

SUBPART FOUR

Other Modes Besides Individual Car

CHAPTER 16

Modeling Public Transport with MATSim

Marcel Rieser

16.1 Basic Information

Entry point to documentation:

<http://matsim.org/extensions> → pt

Invoking the module:

The module is invoked by enabling it in the configuration.

Selected publications:

Rieser (2010)

16.2 Introduction

Public transport—or *transit* as it is sometimes called—plays an important role in many transport planning measures, even those initially targeting only non-transit modes. By making other modes more or less attractive (e.g., by providing higher capacity with additional lanes, allowing higher speeds, or charging money by setting up area road pricing), travelers might reconsider their mode choice and switch to public transport (*pt*) from other modes, or vice versa. Such changes can also occur when transit infrastructure is changed; additional bus lines, changed tram routes with different stops served, or altered headways—all are important for travelers on specific lines, or public transport in general. Around 2007, interest grew in extending MATSim to support detailed simulation of modes other than private car traffic, particularly public transport.

In a first step, MATSim was extended so that modes other than *car* would be teleported; agents would be removed from one location and placed at a later point of time—corresponding to estimated travel time—at their destination location, where they could commence their next activity. Together with a simple mode-choice module, randomly replacing all transport modes in all plan

How to cite this book chapter:

Rieser, M. 2016. Modeling Public Transport with MATSim. In: Horni, A, Nagel, K and Axhausen, K W. (eds.) *The Multi-Agent Transport Simulation MATSim*, Pp. 105–110. London: Ubiquity Press. DOI: <http://dx.doi.org/10.5334/baw.16>. License: CC-BY 4.0

legs and a simple travel time estimation for modes different than *car*, first case studies resulting in modal share changes were performed using MATSim (Rieser et al., 2009; Grether et al., 2009). This teleportation mode is now available, by default, in MATSim and still a very good fallback option to get a multimodal scenario up and running with as little data as possible.

In a second step, QSim was extended to support detailed simulation of public transport vehicles serving stops along fixed routes with a given schedule (Rieser, 2010). The next section describes, in more detail, data required and resulting features for this detailed public transport simulation.

16.3 Data Model and Simulation Features

MATSim supports very detailed modeling public transport; transit vehicles run along the defined transit line routes, picking up and dropping off passengers at stop locations, while monitoring transit vehicles' capacities and maximum speeds. Data used to simulate public transport in MATSim can be split in three parts:

- stop locations,
- schedule, defining lines, routes and departures, and
- vehicles.

This data is stored in two files; vehicles are defined in one file, stop locations and schedule in another. Examples of such files can be seen in Section 16.4.1 and Section 16.4.2, respectively.

The data model is comparable to other public transport planning software, but simplified in several respects. A line typically has two or more routes; one for each direction and additional routes when vehicles start (or end) their service at some point on the full route (coming from, or going to, a depot). Each transit route contains a network route, specifying on which network links the transit vehicle drives, as well as a list of departures, providing information about what time a vehicle starts at the first route stop. A route also includes an ordered list of stops served, along with timing information specifying when a vehicle arrives or leaves a stop. This timing information is given as offsets only, to be added to departure time at the first stop. Each departure contains the time when a vehicle starts the route and a reference to the vehicle running this service. Because timing information is part of the route, routes with the same stops sequence may exist, differing only in time offsets. This is often the case with bus lines, that take traffic congestion and longer rush hour waiting times at stops into account in the schedule.

Stop locations are described by their coordinates and an optional name; they must be assigned to exactly one line of the network for the simulation. Thus, they can be best compared to “stop points” in VISUM. There is, currently, no logical grouping of stop locations to build a “stop area”; this is a cluster of stops often sharing the same name, but located on different intersection arms, served by different lines, many with transfer corridors for passengers.

Each vehicle belongs to one vehicle ‘type’, which describes various characteristics, like seating and standing capacity (number of passengers), its maximum speed and how many passengers can board or depart a vehicle per second.

This data model already supports several advanced public transport modeling aspects: varying travel speeds along routes during different times of day (important for improved simulation realism), using diverse vehicle types on routes at different times of day (interesting for schedule economic analysis) and re-using transit vehicles for multiple headways along one or different routes (allows vehicle deployment planning optimization, or research on delay-propagation effects).

With these data sets, the QSim will simulate all transit vehicle movements. The vehicles will start with their first route stop at the given departure time, allow passengers to enter and then drive along their route, serving stops. At each stop, passengers can enter or leave the vehicle. The simulation generates additional, transit-related events whenever a transit vehicle arrives or departs at a stop, when passengers enter or leave a vehicle, but also when a passenger cannot board a vehicle because

its capacity limit is already reached. This allows for detailed analyses of MATSim's public transport simulations.

For passengers to use public transport in MATSim, they must be able to calculate a route using transit services. For this, MATSim includes a public transport router that calculates the best route to the desired destination with minimal cost, given a departure time. Costs are typically defined only as travel time and a small penalty for changing lines, but other, more complex cost functions could be used.

The routing algorithm is based on Dijkstra's shortest path algorithm (Dijkstra, 1959), but modified to take multiple possible transit stops, around the start and end coordinates, into account to find a route. Multiple start and end stops must be considered to generate more realistic transit routes; otherwise, agents could be forced to travel first in the wrong direction, or wait at an infrequently served bus stop, instead of going a bit further to a busy subway stop location. By modifying the shortest path algorithm to work with multiple start and end locations, a considerable performance gain was achieved when compared to the basic (and somewhat naive) implementation that calculated a route for each combination of start/end location and then chose the best outcome.

16.4 File formats

16.4.1 transitVehicles.xml

To simulate public transport in MATSim, two additional input files are necessary. One is `transitVehicles.xml`, which describes vehicles serving the lines: big buses, small buses, trains or light rail vehicles and description of each vehicle's passenger transport capacity.

Public transport vehicle description can be split into two parts; first, vehicle types must be described, specifying how many passengers a vehicle can transport (Note that the term "vehicle" can refer to multiple vehicles in reality, e.g., a train with several wagons should be specified as one long vehicle with many seats). Second, actual vehicles must be listed. Each vehicle has an identifier and is a previously specified vehicle type. The following shows an example of a such a file, describing one vehicle and two vehicles of the same type.

```
<?xml version="1.0" encoding="UTF-8"?>
<vehicleDefinitions xmlns="http://www.matsim.org/files/dtd"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://www.matsim.org/files/dtd
    http://www.matsim.org/files/dtd/
      vehicleDefinitions_v1.0.xsd">
  <vehicleType id="1">
    <description>Small Train</description>
    <capacity>
      <seats persons="50"/>
      <standingRoom persons="30"/>
    </capacity>
    <length meter="50.0"/>
  </vehicleType>
  <vehicle id="tr_1" type="1"/>
  <vehicle id="tr_2" type="1"/>
</vehicleDefinitions>
```

16.4.2 transitSchedule.xml

The second, rather complex, file necessary to simulate public transport is `transitSchedule.xml`, containing information about stop facilities (bus stops, train stations, or other stop locations) and transit services.

In the first part, stop facilities must be defined; each one is given a coordinate, an identifier and a reference to a network link. The stop can only be served by vehicles driving on that specified link. It is also possible to specify both a name for the stop and whether other vehicles are blocked when a transit vehicle halts at a stop. This last attribute is useful when modeling e.g., different bus stops, where one has a bay, while at another, the bus must stop on the road.

After stop facilities, transit lines, their routes and schedules are described. This is a hierarchical data structure; each line can have one or more routes, each with a route profile, network route and list of departures. The following listing is an example of a basic, but complete transit schedule.

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE transitSchedule SYSTEM "http://www.matsim.org/files/dtd/
transitSchedule_v1.dtd">

<transitSchedule>
  <transitStops>
    <stopFacility id="1" x="990.0" y="0.0" name="Adorf"
      linkRefId="1" isBlocking="false"/>
    <stopFacility id="2" x="1100.0" y="980.0" name="Beweiler"
      linkRefId="2" isBlocking="true"/>
    <stopFacility id="3" x="0.0" y="10.0" name="Cestadt"
      linkRefId="3" isBlocking="false"/>
  </transitStops>
  <transitLine id="Blue Line">
    <transitRoute id="1">
      <description>Just a comment.</description>
      <transportMode>bus</transportMode>
      <routeProfile>
        <stop refId="1" departureOffset="00:00:00"/>
        <stop refId="2" arrivalOffset="00:02:30"
          departureOffset="00:03:00"
          awaitDeparture="true"/>
        <stop refId="3" arrivalOffset="00:05:00"
          awaitDeparture="true"/>
      </routeProfile>
      <route>
        <link refId="1"/>
        <link refId="2"/>
        <link refId="3"/>
      </route>
      <departures>
        <departure id="1" departureTime="07:00:00"
          vehicleRefId="12"/>
        <departure id="2" departureTime="07:05:00"
          vehicleRefId="23"/>
        <departure id="3" departureTime="07:10:00"
          vehicleRefId="34"/>
      </departures>
    </transitRoute>
  </transitLine>
</transitSchedule>
```

Each transit line must have a unique ID and each transit route has an ID, which must be unique within that one line, allowing the same route ID to be used with different lines. The transportMode describes network links where the line runs. (Actually, this is not yet in force, although it might be in the future. It would be possible to let a bus run on train links in the simulation.)

The routeProfile describes the stops this route serves; the route itself describes the series of network links the transit vehicle's driver must navigate, often referred to as network route. Note that the complete route, i.e., all links the vehicle traverses, must be listed in the route, not only those with stops. All specified stops should occur along this route in correct order. Time offsets given for each stop in the routeProfile describe relative time offsets to an actual departure time. If a bus departs at 7 am, and stop 2 has a departureOffset of 3 minutes, this must be read that the

bus is expected to depart at 7:03 am from the specific stop. All stops in the route profile must have a departure offset defined, except the last one. All stops, except the first one, can, optionally, have an arrival offset defined. This is useful for large trains that stop for several minutes at a station; helping the routing algorithm find connecting services at the correct time, namely the expected train arrival time.

As the last part of a transit route description, a departures list should be given. Each departure has an ID, which must be unique within the route, giving the departure time at the first stop of the specified route profile. The departure also specifies the vehicle (which must be defined in the previous transit vehicle list) with which the service should be run.

Because of its complexity, transit schedules often contain small mistakes that will return in an error when the simulation runs. Typical examples include: missing links in the network route, or incorrect defined stop order on the network route. To ensure a schedule avoids such issues before the simulation starts, a special validation routine is available:

```
java -Xmx512m -cp /path/to/matsim.jar
    org.matsim.pt.utils.TransitScheduleValidator
    /path/to/transitSchedule.xml /path/to/network.xml
```

If run, this validator will print out a list of errors or warnings, if any are found, or show a message that the schedule appears to be valid.

16.5 Possible Improvements

While the ability to simulate public transport was a big advance for MATSim, several shortcomings still require attention:

- The data model (and thus, the simulation) does not yet fully support some real world transit lines: for example, circular lines with no defined start and end cannot yet be easily modeled. Some bus or train lines also have stops where only boarding or alighting the vehicle is allowed, but not both (e.g., overnight trains with sleeper cabins). At the moment, MATSim always allows boarding and alighting at stops, leading to agents e.g., using a train with sleeper cabins for a short trip; in reality, they would be denied boarding without a reservation for a longer trip.
- A stop location, as seen by passengers in the real world, is typically modeled as a number of stop facilities in MATSim, detailing different locations where transit vehicles stop (depending on their route and direction). For analysis, one is often interested in aggregated values for such logical stop locations, not for individual stop facilities. Such a logical grouping is still missing in MATSim data format.
- Running simulations with a reduced population sample leads to artifacts when public transport is used. In a simulation with a sampled demand, network capacity is reduced accordingly, to accommodate the fact that fewer private cars are on the road. But because 100 % of public transport vehicles must run (albeit with reduced passenger capacity), calibration becomes difficult. This should be solved, in the future, not by reducing network capacity, but by giving each vehicle and agent a weighting, specifying how much each should count.
- The public transport router available and used by MATSim by default is strictly schedule-based. It assumes vehicles can keep up with the schedule and that enough passenger capacity is provided. In some regions, where transit is chronically delayed and overcrowded, MATSim's router will consistently advise agents to use routes that will perform badly in the simulation. Additional feedback from the simulation back to the router, as already done in the MATSim private car router, will be needed.

- Last, but not least, the current router, based on a modified shortest path algorithm of Dijkstra, can become rather slow and memory-intensive for larger areas with extensive transit offerings. Improved algorithms to generate the routing graph, or different routing algorithms altogether (like the non-graph based Connection Scan Algorithm (Dibbelt et al., 2013)) must be explored in the future.

16.6 Applications

Public transport simulation has been used in myriad applications of MATSim world-wide. The following list highlights some of these applications, pinpointing their special public transport simulation features.

- Berlin: the Berlin scenario (see Chapter 53) was one of the first real applications using public transport simulation in MATSim. The road and rail network, as well as the full transit schedule, was converted from a VISUM model. It is still one of the few known models where bus and tram lines share a common network with private car traffic, enabling full interaction between private and public vehicles (like transit vehicles) getting stuck and delayed in traffic jams.
- Switzerland: Senozon AG maintains a model of Switzerland containing the full timetable of all buses, trams, trains, ships, and even cable cars, in the Swiss alps. The schedule data is retrieved from the official timetable, available in a machine-readable format called “HAFAS (HaCon Fahrplan-Auskunfts-System) raw data format”.
- Singapore: The model of Singapore (see Chapter 57) makes heavy use of public transport, and continually pushes the boundaries of what is currently possible to simulate. Due to the very large number of buses on Singapore’s roads and strong demand for public transport, many extensions had to be implemented to realistically model pt in this context.
- Minibus: The minibus contribution (see Chapter 17) added an optimization layer to public transport functionality in MATSim, allowing automatic generation of an optimized transit schedule for a specific region.
- WagonSim: In the WagonSim contribution (see Chapter 25) public transport simulation was used to simulate rail-bound freight traffic. While the simulation was still moving around transit vehicles and letting passengers enter and leave these vehicles, the scenario had been customized so that vehicles corresponded to freight trains and passengers corresponded to actual goods being transported. Custom implementations of transit driver logic replaced vehicle capacity definition by an alternative definition, ensuring that the trains vehicles represent did not get too long or heavy. The network was constructed so that changing vehicles at stops took minimum time, corresponding to the time needed for switching wagons at freight terminals.

In addition to applications mentioned in the list above, many additional scenarios now use public transport simulation in MATSim. Importantly, the list also shows, that with some custom extensions and imagination, public transport functionality can be used for far more than “just simulating public transport”; it can be employed to solve complex problems previously handled by operations research groups.