A Bi-Modal Simulation Model to Increase the Resilience of Public Transit Networks

Oliver Ullrich, Daniel Lückerath

Fraunhofer Institute for Intelligent Analysis and Information Systems IAIS, Schloss Birlinghoven, 53757 Sankt Augustin, Germany; *oliver.ullrich@iais.fraunhofer.de; daniel.lueckerath@iais.fraunhofer.de*

SNE 32(2), 2022, 103-112, DOI: 10.11128/sne.32.tn.10607
Received: 2021-07-21, Revised: 2021-10-10;
Accepted: 2021-12-15
SNE - Simulation Notes Europe, ARGESIM Publisher Vienna
ISSN Print 2305-9974, Online 2306-0271, www.sne-journal.org

Abstract. Public service infrastructure will increasingly be impacted by climate change and has therefore to become more resilient against extreme weather events and other climate change-related effects. A central part of urban infrastructure is public transit, often mainly consisting of interacting light-rail as well as express and community bus networks. To increase such a system's resilience against small disturbances and larger outages – as they might result from climate change – service providers need a toolbox of potential measures to mitigate such incidents' impact and to re-establish services as soon as possible after an outage. This paper presents a bi-modal urban transit simulation system covering both light rail and (express and community) bus networks. Central aims of the system are to enable operators to evaluate measures against small disturbances and larger outages as they happen, and to evaluate what combination of disaster risk management and resilience-building strategies shows most potential to help increasing the resilience of urban transit systems against extreme weather events resulting from climate change as well as other disasters.

Introduction

To protect their long-term utility against the increasing impact of climate change, urban infrastructure components have to become more resilient against extreme weather events like pluvial and fluvial flooding, heat waves, draughts, and windstorms [1]. Such urban infrastructure includes all types of publically and privately operated communication, electricity, and water networks, waste treatment, industrial facilities, as well as urban transportation.

Over the last decades, infrastructure systems that were perceived up until then as isolated services have transformed to connected ecosystems; tightly organised networks provided by a multitude of actors, involving a myriad of physical and digital structures, and offering services to society through all sorts of physical and digital channels. That includes the different modes of urban transit, including light rail systems, express and community buses, and at least partially integrated individual transportation services like taxi cabs, Uber, and Lyft. Out of all commonly available public transit modes, light rail and bus transit have the highest transit performance [2].

In case of sudden disasters impacting transit systems, including extreme weather and human-made events, operators have to be able to make decisions fast to a) transfer the infrastructure components into a pre-planned disaster mode and b) to be able to re-establish services as soon as the immediate event has passed. These operators can be assisted with a simulation application covering both light rail and bus transit that executes simulation runs sufficiently fast to enable evaluation and comparison of potential decisions and strageties, thereby contributing to increase the resilience of the transit system.

This paper presents a bi-modal simulation model representing an urban area's integrated light rail and bus transit network, designed to assist with increasing the transit infrastructure system's resilience againt extreme weather events and human-made distasters. A specific focus is put on a) fast execution and b) the representation of operating decisions necessary in disaster risk management situations.

Many models representing urban transit are developed as an extension of already established models of individual traffic [3][4][5]. Many of the more recent simulation models including bus transit use microscopic agent-based modeling approaches [3][4][6][7], the mesoscopic approach to bus transit simulation proposed by Toledo et al. [5] extends a mesoscopic simulation model for individual traffic based on queuing theory proposed by Burghout in [8], which represents the street network as a graph of interconnected queues and vehicles as individual entities traversing these queues based on speed/density functions.

Especially models utilizing a fine-grained modeling approach generally necessitate the availability of an extensive data basis, including detailed information on origin-destination matrices, vehicular dynamics, signaling strategies, and lane changing rules [9], and include many components which are not immediately interesting for public transit resilience management. This often leads to long runtimes [10][11], thereby rendering those models inadequate for the use case described above. Therefore, this paper builds upon the work presented by Lückerath [12] by extending a runtime-efficient bus transit model to include light rail transit.

The paper continues by sharing some background on the core components and concepts of urban transit systems (Section 1) and then introduces the developed bi-modal transit simulation model (Section 2). It examines a first round of experiments, specifically concerning execution speed (Section 3), and concludes with an outlook on necessary further research steps (Section 4).

1 Urban Transit Components

Urban transit usually consists of a number of interacting networks, e.g., a light rail system, express and community bus networks, often connected at specific hubs to national rail systems as well as to individual transit systems like taxi cabs, Uber, or Lyft. For the presented model individual transit as well as national rail stations/airports are parameterized and not part of the core model itself.

A mixed light-rail and bus network consists of a network of street and rail segments as well as stops and stations where passenger exchanges take place. These stops and stations are served by a set of transit vehicles executing service trips, i.e., pairings of starting times and sequences of stops, according to a timetable. Each individual vehicle executes several service trips, interspersed with deadhead trips, over the course of an operational day, which is called a rotation. Such a rotation usually begins with a deadhead trip from the vehicle's depot to the first stop of its first service trip and, after a number of service trips, ends with a returning deadhead trip to the depot. The vehicle schedule defines the assignment of specific vehicles to rotations.

While some stops, mainly bus stops, include a bay with capacity for more than one vehicle, many other stops can contain only one vehicle at any given time. Some stops are marked as control points, i.e., locations in the network where control strategies may be employed, e.g., purposely delaying early vehicles until the scheduled departure time is reached. At other stops, vehicles depart as soon as the passenger exchange is completed. Each stop belongs to exactly one station, a geographically grouped collection of stops which usually share a common name.

Directed paths through the network, connecting two successive stops are called connections. They usually consist of several street and/or rail segments, junctions, and signals, that in turn can be shared by several connections.

Signals control access to individual segments, usually at junctions. Often, two or more signals constitute a signal group with a common scheduling strategy.

Urban transit vehicles generally follow pre-defined line routes, consisting of sequences of stops to be serviced. Often, a line consists of a number of line variants: while a main variant might be served by a majority of vehicles, some variants might contain only stops in the city center but not in the suburbs or might branch off the trunk route to connect to a commercial area or business park.

In most public transit systems, daily operations are managed by an operations center, with dispatcher personnel managing procedures for the mitigation of small disturbances and larger outages. While the number and intensity of the smaller disturbances might increase from the impacts of climate change, e.g., changing precipitation patterns, many will originate from everyday incidents, like street segments blocked by accidents, or failing transit vehicle doors. Larger outages might result from extreme weather events, like pluvial or fluvial flooding, high storms, or excessive heat waves – or from human-caused events like protests or terrorist attacks. In case of any of these events, transit operators have a number of remedies at their disposal to keep services running as long as possible, and to restore them as soon as possible. These include the authority to short-turn or cancel trips, to re-route vehicles, and to deploy extra vehicles.

2 Modelling Urban Transit

To represent the described entities and behavior that constitute urban light-rail and bus networks, a bi-modal simulation model based on the event-based approach [13] has been designed.

A description of its design and mechanics are shared in the following sections, highlighting partial models representing physical network components, the logical network, vehicle behavior, operational management, and the necessary randomization. All of these partial models are based on the bus transit model described by Lückerath and Ullrich [14].

2.1 Physical Network Model

The basis of the model is the representation of the physical transit network as a directed graph. Stops, connections and segments are modeled as nodes of this graph, with their neighborhood relations modeled as edges. Each node has a geographic position, identifying attributes, and a maximum vehicle capacity. Furthermore, each node represents an entity in the sense of the event-based simulation paradigm, i.e., it can be producer and consumer of events. Thus, temporary changes of attribute characteristics, e.g., for modeling disruptions, can be mapped in a simple way via events and activities.

Stops are nodes in the model graph where transit vehicle entities – i.e., busses and light rail cars – stop for boarding and disembarking processes. They always belong to exactly one station and have time-of-day and location-specific stopping times. Different capacity or spatial dimensions of stops are modelled by a maximum number of vehicle entities they can service simultaniously. For example, at a larger stop within a bus station, several vehicles can usually stop at the same time. Whereas at a stop without a separate bay on a busy road, an arriving vehicle may have to wait for an already stopping vehicle before it can approach the stop.

Stations group together geographically related stops and give them a uniform name. They form an additional information layer within the model.

Connections are directed paths in the model graph that link two stops. They have a specific length as well as time of day and location specific average travel times. In addition, they are assigned a planned travel time by the timetable. Depending on the transit system to be modeled and the level of detail of the available data, connection nodes also manage model components of the segments, switches and signals belonging to their connection.

Segments represent subsections of connections, representing road or rail segments between two road junctions or between two switches of a rail transport system. Consequently, their corresponding model components have a specific length, a scheduled travel time and manage empirical data on their average travel time. In addition, they have an allowed maximum traversal speed, which can be used, e.g., for microscopic simulation of driving behavior. To represent overlaps between different connections, segment nodes can be part of several connection nodes.

To represent the driving behavior of different traffic modes, the model distinguishes between two types of segment nodes: roads and tracks. Road nodes are segment nodes that are used by entities of individual traffic, have an unrestricted vehicle capacity, and do not enforce a fixed vehicle sequence. Without (detailed) information about lanes, it is assumed that there is sufficient space for overtaking maneuvers on each road node, i.e., travel times of individual vehicle entities can be calculated without considering other entities traveling on the node. A more detailed modeling can be achieved by a simple extension of the model by a specialized road node (e.g., based on the modeling of the Mezzo system [5][8]).

In contrast to road nodes, track nodes are used exclusively by rail vehicle entities and enforce both compliance with a maximum vehicle capacity as well as a fixed vehicle sequence. The latter prevents inadmissible overtaking maneuvers between vehicle entities traveling on the same track node and is realized via the travel time calculation (see Section 2.5): If available, the entity traveling directly ahead is always considered to determine the travel time of a vehicle entity newly arriving on a track node. The calculated simulation time at which the new vehicle entity arrives at the end of the track node can never be earlier than that of the entity directly in front. Without possibilities for overtaking maneuvers, the formation of backlogs – even across neighboring nodes – is considered in the model using the vehicle capacity of nodes. The combined length of all vehicle entities driving on the track node at a given time must always be less than or equal to the track length. If the combined vehicle length would become greater than the track length due to a new entity arriving on the track node, the arriving vehicle entity must wait on the neighboring node upstream of the track node until sufficient space is available.

Switches are locations in rail-based transit systems where track crossings take place without interrupting the journey, i.e., they have a unique geographical position and are related to at least three tracks – at least one each incoming and outgoing.

They are modeled as transfer points without spatial extension and are traversed in zero time. Switches can merge several tracks and must be activated to target the correct incoming/outgoing track before an entity can cross them. This is represented in the model by vehicles reserving switches before crossing them and releasing them after a successful transfer.

Signals represent traffic lights of road traffic as well as light signal systems of rail traffic. They usually form a signal group with other signals and have attributes such as switching time or signal status (e.g., green, yellow, red).

Both switches and signals are modelled as additional information layer and not as nodes of the model graph. They can only be found at start or end positions of segment nodes in the model and can be associated with the corresponding nodes based on these positions.

2.2 Logical Network Model

In addition to the physical network components presented so far, logical components such as lines, trips and timetables have to also be considered to model public transit.

Lines consist of an ordered set of stops, which specifies the route to be followed during regular operation. In the simulation model, this is represented by a reference to a set of corresponding nodes of the model graph. To avoid time-consuming dynamic path finding during the simulation run, lines are additionally supplemented by an ordered set of connection nodes. Furthermore, each line can be assigned to a specific transit mode (e.g., bus or train) and may additionally only be served by vehicle types permitted for it. E.g., a low-floor train may not serve a line whose stops are designed for high-floor trains. Accordingly, lines in the model manage references to their transit mode and the vehicle types permitted for them.

Trips combine ordered sets of stops and connections with a start time and are differentiated into service trips and deadheads. In the model, trips manage references to sets of stop and connection nodes, similar to lines. Service trips additionally refer to the line they serve. Deadheads do not follow a predefined route and therefore do not refer to a line in the model.

2.3 Vehicles

Vehicles are represented as transient entities [13] that encapsulate a significant portion of the event-based simulation logic and move across the model graph during a simulation run. Each vehicle entity has a reference to the trip it is currently serving, i.e., at each simulation time it only has access to the information that is directly relevant for its current activity. All additional information, e.g., about the timetable and the vehicle fleet, is administered by specialized management modules (see Section 2.4).

In the model, vehicles are classified according to their transit mode, their vehicle type, and their individual vehicle characteristics. While transit mode discerns light-rail and bus vehicles, the vehicle type is used for a more detailed subdivision. For example, various types of Vossloh Kiepe GmbH vehicles are in use in the Cologne light rail network, including low-floor vehicles of type K4000 [15] and K4500 [16] and high-floor vehicles of type K5000 [17]. The most detailed classification is based on individual vehicle characteristics. They encapsulate attributes such as passenger capacity, vehicle length, maximum speed, minimum stopping time or boarding rate. This type of modeling allows the representation of vehicle entities of the same type with uniform equipment on the one hand, and on the other hand it allows the representation of disturbances of individual vehicles (e.g., an increased minimum stopping time due to a defective door).

A later extension of the simulation model by further traffic modes is possible in a simple way by creating new vehicle models derived from the generic vehicle model and providing them with individual vehicle characteristics.

Nine simulation event types represent the behavior of bus and light rail vehicles (see Table 1). For a detailed description of the bus-related simulation events see [14]. Figure 1 shows the relationships between the individual event and activity types for light-rail vehicles, based on the associated event process chains.

2.4 Operational Management

The model components presented so far are sufficient for the representation of elementary functions of public transit systems, but they neglect all higher-level management activities that contribute to the functioning and resilience of transit systems. To allow for management on a higher level than individual trips, the timetable must be supplemented by a rotation schedule, which combines trips into groups (so called rotations) [18] that can be executed by individual vehicles within an operating day. These and other management activities are encapsulated in three management modules: the fleet manager, the line manager, and the dispatcher. Thus, changes to the modeling of individual administrative activities do not affect the modeling of other areas of the simulation model. Work in progress on these modules has been reported in [19].

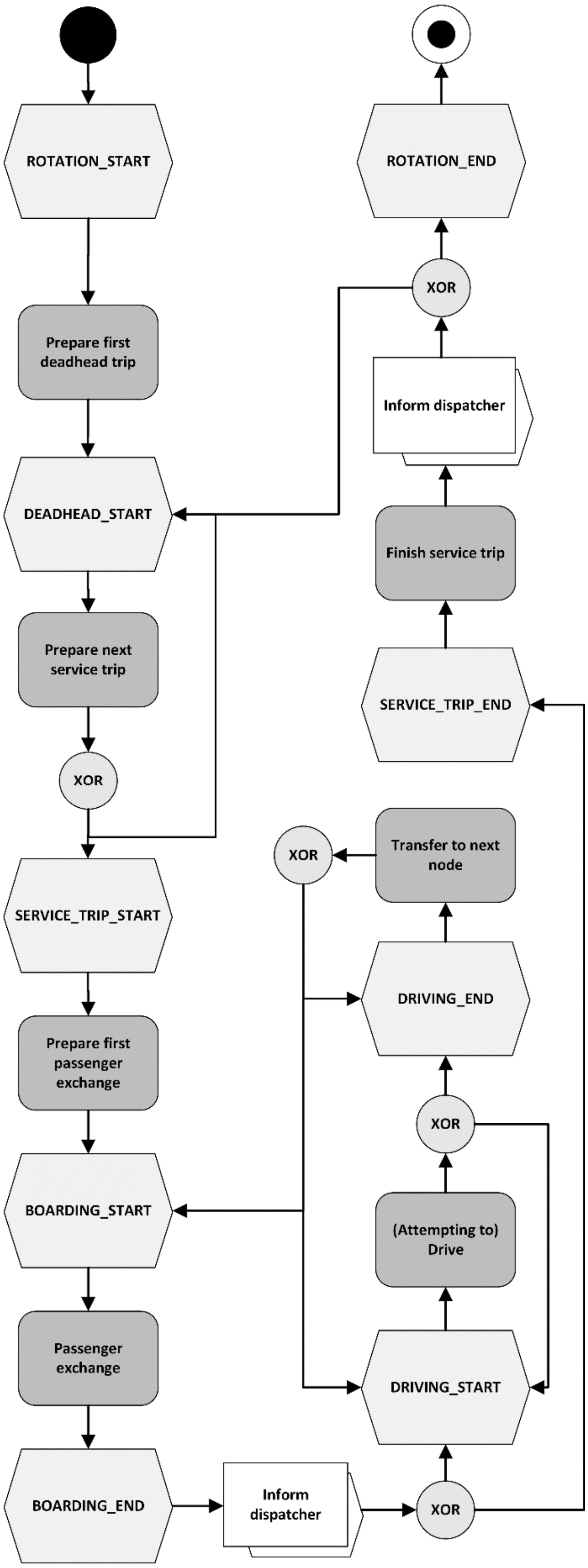

Figure 1: Relationships between the individual event and activity types for light-rail vehicles (based on the associated event process chains).

Simulation event type
ROTATION_START
ROTATION_END
DEADHEAD_TRIP_START
SERVICE_TRIP_START
TRIP_END
BOARDING_START
BOARDING_END
DRIVING_START
DRIVING_END

Table 1: Simulation event types for the light rail and bus vehicles.

Fleet manager

The fleet manager administrates the vehicle fleet and allows other components of the simulation model to access the vehicle fleet via defined interfaces, manages which vehicles are currently in use, and is responsible for generating and managing the initial rotation schedule.

If no rotation schedule is specified by the user, the fleet manager uses a rotation schedule generator to create an (artificial) rotation schedule. Initially, the simulation model is accompanied by a very simple generator, which is able to create schedules with the following properties:

i. Vehicles only make trips that are allowed for their type.
ii. Successive trips must belong to lines with identical line numbers.
iii. End and start stops of successive trips must be part of the same station
iv. Trips assigned to vehicles follow a permissible time order, i.e., the last trip assigned must end before the next trip starts; and
v. A user-definable minimum turn time is observed between successive trips assigned to a vehicle.

As Algorithm 1 shows in pseudocode, the procedure traverses the given set of (service) trips (line 03) for this purpose and, for each trip, searches the set of vehicles with permissible vehicle type (line 05) either for a vehicle whose last assigned trip (line 06) satisfies conditions ii. to v. (lines 09 to 11) or, if no such vehicle exists, for a vehicle that has not yet been assigned a trip (lines 07 and 08). If more than one vehicle is suitable to complete a trip, the vehicle whose most recently assigned trip has the earliest scheduled completion time is selected (lines 13 and 14).

If a given vehicle fleet is not suitable to cover all trips under these conditions, or if no information is given about the vehicle fleet, the procedure generates (artificial) vehicle entities and adds them to the vehicle fleet (lines 18 to 22).

```
INPUT: Set of service trips B, vehicle fleet F, minimum
turnaround time tw
OUTPUT: Rotation schedule
01 procedure createRotationSchedule(B, F)
02  begin
03   for all b ϵ B do
04    Vehicle fopt := 0
05    for all f ϵ F with type(f) ∩ type(b) ≠ Ø do
06     ServiceTrip bf := last(trips(f))
07     if bf = 0 then
08      fopt := f
09     else if lineNumber(bf) = lineNumber(b)
10      and stop(end(bf)) = stop(start(b))
11      and startTime(bf)+travelTime(bf)+tw ≤ startTime(b) then
12       ServiceTrip bopt := last(tryps(fopt))
13       if startTime(bf) + travelTime(bf) < startTime(bopt) +
         travelTime(bopt) then
14        fopt := f
15       fi
16     fi
17    od
18    if fopt = 0 then
19     Vehicle fnew := createVehicle(type)b))
20     F := F ∪ fnew
21     fopt := fnew
22    fi
23    trips(fopt) := trips(fopt) ∪ b
24   od
25 end
```

Algorithm 1: Rotation schedule generator.

Line manager

The line manager administrates the lines served as part of a timetable and associates their outward and return directions with each other. In addition, it provides uniform interfaces for accessing individual lines as well as sets of lines. This allows, for example, access to all lines serving a specific stop or a specific connection.

Dispatcher

The dispatcher is the most important and comprehensive management module and can be understood as a model of the operator's decision processes. It holds all the data required for the operational process, such as the current timetables and rotation schedules at a specific point in time. In addition to managing regular operations, the dispatcher also includes the simulation logic required for traffic management. Four different event types address the module's behavior during regular operation (see Table 2).

Simulation event type
OPERATIONAL_DAY_START
OPERATIONAL_DAY_END
BOARDING_END
SERVICE_TRIP_END

Table 2: Simulation event types concerning the dispatcher module.

The event type OPERATIONAL_DAY_START models the start of the operating day. As a result of this event type, the dispatcher assigns to the vehicle entities of the vehicle fleet, based on the rotation schedule, the first trip to be served by them. A subsequent event of the type ROTATION_START is sent to each assigned vehicle entity.

The end of the operating day is modeled by the event type OPERATIONAL_DAY_END. It signals that all service trips have been performed and all vehicle entities have returned to the depot.

Vehicle entities send events of the type BOARDING_END, which signal the end of the pure boarding process, to the dispatcher during the simulated operating day. The dispatcher then makes further decisions on traffic management measures based on this information. For this purpose, the dispatcher can resort to different strategies (see [14] and [19] for detailed descriptions of different statregies).

For determining the departure time of a vehicle during regular operations, a location-based departure strategy is employed. Under this strategy, selected stops are defined as control stops at which vehicles always have to wait until their planned departure time, as defined by the timetable, has been reached (e.g., to allow transfers between bus and light rail systems). At all other stops of the network vehicles always depart as soon as the boarding process has been completed, regardless of whether the planned departure time has already been reached or not. If the dispatcher receives an event of the type BOARDING_END, it checks whether traffic management measures are to be applied or not. Depending on the result of this check, the waiting time to be added to the entry/exit time is determined.

This waiting time is communicated to the affected vehicle entity by sending it a subsequent event corresponding to the end of the waiting time. This subsequent event can be either of the type SERVICE_TRIP_END or DRIVING_START. The former is the case when the vehicle entity is at the last stop of its current trip. The latter is sent to tell the entity to move to the next node specified in the line route. In addition to traffic management used under 'normal operating conditions', the dispatcher also contains an arsenal of 'emergency traffic management strategies' (as described in [19]), e.g., dynamic rerouting of vehicles in case of blocked segments, shortturning of trips in case of high delay, or temporary splitting of routes.

The last event type relevant for regular operation is SERVICE_TRIP_END. It represents the end of a service trip and the subsequent signaling of the control center. As a result of this event, the dispatcher assigns the next service trip to the vehicle entity according to the current schedule and initiates the previously required deadhead. It is ensured that the minimum turnaround time specified by the user is observed between the end of one service trip and the start of the next one. If the finished service trip was the last planned trip of the vehicle entity for the simulated operational day, the dispatcher instructs it to end its rotation. If all trips to be performed on this operating day are completed at the end of the service trip, the dispatcher ends the operating day by scheduling an event of type OPERATIONAL_DAY_END.

2.5 Randomization

Two randomized elements are part of the proposed model: the vehicle's traversal time for connections, and the passenger exchange times at stops. Both are directly adapted from [14] with only slight adaptation.

A lognormal distribution is assumed for the **traversal times** for a connection c [20]. Lacking detailed data, the parameters of this distribution, i.e., expectancy value and standard deviation, have to be approximated from the planned traversal times $t_p(c)$. These traversal times usually comprise the planned driving time $t_d(c)$ and the planned passenger exchange time $t_b(c)$, which in turn are comprised of average observed driving/passenger exchange times, standard deviations, and unknown terms (see Equation 1).

$$\begin{aligned} \mathrm{t_p}(c) &= t_d(c) + t_b(c) \\ &= (\mu_c^d + \sigma_c^d + \epsilon_c^d) \\ &+ (\mu_c^b + \sigma_c^b + \epsilon_c^b) \end{aligned} \quad (1)$$

It can be assumed that the planned traversal time $t_p(c)$ is greater than the average observed traversal time μ_c^d to avoid systematic delays. The average traversal time can then be roughly approximated as follows:

$$\hat{\mu}_c^d = t_p(c) * \gamma, \forall c \in C, 0 < \gamma < 1 \quad (2)$$

The ratio γ has to be determined by the user. The standard deviation σ_c^d can be approximated in the same way. It can be assumed that the standard deviation is only a small fraction of the planned traversal time. This yields Equation 3.

$$\hat{\sigma}_c^d = t_p(c) * \eta, \forall c \in C, 0 < \eta < 1, \eta \ll \gamma \quad (3)$$

The **passenger exchange times** for busses and light-rail vehicles can be modeled following the method first proposed in [21]. This method is suitable for high frequency transit systems like urban light-rail and bus transit, where it can be assumed that passengers arrive randomly during the inter-arrival time of two successive vehicles, instead of arriving in bulk shortly before the planned departure time. Furthermore, the method facilitates the modeling of vehicle bunching, i.e. the effect that two vehicles form an undesired platoon because the vehicle in front takes on more passengers than planned and subsequently suffers longer passenger exchange times, while the rear vehicle takes on fewer passengers as planned and thus catches up to the vehicle in front.

If the number $N_{b,s}$ of passengers entering a vehicle b at a stop s, and the average time I_b a passenger takes to enter vehicle b are known, the passenger exchange time $T_{b,s}$ can be determined as follows:

$$T_{b,s} = T_b^{min} + I_b + N_{b,s} \quad (4)$$

Here T_b^{min} describes a vehicle specific minimum time, e.g., for opening and closing the vehicle's doors. If the passenger arrival rate a_s at stop s is known, $N_{b,s}$ can be modeled dependent on the basic interval $T_{L(b)}$ of line $L(b)$ currently served by vehicle b. With $N_{b,s} = T_{L(b)} * a_s$ the passenger exchange time can then be approximated as shown in Equation 5.

$$T_{b,s} = T_b^{min} + I_b + T_{L(b)} * a_s \quad (5)$$

If instead of the basic interval between vehicles of the same line, simulated headways between successive vehicles servicing the same stop are used, the model becomes dynamic and thus suitable for a simulation model.

If $t_{dep}(b-1,s)$ describes the time a vehicle b's predecessor has serviced the stop, the passenger exchange time $T_{b,s}(t_{sim})$ can be determined by Equation 6.

$$T_{b,s}(t_{sim}) = \begin{cases} T_b^{min}, & b \textit{ is first vehicle at } s \\ T_b^{min} + \left(t_{sim} - t_{dep}(b-1,s)\right) * a_s * I_b, & else \end{cases} \quad (6)$$

3 Results and Discussion

For the model to be practically applicable in the use case described above, transit operators have to be able to execute it quickly: When disaster hits, it is not feasible to wait many hours for first simulated data on the impact of mitigation strategies to to come in.

To estimate its execution time, the model was implemented as a Java application [12], and a number of experiments were conducted with representations of the transit network of the authors' hometown of Cologne, Germany. Table 3 depicts the number of simulation events executed per simulation run, while Table 4 depicts the runtime in seconds of the various phases of the simulation run, each for a model of the urban light-rail and the bus network. The preprocessing stage is mainly concerned with reading the model from a database and building up data structures, it has to be run only once independent of the number of simulations runs to be executed. Following each model execution itself a short postprocessing phase serializes the resulting data and releases memory. After all simulation runs have been successfully completed, the application engages in a final and more comprehensive postprocessing phase, reading the results of the individual runs, calculates statistics, assigns them to the relevant entities, and constructs Excel sheets with graphical overviews and detailed reports.

Simulation event type	Light rail	Bus network
DEADHEAD_TRIP_START	1,827	4,080
SERVICE_TRIP_START	1,800	4,080
TRIP_END	3,585	8,150
BOARDING_START	50,470	89,015
BOARDING_END	50,470	89,015
DRIVING_START	60,746	84,951
DRIVING_END	62,913	84,961

Table 3: Average number of main simulation events per simulation run for the light rail and bus networks.

Execution stage	Light rail	Bus network
Preprocessing	6.19	9.31
Model execution	6.77	12.88
Postprocessing per run	0.15	0.30
Postprocessing final	1,270.72	2,229.28
Total runtime (1 run)	1,283.83	2,251.77
Total runtime (100 runs)	1,968.91	3,556.59
Runtime per run (100 runs)	19.69	35.57

Table 4: The simulation application's run time in seconds for the bus and light-rail models, broken down to the common preprocessing step executed only once, the duration of an actual simulation run itself, the postprocessing phase for each run, and the final postprocessing phase only executed once. Assuming the application is used to execute 100 runs, the run time per run amounts to 19.69 seconds for the light-rail model and 35.57 seconds for the larger bus network model.

As seen, the timing requirement is only partially met: While the core model execution itself is concluded in under 7 seconds for the light rail network and under 13 seconds for the much larger bus network, the postprocessing phase with its 1,270 and 2,229 seconds is much too long for the envisioned use case.

Here, instead of compiling comprehensive statistics on all involved model entities, the application has to be reworked to only collect and depict only the statistics most relevant to the operators.

4 Conclusions

This paper presented a bi-modal transit simulation model aimed at supporting public transit operators to increase the resilience of public transit infrastructure. Central aims include to enable operators to evaluate measures against small disturbances and larger outages as they happen, as well as to evaluate what combination of disaster risk management and resilience-building strategies shows most potential to help increasing the resilience of urban transit systems against extreme weather events resulting from climate change as well as other disasters.

The paper discussed the components of public transit infrastructure systems and then went on to present the developed simulation model, focusing on modelling physical components, the logical network, vehicle behaviour, operational decisions, and the necessary randomization.

The discussion of first results showed that the model – implemented as a Java application – works well and executes the core model quickly enough for the stated use cases. However, the currently employed analytical post-processing engine collects a much too comprehensive set of statistics and has to be adapted to priorize information that is most relevant to transit operators in case of emergencies.

In further research and development steps, after overhauling the statistics engine, the model will be applied to evaluate what combination of disaster risk management and resilience-building strategies shows most potential to help increasing the resilience of urban transit systems against extreme weather events resulting from climate change as well as other disasters.

Acknowledgements

This paper is an updated and extended version of the conference paper "Towards Increasing Resilience of Public Transit Infrastructure – A Bi-Modal Simulation Model" by Daniel Lückerath and Oliver Ullrich that was published as part of the proceedings of ASIM 2022, 26th Symposion on Simulation Techniques. It has been partially supported by the framework of the European project ARCH – Advancing Resilience of historic areas against Climate-related and other Hazards. This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement no. 820,999. The sole responsibility for the content of this publication lies with the authors. It does not necessarily represent the opinion of the European Union. Neither the REA nor the European Commission are responsible for any use that may be made of the information contained therein.

References

[1] Olfert A, et al. Sustainability and resilience - a practical approach to assessing sustainability of infrastructure in the context of climate change. Hutter G, Neubert M, Ortlepp R. Building resilience to natural hazards in the context of climate change - Knowledge integration, implementation, and learning. Springer, 2021.

[2] Vuchic VR: *Urban Transit - Systems and Technology*. Verlag John Wiley & Sons, Hoboken, New Jersey 2007.

[3] Behrisch M, Erdmann J, Krajzewicz D. Adding intermodality to the microscopic simulation package SUMO. In: Al-Akaidi, M., editor. MESM' 2010 – GAMEON-ARABIA'2010. *11th Middle Eastern Simulation Multiconference*; 2010 Dec; Alexandira. Alexandira: eurosis. 59-66.

[4] Kendziorra A, Weber M. Extensions for logistics and public transport in SUMO. In: Behrisch M, Weber M, editors. SUMO 2015 – Intermodal Simulation for Intermodal Transport. *3rd SUMO User Conference*; 2015 May. Berlin: Deutsches zentrum für Luft- und Raumfahrt, Institut für Verkehrssystemtechnik. 83-90.

[5] Toledo T, Cats O, Burghout W, Koutsopoulos H. Mesocopic simulation for transit operations. *Transport. Res. C-Emer*. 2010; 18(6): 896-908.

[6] Suzumura T, Kanezashi H. Multi-modal traffic simulation platform on parallel and distributed systems. In: Tolk A, Diallo S, Ryzhow I, Yilmaz L, Buckley S, Miller J. Proceedings of the 2014 Winter Simulation Conference. *Winter Simulation Conference*; 2014 Dec; Savannah. Piscataway, NJ, USA: IEEE Press. 769-780.

[7] Suzumura T, McArdle G, Kanezashi, H. A high-performance multi-modal traffic simulation platform and its case study with the Dublin city. In: Yilmaz L, Chan W, Moon I, Roeder T, Macal C, Rossetti M, editors. Proceedings of the 2015 Winter Simulation Conference. *Winter Simulation Conference*; 2015 Dec; Huntington Beach. Piscataway, NJ, USA: IEEE Press. 767-778.

[8] Burghout W. *Hybrid microscopic-mesoscopic traffic simulation* [dissertation]. Department of Infrastructure, Royal Institute of Technology, Sweden. University of Stockholm, 2004.

[9] Ullrich O, Proff I, Lückerath D, Kuckertz P, Speckenmeyer E. Agent-based modeling and simulation of individual traffic as an environment for bus schedule simulation. In: Busch F, Spangler M, editors. ITS for Connected Mobility. *mobil.TUM*; 2014; Munich. Munich: Schriftenreihe des Lehrstuhls für Verkehrstechnik der Technischen Universität München. 89-98.

[10] Kastner K, Keber R, Pau P, Samal M. Real-Time Traffic Conditions with SUMO for ITS Austria. In: Behrisch M, Knocke M, editors. 1st SUMO User Conference 2013. *1st SUMO User Conference*; 2013 May, Berlin. Berlin: Deutsches Zentrum für Luft- und Raumfahrt, Institut für Verkehrssystemtechnik. 43-53.

[11] Kastner K, Pau P. Experiences with SUMO in a Real-Life Trafic Monitoring System. In: Behrisch M, Weber M, editors. SUMO 2015 – Intermodal Simulation for Intermodal Transport. *3rd SUMO User Conference*; 2015 May. Berlin: Deutsches zentrum für Luft- und Raumfahrt, Institut für Verkehrssystemtechnik. 1-10.

[12] Lückerath D: *Ein Simulationsmodell für Öffentlichen Personennahverkehr mit regelbasiertem Verkehrsmanagent* (Dissertation). Institut für Informatik, Universität zu Köln; 2017, 182 pg.

[13] Ullrich O, Lückerath D. An Introduction to Discrete-Event Modeling and Simulation. *Simulation Notes Europe (SNE)* 2017; 27(1): 9-16.

[14] Lückerath D, Rishe N, Speckenmeyer E, Ullrich O. A Mesoscopic Bus Transit Simulation Model Based on Scarce Data. *Simulation Notes Europe (SNE)* 2018; 28(1): 1-10.

[15] Vossloh Kiepe GmbH. *Elektrische Ausrüstung der Niederflur-Stadtbahnwagen K4000 der Kölner Verkehrs-Betriebe AG 2003.* Druckschrift Nr. 00KV7DE.

[16] Vossloh Kiepe GmbH. *Elektrische Ausrüstung des Niederflur-Stadtbahnwagens K4500 für die Kölner Verkehrs-Betriebe AG 2003.* Druckschrift Nr. 00KN2DE.

[17] Vossloh Kiepe GmbH. *Elektrische Ausrüstung der Hochflur-Stadtbahnwagen K5000 der Kölner Verkehrs-Betriebe AG 2003*, Druckschrift Nr. 00KB5DE.

[18] Lückerath D, Ullrich O, Kupicha A, Speckenmeyer E. Multi-depot multi-vehicle-type vehicle scheduling for Cologne's tram network. In: Proc. *ASIM-Workshop STS/GMMS 2014*, ARGESIM Report 42, ASIM-Mitteilung AM 149, ARGESIM/ASIM Pub., TU Vienna/Austria, 191-197.

[19] Lückerath D, Bogen M, Rome E, Sojeva B, Ullrich O, Worst R, Xie J. Strategies to Mitigate the Impacts of Climate Change Related Events on Public Transit Networks. In: Proc. *24th Symp.Simulationstechnik*, Hamburg, Germany, October 03-04, 2018, 175-182.

[20] Andersson P, Hermansson A, Tengvald E, Scalia-Tomba G. Analysis and simulation of an urban bus route. *Transport. Res. A-Pol.*, 1979, 13(6): 439-466.

[21] Chapman R, Michel J. Modelling the Tendency of Buses to Form Pairs. *Transport. Sci.*, 1978, 12(2): 165-175.