# Agent-based Simulation of Urban Air Mobility

Raoul L. Rothfeld*
*Bauhaus Luftfahrt e. V., Taufkirchen, Bavaria, 82024, Germany*

Milos Balac[†]
*ETH Zurich, Zurich, Canton of Zurich, 8093, Switzerland*

Dr. Kay O. Ploetner[‡]
*Bauhaus Luftfahrt e. V., Taufkirchen, Bavaria, 82024, Germany*

Prof. Dr. Constantinos Antoniou[§]
*Technical University of Munich, Munich, Bavaria, 80333, Germany*

**While a new generation of electric Vertical Take-off and Landing (eVTOL) or Personal Air Vehicles (PAV) projects receive much attention, little research is available on modeling their integration into urban transport systems. An early methodology of the Urban Air Mobility (UAM) extension for the multi-agent transport simulation, MATSim, is being presented, which enables first analyses of system-wide urban transport performance considering the effects of variations in VTOL vehicle properties and dedicated VTOL infrastructure placement.**

## Nomenclature

| | | |
|---|---|---|
| API | = | Application Program Interface |
| DVRP | = | Dynamic Vehicle Routing Problem |
| eVTOL | = | electric Vertical Take-off and Landing |
| GIS | = | Geographic Information System |
| PAV | = | Personal Air Vehicle |
| PT | = | Public Transport |
| UAM | = | Urban Air Mobility |
| VTOL | = | Vertical Take-off and Landing |

## I. Introduction

Urban Air Mobility (UAM), the concept of utilizing flying vehicles that are, predominantly, able to take-off and land vertically (VTOL), increasingly becomes of industrial and research interest. Though the idea of airborne mobility using VTOL vehicles is not novel, as helicopter-based taxi services have been and are still existent in global metropolises, it is now that technological advances enabled the development of so-called Personal Air Vehicles (PAV) or next-generation/electric VTOL (eVTOL) vehicles. Next-gen VTOL, based on advances in electrical and distributed propulsion as well as an increase in energy densities of batteries, are believed to be able to make urban air travel economically feasible and acceptable to society, by having a reduced noise and emission footprint in comparison to conventional helicopters.

While numerous PAV development projects (see Fig. 1 for an example) have been launched within the last years, as presented by Shamiyeh et al. [1], there is only little research on the potential integration of VTOL vehicles into urban transportation systems. Due to the limited space in urban environments, interaction between existing transport systems and UAM is expected to be required and unavoidable. Especially the number, extent, and placement of UAM infrastructure–as UAM is expected to require dedicated VTOL-infrastructure to ensure public and passenger safety–will

---

*Research Associate, Economics and Transportation, Raoul.Rothfeld@bauhaus-luftfahrt.net

[†]Research Associate, Institute for Transport Planning and Systems (IVT), Milos.Balac@ivt.baug.ethz.ch

[‡]Team Lead, Economics and Transportation, Kay.Ploetner@bauhaus-luftfahrt.net

[§]Head, Chair of Transportation Systems Engineering, C.Antoniou@tum.de

**Fig. 1    Rendering of Airbus' CityAirbus, a four-seats wingless PAV concept**

necessitate integration and, thus, changes to existing, ground-based transport infrastructure. It is, hence, fundamental to understand and evaluate the effects of UAM introduction on an urban transport system-wide scale and in the context of different urban settings from around the world.

In order to perform system-wide analyses, existing ground-based transportation simulation frameworks must be extended with UAM modeling capabilities. For this, UAM vehicles, infrastructure, and airspace requires definition, modeling, and simulation. It is the intention to provide such modeling capabilities and enable system-wide transport analyses, especially with regards to modal shares and overall transport performance, with the–hereinafter–presented UAM extension for the open-source multi-agent transport simulation framework, MATSim [2].

## II. Literature Review

Besides MATSim, there are other capable transport modeling frameworks, such as PTV Visum and TransCAD. PTV Visum, for example, utilizes the four-step modeling methodology for transport of (1) trip generation, (2) trip distribution, (3) mode choice, and (4) assignment; and, thus, is a trip-based, macroscopic model. TransCAD, on the other hand, utilizes geographic information system (GIS)-based travel demand modeling to provide multiple demand modeling methodologies, including the above-mentioned four-step approach, activity models, and sketch planning methods. MATSim [2] uses a microscopic, activity-based methodology, where individual households have their trips chained and modeled as parts of larger daily activity patterns. While each framework has its distinct advantages and disadvantages in, e.g., run-time, data requirements, and features, MATSim has been selected for UAM modeling as it is a research-driven, open-source, extensible, and license-free transport-modeling tool. Further, MATSim proofed to be a highly versatile simulation framework that has already been used to model novel transport modes and futuristic transport settings.

Further, according to Balac et al. [3], MATSim has been the "only known attempt to simulate commercial flights using an agent-based approach, where both aircraft and passengers are modeled on an agent level". The inclusion of air travel in MATSim was done by Grether [4] with its focus on conventional, i.e. inter-city, commercial air transport with large passenger aircraft. An introduction of UAM, with its small-scale vehicles and intra-city operations, thus, poses novel challenges for MATSim integration. With the introduction of autonomous vehicle simulation within MATSim [5, 6], as Balac et al. [3] states, the foundation for the introduction of UAM in MATSim has been set. Under the assumption of on-demand operational models for UAM, the transport modeling of VTOL vehicles mirrors that of autonomous ground-based taxis. Thus, the UAM extension for MATSim is also based on the Dynamic Vehicle Routing Problem (DVRP) MATSim contribution by Maciejewski [7].

As mentioned, MATSim uses an agent-based demand and traffic assignment method, where each agent has plans. Plans are, as Ziemke et al. [8] describes, a "chain of activities (e.g. home–work–shop–home), including their locations and end times". The plans are fulfilled using transport, by road network or public transport schedules, which have to be provided as input data together with a population definition. Within execution loop of MATSim (see Fig. 2), three major recurring events make up the main functionality of MATSim: (1) the mobility simulation (mobsim), (2) scoring, and (3) re-planning. The mobility simulation executes each agent's plan using the provided transport network–potentially causing partially high network loads, i.e. traffic congestion. During scoring, all executed plans are given a score by utility functions based on their plan's simulated performance. Finally, agents might re-plan their day by switching to an alternative plan or creating new ones utilizing different routes or modes of transport. This process is repeated until the system nears a Nash equilibrium-like state, were–as Ziemke et al. [8] describes it–the further development of agents' plan scores is "sufficiently relaxed".
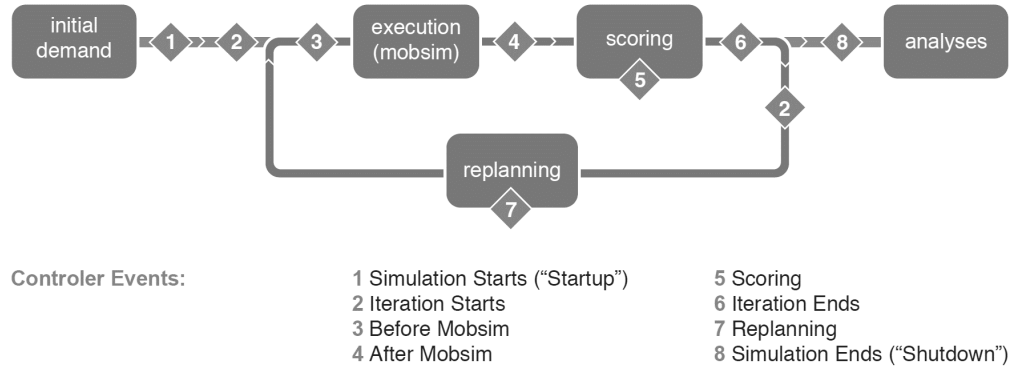
**Fig. 2  MATSim execution loop with numbered controller events [9]**

The major distinction between these autonomous ground-based vehicles and the introduction of UAM are: (1) the requirement of VTOL infrastructure for operation instead of roads, (2) the usage and management of urban airspace and aerial networks, and (3) the introduction of varying fleet properties and a fleet mix, i.e. different combinations of vehicle properties, such as speed, range, and maximum payload. In contrast to the modeling of conventional vehicles, not all VTOL vehicles are able to fulfill any given agent's trip, as there are great variations in current VTOL projects with some having ranges of less than 50 km [1]. Thus, the UAM extension provides the ability to define properties of potential on-demand VTOL vehicles and infrastructure. Additionally, the results of a stated preference survey for the Munich metropolitan region including autonomous flying taxis by Fu [10] and the methodology for positioning UAM infrastructure as presented by Fadhil [11] have partly been integrated within the UAM extension in MATSim. For further guidance, the implementation of public transport into MATSim by Rieser [12] has often been consulted.

## III. Methodology

The UAM extension of MATSim relies on the previous work of Maciejewski [7] on dynamic vehicle routing (DVRP), which provides modeling of one-way, autonomous, and on-demand vehicle usage, and makes extensive use of its API. In its current version, v0.2.3, the UAM extension provides three major building blocks for any potential UAM implementation: the UAM infrastructure, vehicles, and airspace usage.

### A. UAM Infrastructure, Vehicle, and Network Modeling

While road-able PAV concepts exist, which describes the ability to take-off from and land on common road infrastructure, the majority of concepts requires dedicated infrastructure for take-off and landing (see [1, 11]) and will be subject to similar regulations and specifications as conventional helicopter landing sites. As for VTOL infrastructure, its position and capacity have been identified as the main characteristics. Thus, for dedicated VTOL landing sites–i.e. UAM stations–the following properties have been implemented within the UAM extension:

*Properties of UAM Stations*
- Unique identifier (e.g. unique string or consecutive numbers)
- Location (i.e. the MATSim network link, to which the station is connected)
- Vehicle capacity (i.e. capacity for simultaneous VTOL operation)

Despite the various PAV archetypes and great variations between the concepts regarding performance proclamations (see [1]), first key PAV properties could be identified: The following parameters have, thus, been implemented in the current UAM extension version:

*Properties of UAM Vehicles*
- Unique identifier (e.g. unique string or consecutive numbers)
- Initial location (i.e. unique identifier of overnight-parking UAM station)
- Passenger capacity (i.e. maximum number of simultaneous passengers)

- Cruising speed (i.e. maximum horizontal flight speed)
- VTOL speed (i.e. maximum vertical ascent/descent speed)
- Maximum range (i.e. maximum distance between take-off and landing stations)
- Operating start and end times (i.e. earliest and latest time of day for vehicle usage)

In the current implementation, UAM vehicles can utilize airspace by traveling along predefined air routes (links) and way-points (nodes), which mirrors the implementation of MATSim's road network (see Rieser et al. [13]). Thus, the UAM network provides the same basic characteristics as conventional MATSim road networks:

*Properties of UAM Networks*
- Nodes
    - Unique identifier (e.g. unique string or consecutive numbers)
    - Location and height (i.e. $x$, $y$, and $z$ coordinates)
- Links
    - Unique identifier (e.g. unique string or consecutive numbers)
    - Origin and destination nodes (i.e. unique identifiers of the connected nodes)
    - Length (i.e. to be flown distance between the connected nodes)
    - Throughput capacity (i.e. number of vehicles that can traverse the link within a pre-defined time frame)
    - Maximum, free-flowing speed (i.e. maximum allowed/possible cruise speed)

Currently, additional road link parameters, such as the number of lanes and allowed modes, remain fixed for UAM networks with one lane and UAM being the only allowed mode, respectively. Despite using the same notation as MATSim's road network, the UAM network follows a distinct, suggested encoding–as illustrated in Fig. 3.
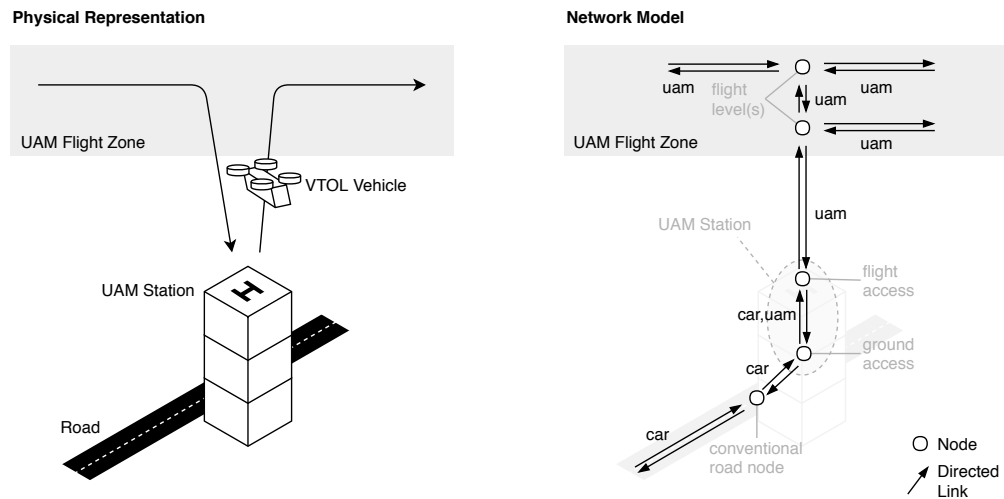


**Fig. 3   Illustration of current UAM network encoding, adapted from Rieser [12, p. 71]**

Two nodes, one ground- and one flight-access node, together represent a UAM station. The connecting links between those two nodes, the so-called station link, serves as the anchor for the previously describes UAM stations and should have a sufficiently short length as to avoid inflicting noticeable travel time. The ground access node is used to connect the UAM station to the conventional road network, e.g. by including one or more roads from existing road nodes in proximity. The flight access node of the UAM station is used to connect the station to one or more flight levels, which are nodes positioned directly above the flight access node (i.e. inheriting the flight access node's $x$ and $y$ values, yet increasing the $z$ value for altitude). Flight level nodes represent the transition point from vertical flight (VTOL) to horizontal (cruising) flight and connect the UAM station to the wider UAM network that is now modeled similarly to road networks.

**B. UAM Operation Modeling**

UAM vehicles are dynamically routed and dispatched to UAM stations during the simulation. Upon an agent's departure from an activity/facility, for which a newly developed `DepartureHandler` is registered, the agent requests a UAM vehicle. Currently, upon receiving a vehicle request, the dispatcher assigns the first-available, closest UAM vehicle to the requesting agent. During the agent's access leg, i.e. the agent's travel to the departure UAM station, the assigned UAM vehicle is being sent to the departure UAM station if necessary. If no vehicle is available, the `UAMDispatcher` puts the request into a queue and checks at each simulation time step whether some UAM vehicle has become available. In case the agent arrives at the departure UAM station before the UAM vehicle does, the agent will wait until the assigned vehicle has landed. On the other hand, if the vehicle arrives before the agent, the vehicle will also wait for the agent to board the UAM vehicle before departing to the destination station.

The registered `DepartureHandler` utilizes DVRP's `PassengerEngine` [7], which provides a method to pre-book transportation for a certain place and time. While the current booking implementation is simplistic (first-available, nearest UAM vehicle), this implementation allows for pre-booking at any time during the simulation, thus, enabling the development of more elaborate booking, assignment, and dispatching algorithms in later development stages.

At the beginning of a simulation, all UAM vehicles are added by utilizing the DVRP package [7], which allows the UAM vehicles to be simulated on the network and be dynamically routed. Each vehicle is initialized with a schedule (i.e. tasks the vehicle has to perform) that contains a continuous `UAMStayTask`, a task which instructs the vehicle to stay at its current UAM station and await potential vehicle assignments. Together with the `UAMStayTask`, the following tasks have been defined, which can be part of a UAM vehicle's schedule:

- `UAMStayTask` (vehicle has to stay at its current link, from defined start and end times)
- `UAMFlyTask` (extends DVRP's `DriveTask` and contains the flight path and the set of requests the vehicle has to serve)
- `UAMDropoffTask` (contains the start and end times and location (network link) of passenger drop-off)
- `UAMPickupTask` (contains the start and end times and location (network link) of passenger pick-up)

The schedules of UAM vehicles are never-ending, meaning that the vehicles are never removed from the simulation, as that is the default DVRP behavior for vehicle that have fulfilled their schedule. Thus, the last task of any UAM vehicle schedule is always a non-ending `UAMStayTask`. Upon being requested, a schedule task is appended to the assigned UAM vehicle's schedule. The schedule is populated with a `UAMFlyTask` to the originating or pick-up UAM station (if necessary), a `UAMPickupTask` at the originating UAM station, a `UAMFlyTask` to the destination UAM station, and a `UAMDropoffTask` at the destination station. Accordingly, the existing `UAMStayTask` is updated in order to reflect the current expected start time, and an additional `UAMStayTask` at the destination station concludes the schedule.

Passenger pooling within UAM vehicles can also be enabled in the current simulation version. Agents, who have the same origin and destination UAM stations, can be pooled into the same UAM vehicle as long as the vehicle has spare passenger capacity. In its current implementation, the UAM vehicle accepts all agent's requests for the same origin and destination pair from the moment of the first passenger's request until the `UAMPickupTask` commences at the pick-up UAM station.
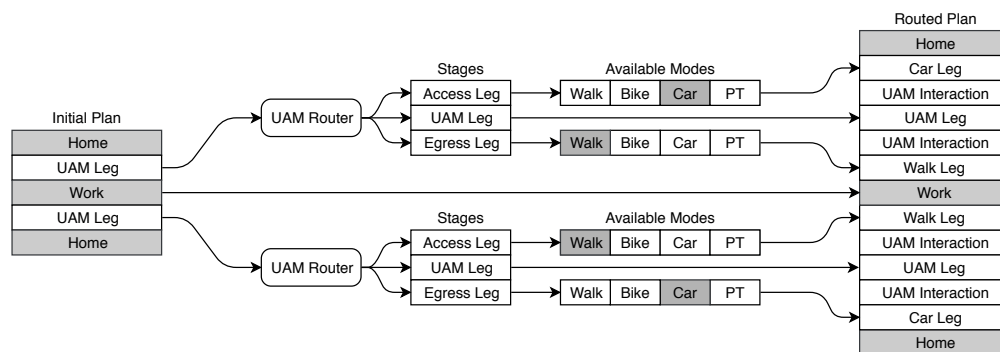


**Fig. 4   Plan modification by UAM inter-modal router, adapted from Rieser [12, p. 88]**

While the schedule task describes sections in the time plan of UAM vehicles, their routing is calculated within the re-planning phase of the simulation. For that, all stations in a pre-defined radius (see List. 1) around the agent's current

location and target destination are selected as potential starting and ending UAM stations. Using Dijkstra algorithm, paths between all combinations of origin and destination UAM stations are retrieved. As indicated in Fig. 4, there are currently four UAM station access and egress alternatives for agents to use: walk, bike, car/taxi, and public transport (PT). The origin-destination pairs are combined with estimated access and egress legs from and to the respective UAM stations. Based on travel time and cost, the agent decides for the desired UAM departure and arrival UAM station and its respective access and egress modes.

Figure 4 also illustrates the modification of agent's day plans by the UAM Router, where initial un-routed UAM legs are split into an access leg, a UAM leg (flight leg), and an egress leg. As described above, for both, the access and egress leg, the utilities of various transport modes are estimated to allow a suitable decision by the agent. The current UAM router implementation allows for different access and egress mode combinations. It might be the most suitable to, e.g., walk to the nearest UAM station but to take public transport to cover the distance from the destination UAM station to the agent's actual destination location. The potential attractiveness of various access and egress modes varies greatly with the placement of UAM stations, which might benefit certain modes over other, e.g. walking for UAM stations in residential areas or public transport for UAM station in close proximity to public transport hubs. The placement of UAM stations can be defined via the scenario configuration as explained hereafter.

**C. UAM Scenario Configuration**

MATSim, or rather the desired MATSim scenario, requires three additions to enable UAM usage within said scenario: (1) the addition of a UAM module within the scenario configuration, (2) the addition of a UAM infrastructure and vehicle definition file, and (3) the addition of UAM stations and flight links to the scenario network. Naming, structure, and formatting of parameter/XML-file additions are intended to follow the existing MATSim configuration's conventions for which Horni and Nagel [14] have been consulted.

*1. Addition of UAM Module in Scenario Configuration*

Listing 1, an excerpt from a UAM-enabled MATSim scenario configuration, provides an overview of the available parameters that can or must be defined within the MATSim scenario configuration file:

**Listing 1   Exemplary UAM module addition to scenario configuration**

```
<module name="uam" >
  <!-- Link to UAM input file containing UAM station and vehicle definitions
      -->
  <param name="inputUAMFile" value="uam.xml" />

  <!-- List of transport modes available for UAM station access/egress  -->
  <param name="availableAccessModes" value="walk,bike" />

  <!-- Duration for a passenger to (de-)board a UAM vehicle in [s] -->
  <param name="pickupDuration" value="30.0" />

  <!-- Number of parallel UAM inter-modal router threads  -->
  <param name="parallelRouters" value="2" />

  <!-- UAM stations within this radius around origin/destination are available
      for UAM leg routing in [m] -->
  <param name="searchRadius" value="5000.0" />
</module>
```

While the displayed values for both required parameters (`inputUAMFile` and `availableAccessModes`) are exemplary, the values of the optional parameters (`pickupDuration`, `parallelRouters`, and `searchRadius`) represent the parameters' default values. While there are only the pre-defined options of `walk`, `bike`, `car`, and `pt` for `availableAccessModes` to chose from in the current UAM extension, `inputUAMFile` can freely be chosen to point to the station and and vehicle definitions file either via a relative or absolute path.

*2. Addition of UAM Infrastructure and Vehicle Definition File*

The UAM definition file, as illustrated in List. 2, provides basic information on the placement and landing capacity of UAM stations, as well as the number and properties of vehicles:

### Listing 2  Exemplary UAM definition file

```xml
<?xml version="1.0" encoding="utf-8"?>
<!DOCTYPE uam SYSTEM "src/main/resources/dtd/uam.dtd">
<uam>
  <stations>
    <!-- UAM stations with identifiers, landing capacity, and connected
      network link -->
    <station id="01" landingcap="1" link="uam_sl-uam_st_01_ga-uam_st_01_fa" />
    <station id="02" landingcap="1" link="uam_sl-uam_st_02_ga-uam_st_02_fa" />
    <!-- [...] -->
  </stations>
  <vehicles>
    <!-- UAM vehilces with identifiers, passenger capacity, start/end times,
      cruise and vertical speed, and initial UAM station -->
    <vehicle id="01" capacity="2" starttime="00:00:00" endtime="23:59:59"
      cruisespeed="42" verticalspeed="10" initialstation="01" />
    <vehicle id="02" capacity="2" starttime="00:00:00" endtime="23:59:59"
      cruisespeed="42" verticalspeed="10" initialstation="02" />
    <!-- [...] -->
  </vehicles>
</uam>
```

The infrastructure and vehicle definition file in XML-format contains the encompassing <uam> element which holds the <stations> and <vehicles> elements. Both of which serve as collections for the actual UAM stations and vehicles. The provided station properties reflect the before mentioned UAM station characteristics from Sec. III.A, with a unique identifier, a vehicle landing capacity, and its location in the form of the closest network link. The exemplary link naming follows the logic network modeling logic as illustrated in Fig. 3, in which the UAM stations are linked to dedicated station links (_sl), which connect the station's ground (_ga) and flight (_fa) access nodes. Similarly, the vehicles also provide the before mentioned properties of a unique identifier, passenger capacity, start and end times, cruising speed, vertical speed, and the vehicle's initial starting station. The exemplary speeds above are provided in meters per second. The number of stations and vehicles can be set by adding or removing station or vehicle elements.

*3. Addition of UAM Stations and Flight Links to Scenario Network*

Finally, the conventional MATSim scenario's network has to be extended with UAM nodes and links. Listing 3 shows exemplary additions of UAM nodes and links in the, otherwise, conventional network structure. For clarity, road nodes and links have been omitted from List. 3.

### Listing 3  Exemplary network file with UAM nodes and links

```xml
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE network SYSTEM "http://www.matsim.org/files/dtd/network_v1.dtd">
<network>
  <nodes>
    <!--- UAM nodes with identifiers and x, y, and z (height) coordinates-->
    <node id="uam_st_01_ga" x="10" y="5" z="0" />
    <node id="uam_st_01_fa" x="10" y="5" z="0" />
    <node id="uam_st_01_fl" x="10" y="5" z="500" />
    <node id="uam_st_02_ga" x="20" y="0" z="0" />
    <node id="uam_st_02_fa" x="20" y="0" z="0" />
    <node id="uam_st_02_fl" x="20" y="0" z="500" />
    <!-- [...] -->
```

7

```xml
    <!--- Road nodes (conventional MATSim road nodes) -->
    <!-- [...] -->
  </nodes>
  <links>
    <!--- UAM links with identifier, origin and destination nodes, length,
        hourly throughput, free-flowing speed limit, number of parallel lanes,
        and allowed transport modes -->
    <link id="uam_gl-road_01-uam_st_01" from="road_01" to="uam_st_01_ga"
        length="50" capacity="999" freespeed="10" permlanes="1" modes="car" />
    <link id="uam_gl-uam_st_01_ga-road_01" to="road_01" from="uam_st_01_ga"
        length="50" capacity="999" freespeed="10" permlanes="1" modes="car" />
    <link id="uam_sl-uam_st_01_ga-uam_st_01_fa" from="uam_st_01_ga" to="
        uam_st_01_fa" length="1" capacity="999" freespeed="10" permlanes="1"
        modes="car,uam" />
    <link id="uam_sl-uam_st_01_fa-uam_st_01_ga" to="uam_st_01_ga" from="
        uam_st_01_fa" length="1" capacity="999" freespeed="10" permlanes="1"
        modes="car,uam" />
    <link id="uam_vl-uam_st_01_fa-uam_st_01_fl" from="uam_st_01_fa" to="
        uam_st_01_fl" length="500" capacity="999" freespeed="99" permlanes="1"
         modes="uam" />
    <link id="uam_vl-uam_st_01_fl-uam_st_01_fa" to="uam_st_01_fa" from="
        uam_st_01_fl" length="500" capacity="999" freespeed="99" permlanes="1"
         modes="uam" />
    <link id="uam_hl-uam_st_01_fl-uam_st_02_fl" from="uam_st_01_fl" to="
        uam_st_02_fl" length="2000" capacity="999" freespeed="99" permlanes="1
        " modes="uam" />
    <link id="uam_hl-uam_st_02_fl-uam_st_01_fl" to="uam_st_01_fl" from="
        uam_st_02_fl" length="2000" capacity="999" freespeed="99" permlanes="1
        " modes="uam" />
    <!-- [...] -->

    <!--- Road links (conventional MATSim road links) -->
    <!-- [...] -->
  </links>
</network>
```

In the shown example, two UAM stations have been defined, each consisting of three nodes: a ground access node (_ga), a flight access node (_fa), and one flight level (_fl) (c.f. Fig. 3). Multiple flight levels could be defined with additional nodes on different heights (i.e. z values). All nodes that–combined–make up a single UAM station have been defined with equal x and y values.

With regards to UAM links, the above example, List. 3., first illustrated the connection of the UAM station's ground access node to and from a conventional road network node via ground links (uam_gl). These two links are followed by the station links (uam_sl) in both directions between the ground and flight access nodes, one of which is used for UAM stations to attach to. Thereafter follow the vertical flight link from the flight access to the flight level node (uam_vl), to enable vehicles to start and land vertically. Finally, the flight level nodes are connected to other flight level nodes via horizontal flight links (uam_hl) to create a fully connected UAM network. In its current implementation, the link identifiers are used for deciding whether a UAM vehicle has to traverse it vertically or horizontally.

With the described methodology of UAM network encoding, different types of network structures can already be defined and, thus, simulated. Fig. 5 shows two exemplary types of UAM networks which have been set up for the MATSim scenario Sioux Falls by Hörl [15]. Fig. 5a represents a direct, point-to-point network structure–where each UAM station is directly connected to each other station–, whereas Fig. 5b represents an indirect, centralized network in which each UAM flight has to pass a central node. While the latter is an impractical scenario, it shows the versatility of MATSim's network configuration and already allows to model no-fly-zones, height restrictions, or desired flight paths along, e.g., rivers or highways.
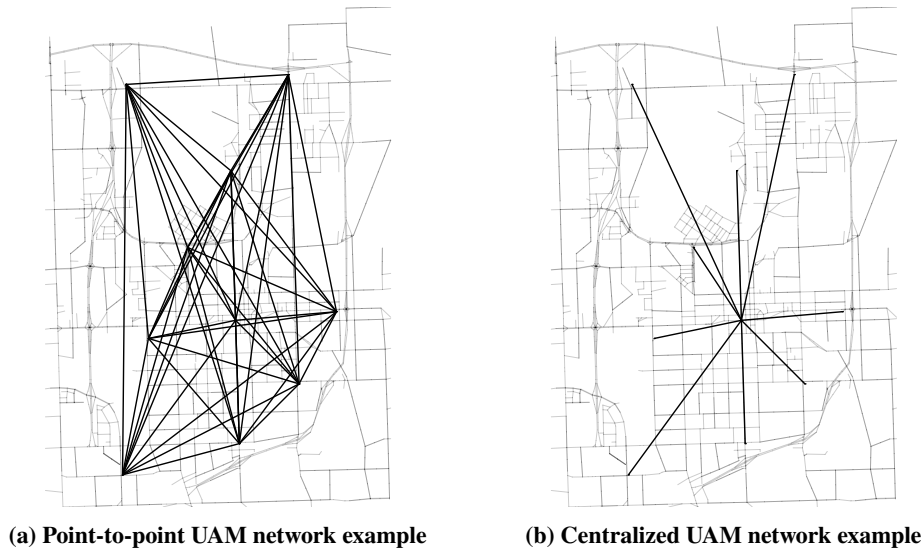
(a) Point-to-point UAM network example      (b) Centralized UAM network example

**Fig. 5   Illustration of different UAM network structures for Sioux Falls**

## D. Further Extension

The UAM extension in its current implementation uses–partly drastic–simplifications, such as the mutual exclusive flight behavior of either vertical or horizontal flight or vehicle distribution without any optimization towards ensuring balanced vehicle availability. Thus, continuous development is planned to enhance the current UAM extension version with additional and more elaborate features such as:

- Modeling of charging and refueling: UAM vehicles are not only bound to a certain maximum mission range, yet also to recharging/refueling–depending on their energy source for propulsion. The need and time required for charging/refueling should be considered in future implementations.
- Elaborate vehicle-sharing algorithm: Currently, pooling follows a simple logic of combining similar UAM transport requests that occur within a short period of time. In later version, intelligent pooling should be provided where various passenger requests can be combined with differing origin and destination stations, where passengers enter/leave the UAM vehicle at, e.g., intermediary stops.
- Elaborate vehicle distribution algorithm: While unused UAM vehicle currently await new assignments at the destination station of their last trip, an algorithm should be implemented that anticipates demand and redistributes vehicles to stations that are expected to experience vehicle capacity bottlenecks.
- Elaborate station capacities: While landing capacities can already be defined, more parameters are required to reflect the different types and sizes of UAM infrastructure, e.g. passenger throughput, parking capacities for UAM vehicles and cars, different mode-dependent access times. Thus, UAM infrastructure modeling is to be enhanced in further development.
- Differentiating UAM pricing: Currently, usage of all UAM vehicle is priced the same. The current ability to define different UAM vehicle types will only be able to have realistic influence on passenger decisions if pricing can be different for each vehicle based, e.g., on the vehicles speed and capacity.

While the above-listed items provide an overview of the major planned additions, the list is absolutely not exhaustive. There are many more intricacies of UAM and its integration into existing transport systems that merit their own consideration, development, and implementation.

## IV. Conclusion

The documentation of the UAM extension for MATSim and its methodology for simulating the use of on-demand VTOL vehicle in urban transport environments, so-called Urban Air Mobility, is being presented. UAM simulation requires modeling of UAM infrastructure, its vehicles, network, and operations. Building upon the existing DVRP extension for MATSim, an early UAM extension can be outlined and explained in its functioning. While the presented

methodology still provides plenty opportunities for improvement and enhancement, the UAM extension in its early version already provides the ability to define UAM station and vehicles to identify UAM's interrelation with existing transport types in MATSim.

## Funding Sources

## References

[1] Shamiyeh, M., Bijewitz, J., and Hornung, M., "A Review of Recent Personal Air Vehicle Concepts," Bucharest, Romania, 2017.

[2] Horni, A., Nagel, K., and Axhausen, K. W., *The Multi-Agent Transport Simulation MATSim*, Ubiquity Press, 2016. doi: 10.5334/baw, URL `http://www.ubiquitypress.com/site/books/10.5334/baw/`.

[3] Balac, M., Vetrella, A. R., and Axhausen, K. W., "Towards the Integration of Aerial Transportation in Urban Settings," *97th Annual Meeting of the Transportation Research Board (TRB 2018)*, 2018.

[4] Grether, D. S., "Extension of a Multi-Agent Transport Simulation for Traffic Signal Control and Air Transport Systems," Dissertation, Technischen Universität Berlin, Berlin, 2014.

[5] Hörl, S., "Agent-Based Simulation of Autonomous Taxi Services with Dynamic Demand Responses," *Procedia Computer Science*, Vol. 109, 2017, pp. 899–904. doi:10.1016/j.procs.2017.05.418, URL `http://linkinghub.elsevier.com/retrieve/pii/S1877050917310943`.

[6] Bösch, P., Ciari, F., and Axhausen, K., "Autonomous Vehicle Fleet Sizes Required to Serve Different Levels of Demand," *Transportation Research Record: Journal of the Transportation Research Board*, Vol. 2542, 2016, pp. 111–119.

[7] Maciejewski, M., "Chapter 23: Dynamic Transport Services," *The Multi-Agent Transport Simulation MATSim*, edited by A. Horni, K. Nagel, and K. W. Axhausen, Ubiquity Press, 2016, pp. 145–152. doi:10.5334/baw, URL `http://www.ubiquitypress.com/site/books/10.5334/baw/`.

[8] Ziemke, D., Nagel, K., and Moeckel, R., "Towards an Agent-Based, Integrated Land-Use Transport Modeling System," *Procedia Computer Science*, Vol. 83, 2016, pp. 958–963. doi:10.1016/j.procs.2016.04.192, URL `http://linkinghub.elsevier.com/retrieve/pii/S1877050916302253`.

[9] Zilske, M., "Chapter 45: How to Write Your Own Extensions and Possibly Contribute Them to MATSim," *The Multi-Agent Transport Simulation MATSim*, edited by A. Horni, K. Nagel, and K. W. Axhausen, Ubiquity Press, 2016, pp. 297–304. doi:10.5334/baw, URL `http://www.ubiquitypress.com/site/books/10.5334/baw/`.

[10] Fu, M., "Exploring Preferences for Transportation Modes in an Urban Air Mobility Environment - a Munich Case Study," Master's Thesis, Technical University of Munich, Munich, 2018.

[11] Fadhil, D. N., "A GIS-Based Analysis for Selecting Ground Infrastructure Locations for Urban Air Mobility," Master's Thesis, Technical University of Munich, Munich, 2018.

[12] Rieser, M., "Adding Transit to an Agent-Based Transportation Simulation: Concepts and Implementation," Dissertation, Technischen Universität Berlin, Berlin, 2010.

[13] Rieser, M., Horni, A., and Kai, N., "Chapter 2: Let's Get Started," *The Multi-Agent Transport Simulation MATSim*, edited by A. Horni, K. Nagel, and K. W. Axhausen, Ubiquity Press, 2016, pp. 35–44. doi:10.5334/baw, URL `http://www.ubiquitypress.com/site/books/10.5334/baw/`.

[14] Horni, A., and Nagel, K., "Chapter 4: More About Configuring MATSim," *The Multi-Agent Transport Simulation MATSim*, edited by A. Horni, K. Nagel, and K. W. Axhausen, Ubiquity Press, 2016, pp. 35–44. doi:10.5334/baw, URL `http://www.ubiquitypress.com/site/books/10.5334/baw/`.

[15] Hörl, S., "Implementation of an Autonomous Taxi Service in a Multi-Modal Traffic Simulation Using MATSim," , 2016. doi:10.13140/rg.2.1.2060.9523.