/>