A Mesoscopic Bus Transit Simulation Model based on Scarce Data

Daniel Lückerath[1*], Ewald Speckenmeyer[2], Oliver Ullrich[3], Naphtali Rishe[3]

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

[2]Department of Computer Science, University of Cologne, Weyertal 121, 50931 Cologne, Germany

[3]School of Computing and Information Sciences, Florida International University, ECS 243C, 11200 SW 8th St, Miami FL-33199, USA

SNE 28(1), 2018, 1 - 10, DOI: 10.11128/sne.28.tn.10401
Received: July 7, 2017; Revised December 10, 2017;
Accepted December 15, 2017
SNE - Simulation Notes Europe, ARGESIM Publisher Vienna,
ISSN Print 2305-9974, Online 2306-0271, www.sne-journal.org

Abstract. Typically, public transit modeling requires the availability of an extensive data basis to enable detailed modeling, calibration, and validation. Sometimes such data is not available, even though there is demand for a simulation model to examine the impacts of planning decisions and strategies. However, available data may support a simulation model which, while not perfect, at least yields plausible results allowing the examination of broader impacts of planning decisions and strategies.

In this paper a bus transit simulation model custom-tailored to manage on a scarce data basis is described. After an introduction to aims and scope, some background on bus transit systems, data availability, and related research is shared. Then a simulation model utilizing the available scarce data is proposed and the representation of the physical transit network, logical components, vehicle behavior, and transit provider's everyday operational management decisions are described. Finally, the outcomes of initial experiments on a small artificial transit network model are discussed, demonstrating the model's ability to yield plausible results.

Introduction

Typically, public transit modeling requires the availability of a broad data basis, to enable detailed modeling, calibration, and validation.

In some cases, such data is not available, even though there is demand for a simulation model to examine the impacts of planning decisions and strategies. However, available data may support a simulation model which, while not perfect, at least yields plausible results allowing for the examination of broader impacts of planning decisions and strategies.

In this paper an event-based simulation model of bus transit based on scarce data is presented. The model is intended to be used as a tool to evaluate planning decisions and operational management rules to mitigate disturbances in transit networks, e.g. faulty vehicles, blocked stops, or unusually high traffic on street segments. For this intended usage, it is not necessary to include every single behavioral decision in a model, but rather design it in such a way that specific key indicators, e.g. delays, kept connections, or service regularity can be measured. Therefore, the proposed model utilizes a more abstract mesoscopic approach to model part of the vehicle behavior.

Many transit models are developed as an extension of already established models of individual traffic (see e.g. [3], [8], [17]). If these models utilize a fine grained modeling approach, they generally necessitate the availability of an extensive data basis, including detailed information on origin-destination matrices, vehicular dynamics, signaling strategies, and lane changing rules (e.g. see [18]), and include many components which are not immediately interesting for public transit systems. Moreover, employing microscopic models of individual traffic for the examination of large areas (e.g. public transit networks of whole cities) often results in prolonged run times (see e.g. [6], [7]) and the parametrization of individual traffic (see e.g. [17]), reversing the initial modeling decision.

The model proposed in this paper is not based on any pre-existing model of individual traffic, but is newly developed based on the specifics of bus transit, and specifically prepared with the intended use in mind. That, in unison with using a mesoscopic modeling approach, allows for the construction of a simulation model which manages on a very scarce data basis, most of which is publicly available for many transit systems.

The paper continues with some background on bus transit systems, available data, and related research (see section 1). Afterwards a mesoscopic, event-based simulation model of bus transit utilizing scarce data is proposed, describing in detail the representation of the physical transit network, logical components, vehicle behavior, and the transit provider's operational management strategies (see section 2). To evaluate the simulation model's behavior some experiments are conducted based on a model of a small artificial transit network (see section 3). The paper concludes with a summary and an outlook of further research steps (see section 4).

1 Background

1.1 Bus transit networks and available data

A bus transit system consists of a street network and a set of bus stops where passenger exchanges take place. These bus stops are served by a set of transit vehicles executing service trips, i.e. pairings of starting times and sequences of bus 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.

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.

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

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

Public transit vehicles generally follow pre-defined line routes, consisting of sequences of stops to be serviced.

In most public transit systems, daily operations are managed by an operations center, with dispatcher personnel managing procedures for the mitigation of disturbances originating e.g. from street segments blocked by accidents, or failing transit vehicle doors. These operators have a number of remedies at their disposal, including the authority to short-turn or cancel trips, and to deploy extra vehicles.

The proposed model manages on a minimum of data which is publicly available for many transit systems: a list of stops and their connections, including planned traversal times; timetables for each of these stops; descriptions of the lines and their variants, including the order of stops to be serviced; as well as type and attributes of the vehicles used.

Additional data improves the simulation's accuracy: the distribution of traversal times for each connection and the passenger arrival rate for each stop, both depending on the time of day. In addition, empirical data on individual departure times help with calibration and validation of the model.

1.2 Related research

A number of simulation models covering bus transit can be found in the literature (see e.g. [1], [3], [8], and [15] - [17]).

One of the first models was proposed in 1979 by Andersson et al. in [1]. The authors develop a mesoscopic event-based interactive simulation model for bus transit systems allowing users online testing of operational strategies, like short-turning trips or deadheading vehicles. Andersson et al. model the bus system as a set of lines, i.e. collections of linked stops, where each stop posseses a separate holding bay for every line serving it. As a result, the model does not represent direct vehicle interactions. Instead, interactions between vehicles are modeled indirecly via the passenger exchange process. Because passengers can be served by multiple lines, delays or earliness of a vehicle of one line may affect the passenger exchange processes of vehicles of other lines, resulting in vehicle bunching effects.

The traversal process of vehicles between successive stops is modeled mesoscopically using lognormal distributed random values dependent on the time of day.

A newer mesoscopic approach to bus transit simulation is proposed by Toledo et al. in [17]. The authors extend a mesoscopic simulation model for individual traffic based on queuing theory proposed by Burghout in [4], which represents the street network as a graph of interconnected queues and vehicles as individual entities traversing these queues based on speed/density functions. The nodes of the graph represent junctions, and are modeled as collections of servers, one for each turning movement and each with different processing times based on e.g. the green time ratio of the corresponding signal.

Other recent simulation models including bus transit use microscopic agent-based modeling approaches all of which are based on generic multi-modal transit models (see e.g. [3], [8], [15], and [16]). The open source simulation framework *SUMO* (Simulation of Urban Mobility, for an overview see [2]) started as an agent-based simulation model for individual traffic, representing individual drivers as agents with individual attributes and microscopic vehicle dynamics as well as lane changing behavior. These agents traverse a graph representing the street network, with street segments as edges and junctions as nodes. In [2] and [8] Behrisch et al. and Kendziorra and Weber extend SUMO with the capability to incorporate public transit and individual travellers, respectively. For this purpose, public transit vehicles are modeled as agents of individual traffic with fixed routes, which stop at predefined locations to let passengers board and alight.

A very comprehensive agent-based simulation model including bus transit is proposed by Suzumura et al. in [15] and [16]. They employ the *IBM Mega Traffic Simulator* (see [14]) to develop a parallel, agent-based model for microscopic transit simulation called *M3*. Every single participant and every potential transit mode (e.g. bus, light rail, car, bicycle) is represented by agents, which move through multiple interconnected graphs representing the different transit networks. As in *SUMO*, public transit agents are modeled similar to agents of individual traffic, but with fixed routes on which they have to serve predefined stops.

Apart from the model proposed by Andersson et al., all discussed modeling approaches require large amounts of data about individual traffic and the underlying street network (e.g. origin-destination matrices, signaling strategies). However, when employed to real-world test cases most users of these models choose to parameterize some or all of these model aspects, due to run-time or other concerns (see. e.g. [6], [7], and [17]). Therefore, this paper continues with a new modeling approach taking these considerations into account by basing all modeling decisions on the specifics of public transit systems and simplifying some transit modeling aspects using a mesoscopic approach.

2 Modeling Bus Transit

A public transit system can be decomposed into a number of sub-systems: the physical network consisting of stops, connections and signals, the logical network consisting of lines, rotations, and planned trips, the vehicle sub-system consisting of the buses and their behavior, and the transit provider's operational management decisions. Some of these components show a stochastic behavior and are therefore subject to randomization.

In the following, these system components and their behavior are represented by simulation entities, events and activities, and thus translated to an object-oriented, event-based simulation model (see [19]).

Given the scarcity of the available data, a main goal of the modeling process is to avoid unjustified complexity. While individual vehicles are considered explicitly, their driving dynamics are not modeled in detail, but are subsumed with a certain abstraction.

2.1 Physical network

The physical network is represented by a directed graph $G = (V, E)$, where stops and connections are modeled as nodes $v \in V$, and their neighborhood relations are represented by edges $e \in E$.

A stop $s \in S$ is attributed with an identifier, time of day specific passenger arrival rates and a maximum capacity for concurrently stopping vehicles. In addition, some stops are marked as control points. As stops are assigned to exactly one station, each entity contains a reference to its station object.

Connections $c = (s_i, s_j) \in C \subseteq S \times S$ are directed paths through the transit network between two successive stops s_i and s_j. Connections are attributed with a length, a planned traversal time $t_p(c)$ according to the timetable, and might also contain a list of atomic street segments and a set of signals.

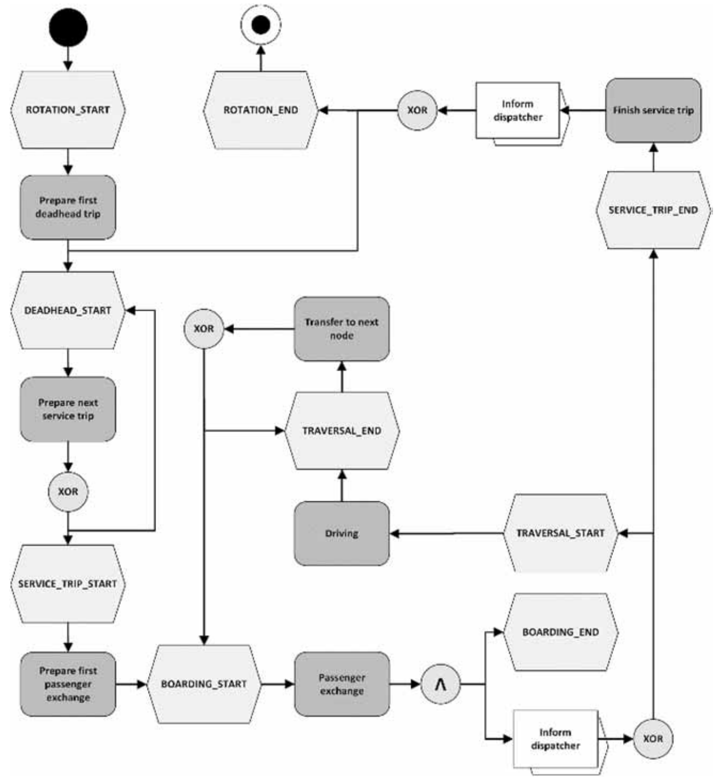

Figure 1: Event-activity-chain for the vehicle sub-model. Hexagons represent events, squares with rounded edges represent activities.

2.2 Logical network

A line $l \in L$ is modeled as an ordered list $l_i = (s_{i_1}, c_{i_1}, s_{i_2}, c_{i_2}, \dots, c_{i_{n-1}}, s_{i_n})$ of stops $s_{i_j} \in S$, which are to be serviced successively by a vehicle of a specific type, interspersed with the relevant connections $c_{i_j} \in C$ between each two successive stops. This avoids elaborate path finding over the course of a simulation run.

Planned service trips are tuples of a line to be served and a planned departure time at the first stop of that line. The set of all service trips defines the services available to prospective passengers during an operational day, i.e. the timetable.

Rotations are ordered sequences of all the trips to be executed by one – not yet specified – vehicle during a single operational day. A vehicle schedule assigns one specific vehicle to each rotation.

2.3 Vehicles

Vehicles are represented by entities traversing the model graph over the course of a simulation run according to timetable, vehicle schedule, and operational strategies, encapsulating a significant part of the simulation logic.

Vehicles are classified according to their type and attributes: the vehicle type defines vehicle length, capacity, maximum velocity, minimum passenger exchange time and exchange rate. These attributes can be overridden by setting values for individual vehicles, enabling both the representation of vehicles of the same type but with different equipment and individual vehicle defects (e.g. lower passenger exchange rates caused by a faulty door).

Each vehicle entity keeps a reference to the node it currently inhabits, i.e. its current position, as well as a reference to the trip it is currently executing. The entity therefore only has knowledge about the environment immediately important to it, all other information, e.g. regarding rotations, timetable or vehicle schedule, is managed by the dispatcher module (see section 2.4).

The vehicle sub-models execute nine types of simulation events (see figure 1): The event types ROTATION_START and ROTATION_END concern the start and end of a vehicle's assigned rotation and associated activities, e.g. notifying the relevant operational management modules. The event type DEADHEAD_START represents the beginning of a deadhead trip.

As the scarce data basis does not include information on actual planned deadhead trips, the model does not execute these trips, but approximates them by following the strategy described in [10] and [12]: after the conclusion of a service trip the vehicle entity is taken out of the model, and right before the planned start of the next planned service trip the entity tries to access the initial stop of the scheduled trip's route. If this node is already filled up to capacity, the respective event is rescheduled until the next vehicle leaves the stop. While this strategy circumvents the missing information, it omits the representation of traffic load generated by deadhead trips. As real-world vehicle schedules are aimed at minimizing costly deadhead trips, the loss of accuracy resulting from this approach is justifiable. The event types SERVICE_TRIP_START and SERVICE_TRIP_END frame the execution of service trips. While the activities triggered by SERVICE_TRIP_START mainly concern preparations for the first passenger exchange, SERVICE_TRIP_END events notify the operational management modules that the vehicle is available to execute further trips. The event types BOARDING_START and BOARDING_END concern the passenger exchange. The passenger exchange time depends on the stop, the vehicle, as well as the time of day and takes the inter-arrival time of successive vehicles into account to model bus bunching (see section 2.5). In addition to being executed as part of the vehicle sub-model, a BOARDING_END event is also sent to relevant operational management modules to allow for the execution of operational strategies. Once the passenger exchange is completed and the operational management module scheduled a departure time, a TRAVERSAL_START event is triggered. The subsequent driving activity is modeled mesoscopically by drawing the necessary traversal time from a random distribution (see section 2.5). An event of type TRAVERSAL_END completes the traversal of the connection. In case the next stop is filled up to capacity, the TRAVERSAL_END event is rescheduled for the predicted time of the blocking vehicle's departure, so the current vehicle can approach the stop.

In case more detailed data is available, the traversal behavior can be represented microscopically, e.g. by dividing the traversal activity into smaller, interconnected activities. An example of this strategy was presented by the authors in [12].

2.4 Operational management

The introduction of operational management modules allows to separate the simulation logic for the execution of a single trip from the simulation logic for the overall organization of an operational day. Furthermore, it allows for easy incorporation of operational strategies used by transportation providers (see e.g. [13]). The simulation model includes three operational management modules: one for vehicle scheduling and fleet management, one for line management, and one for operational decision making and disturbance mitigation. The latter – the dispatcher – constitutes the most important module, encapsulating the simulation logic for the overall organization of the operational day as well as the simulation logic for decisions regarding the operational behavior of vehicles (see figure 2). For this purpose, the module holds data on nearly all model components, including planned and actual timetable as well as vehicle schedule.

The complete operational day is framed by events of the types OPERATIONAL_DAY_START and OPERATIONAL_DAY_END. While executing OPERATIONAL_DAY_START the dispatcher assigns a first trip to execute to each vehicle entity in the vehicle pool, based on the vehicle schedule. Each vehicle's first trip is started by enqueuing an event of the type ROTATION_START. An event OPERATIONAL_DAY_END signals the completion of all service trips, and the return of all vehicles to their respective depots. At each occurrence of an event of type BOARDING_END the module determines statistical indicators for the corresponding vehicle entity, decides on potential operational strategies to carry out, and schedules a TRAVERSAL_START event accordingly. The simplest form of operational strategy only considers the current simulation time t_{sim} and the planned time of departure $t_{dep}(b,s)$ of vehicle b at stop s, and only in case the stop is a control point, thus scheduling the departure at $t = \max\big(t_{dep}(b,s), t_{sim}\big)$. If s is not a control point, the event is scheduled for the current simulation time t_{sim}. The end of a service trip and its signalization to operational management is represented by an event of type SERVICE_TRIP_END. The dispatcher assigns the next trip to be executed to the vehicle entity, and, if necessary, prompts a deadhead trip. If the completed trip is the last trip of the vehicle's rotation, the dispatcher orders it to move to its depot and complete its rotation.

If all trips of all rotations are completed, the dispatcher schedules an event of type OPERATIONAL_DAY_END to effectively end the simulation run.

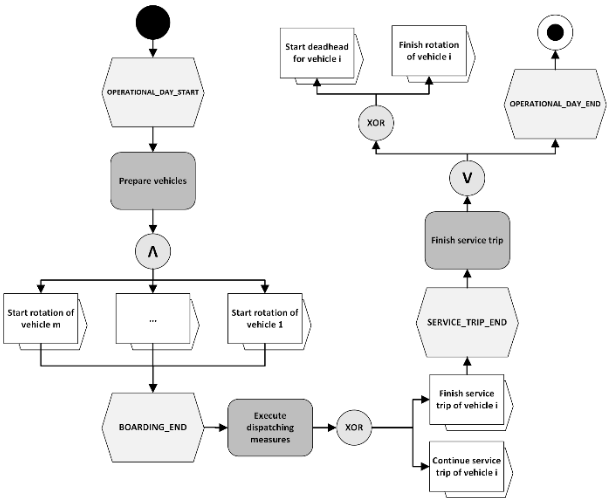
OPERATIONAL_DAY_START
Prepare vehicles
∧
Start rotation of vehicle m
...
Start rotation of vehicle 1
BOARDING_END
Execute dispatching measures
XOR
Finish service trip of vehicle i
Continue service trip of vehicle i
SERVICE_TRIP_END
Finish service trip
V
XOR
Start deadhead for vehicle i
Finish rotation of vehicle i
OPERATIONAL_DAY_END

Figure 2: Event-activity-chain for operational management logic.

2.5 Randomization

The described model includes two elements which are influenced by randomization: the vehicle's traversal time for connections, and the passenger exchange times at stops.

Ideally these values are randomized based on distributions whose types and parameters are determined by evaluating empirical data, dependent on location and time of day. As data is scarce in the described scenario though, other methods have to be applied to determine distribution type and parameters.

Traversal times. A lognormal distribution is assumed for the traversal times of a connection $c = (s_i, s_j)$ (see [1]). 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).

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

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.

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

Passenger exchange times. The passenger exchange times are modeled following the method described in [5]. This method is suitable for high frequency transit systems, 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 bus 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 as in shown in 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 Experiments

Given the scarcity of the available data, an in-depth comparison of the simulation output to real-world data cannot be conducted, and has to be replaced by a theory-driven evaluation (see e.g. [9], p. 206 ff.). Therefore, to evaluate whether the described simulation model yields plausible results, experiments are conducted based on a small artificial bus network. This network, named Universal City Link (UCL, see figure 3), is simple enough for simulation results to be counter-checked by hand, while also constructed to include many of the complications usually found in real-world bus systems (e.g. circular lines and differing basic intervals for different lines). It includes eight lines servicing 40 stops connected by 50 connections.

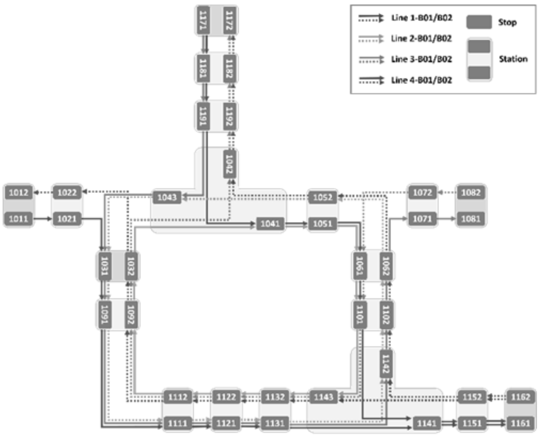

Figure 3: Artificial transit network UCL. Stations colored in darker gray represent starting stations of lines.

Two twelve-hour timetables from 7am to 7pm are compared. Both timetables employ basic intervals of ten minutes for lines 1-B01, 1-B02, 3-B01, 3-B02, 4-B01, and 4-B02 and twenty minute basic intervals for circular lines 2-B01 and 2-B02. For each timetable 100 simulation runs are executed under the same conditions: The first stop of every line is chosen as a control point where vehicles are not allowed to depart before their planned departure time, at all other stops vehicles can depart as soon as the passenger exchange is finished.

The average time a passenger needs to enter a vehicle I_b is set to three seconds for all vehicles, based on random samplings by the authors during their own commutes.

The minimum passenger exchange time T_b^{min} is set to twelve seconds, the time a public transit vehicle usually needs to open and close its doors (see [10]).

The passenger arrival rates a_s are assumed to be constant and chosen in such a way that the average passenger exchange time $T_{b,s}$ equals 20 seconds at every stop, i.e. $T_{b,s} = \bar{T} = 20$. To this end, the values for $\bar{T}$, T_b^{min}, I_b, and $T_{L(b)}$ are inserted into equation 5 and it is solved for a_s. Here, the values for the basic intervals $T_{L(b)}$ are replaced by the theoretically best achivable headways at the different stops of the network, i.e. the equidistanly divided common basic interval at every stop (see equation 7).

$$T_{L(b)} = \left\lfloor \frac{\min\limits_{l_i,l_j \in L(s), l_i \neq l_j} \gcd(T_{l_i}, T_{l_j})}{|L(s)|} \right\rfloor \tag{7}$$

$L(s)$ represents the set of all lines serving stop s and T_{l_i} the basic interval of line $l_i \in L$.

In order to determine values for the ratios γ and η of the traversal times (see equations 2 and 3), the average planned traversal time $\overline{t_p}$ and average passenger exchange time $\bar{T}$ are employed, resulting in $\gamma = 1 - \left(\frac{\bar{T}}{\overline{t_p}}\right) = 1 - \left(\frac{20}{120}\right) \approx 0.83$ and $\eta = 1 - \gamma = 0.17$. Together with the chosen arrival rates, these values should result in rather moderate departure time deviations under both timetables. However, while the first timetable, called UCL+, fits the simulated conditions, i.e. the passenger arrival rates, well, the second timetable, called UCL-, does not fit the simulated conditions. Accordingy, the simulation results should allow to identify a better performance under timetable UCL+ than under timetable UCL-, despite only moderate departure deviations.

The examination of the average departure deviation under both timetables confirms these assumptions: Under timetable UCL- delayed departures on average deviate 11.9s from their planned departure time, while they on average only deviate 8.5s under timetable UCL+, a reduction of 28.6%. Simultaneously, early departures under timetable UCL- on average deviate 3.9s from their planned departure times, while the deviation of early departures under timetable UCL+ is on average 4.7s, an increase of 20.5%. Accordingly, timetable UCL+ exhibits more early departures, while timetable UCL- shows a higher number of delayed departures (see figure 4): Under timetable UCL- 2,699 of 4,740 departures (56.9%) are late, of which 2,175 (45.9%) exhibit a delay of 30s or less. The number of early departures under timetable UCL- is 1,391 (29.4%), of which 1,258 (26.5%) do not exceed a deviation of 30s.

In contrast, under timetable UCL+ 2,361 of 4,740 departures (49.8%) are late, with 2,034 (42.9%) exhibiting a delay of 30s or less. 1,700 departures (35.9%) under timetable UCL+ are early, of which 1,575 (33.2%) are at most 30s early.

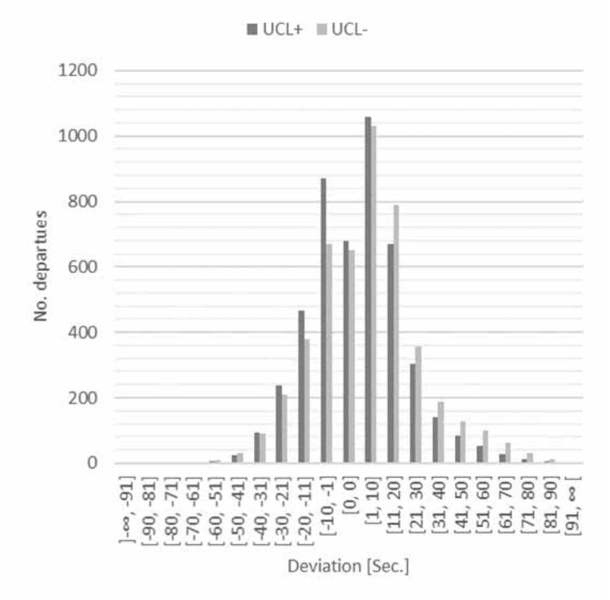

Figure 4: Departure deviation frequency distribution.

Analysing the observed vehicle bunching effects under both timetables, their difference in suitability becomes more distinct. Vehicle bunching effects are measured during simulation runs via the cumulative relative headway redaction ratio, which measures the cumulative percentage of observed headways exhibiting a certain amount of reduction relative to their scheduled value.

As can be seen in figure 5 both timetables exhibit roughly the same amount of overall headway reduction, with timetable UCL- having a 2.01% higher reduction than timetable UCL+. However, under timetable UCL+ the bulk of the reduced headways (16,5%) stay under ten percent, while under timetable UCL- the mayority of reduced headways exhibit a reduction of ten percent or more. And while there is virtually no headway reduction of 50% or more under timetable UCL+, 7.60% of reduced headways under timetable UCL- exhibit such a significant reduction. This suggests that timetable UCL- more heavily suffers reliability issues than timetable UCL+, verifying that the latter is better suited to the simulated conditions and that the simulation model can indeed be used to evaluate the suitability of a timetable for certain simulation conditions.

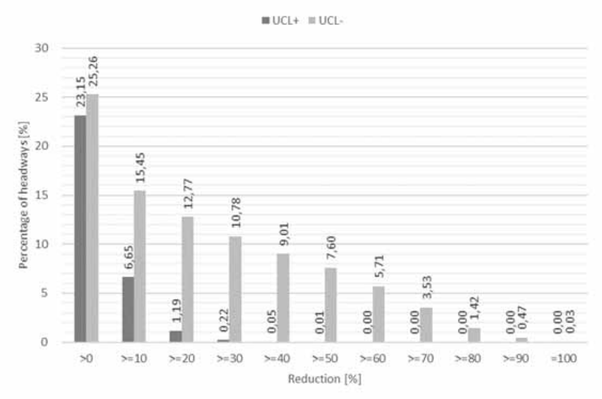

Figure 5: Cumulative relative headway reduction ratio.

To further ensure the plausibility of the simulation model, an examplary analysis of the departure deviation development of line 1-B01 under both timetables is conducted. To this end, the average, median, maximum, and minimum departure deviation as well as its 25%- and 75%-quantile is measured at every stop along the route of line 1-B01. As can be seen in figures 6 and 7 the departure deviation development of line 1-B01 shows roughly the same pattern under both timetables, reaching the highest delay at stop 1121, three stops after line 1-B01 joins lines 2-B02 and 3-B01 at stop 1031. Under timetable UCL+ the delay at stop 1121 is 14.2s on average, while it is 9.6s under timetable UCL-. This is due to a higher (planned) headway between vehicles of line 1-B01 and their predecessors between stops 1031 and 1131 under timetable UCL+ than under timetable UCL- (four minutes versus three minutes), subsequently resulting in a slightly higher number of passengers boarding vehicles of line 1-B01 under timetable UCL+, provoking vehicle bunching effects and prolonging the passenger exchange time.

Lastly, figures 6 and 7 reveal another interesting phenomenon: The range between the 25%- and 75%-quantile is significantly larger under timetable UCL- than under timetable UCL+. This is due to the difference in basic intervals between lines 1-B01 and 2-B02 and the fact that vehicles of line 1-B01 are scheduled to depart three minutes after vehicles of line 2-B02 at stop 1031 under timetable UCL-, while they are scheduled to depart four minutes after vehicles of line 3-B01 under timetable UCL+. Consequently, every second vehicle of line 1-B01 is subject to systematically higher headways under timetable UCL-, namely every time no vehicle of line 2-B02 departs. On the other hand, this phenomenon is not present for vehicles of line 1-B01 under timetable UCL+, but instead for vehicles of line 3-B01.

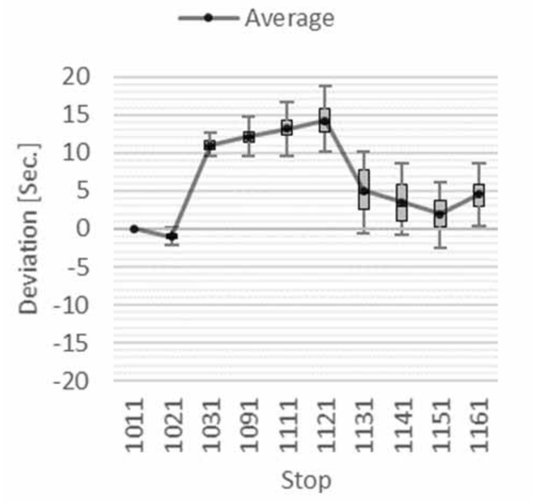

Figure 6: Departure deviation development, line 1-B01, timetable UCL+.

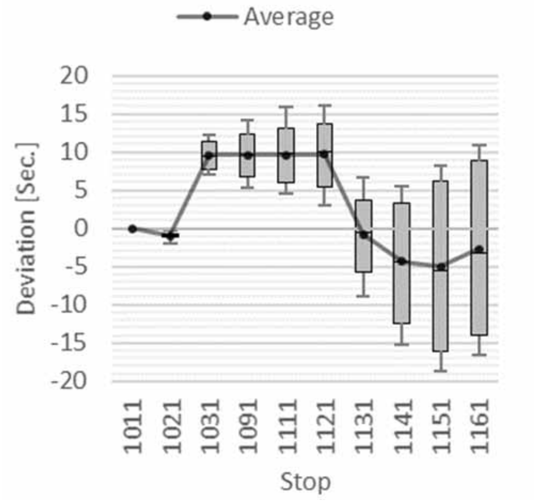

Figure 7: Departure deviation development, line 1-B01, timetable UCL-.

In summary, the results show that the simulation model is suitable to represent bus transit systems. The examined indicators suggest that the model shows plausible behavior under both timetables, even under scarce data.

4 Conclusions

This paper presented a mesoscopic event-based simulation model of bus transit systems. The model consists of sub-models for the representation of the physical network, the logical network, transit vehicles, and the transit provider's operational management. The model is designed to manage on scarce data, especially publicly available time table and network data.

Even though a data-driven evaluation was not feasible without a more comprehensive data basis, the theory-based evaluation based on a model of a small artificial transit network demonstrated the model's plausible behavior.

In further steps, the simulation model first will be applied to models of real world transit systems, before being extended to include light rail transit and further rule-based methos to mitigate disturbances in multimodal transit networks.

Acknowledgments

This material is based in part upon work supported by the National Science Foundation under Grant Nos. I/UCRC IIP-1338922, AIR IIP-1237818, III-Large IIS-1213026, MRI CNS-1532061, MRI CNS-1429345, MRI CNS-0821345, MRI CNS-1126619, MRI CNS-0959985, RAPID CNS-1507611, and U.S. DOT Grant ARI73.

References

[1] 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.

[2] Behrisch M, Bieker L, Krajzewicz D. SUMO – Simulation of Urban Mobility – an Overview. In: Omerovic, A., Simoni, D., Bobashev, G., editors. SIMUL 2011: The Third International Conference on Advances in System Simulation. *Third International Conference on Advances in System Simulation*; 2011 Oct; Barcelona. Barcelona: IARIA. 55-60.

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

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

[6] 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.

[7] 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.

[8] 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.

[9] Liebl F. *Simulation*. 2nd edition. Munich: Oldenbourg Pub.; 1995. 273p.

[10] Lückemeyer G. *A Traffic Simulation System Increasing the Efficiency of Schedule Design for Public Transport Systems Based on Scarce Data* [dissertation]. Department of Computer Science. University of Cologne, 2007.

[11] Lückerath D. *Ein Simulationsmodell für Öffentlichen Personennahverkehr mit regelbasiertem Verkehrsmanagement* [dissertation]. Department of Computer Science. University of Cologne, 2017.

[12] Lückerath D, Ullrich O, Speckenmeyer E. Modeling time table based tram traffic. *SNE*. 2012, 22(2): 61-68. doi: 10.11128/sne.22.tn.10121.

[13] Lückerath D, Ullrich O, Speckenmeyer E. Applicability of rescheduling strategies in tram networks. In: Reichardt R, editor. Proceedings of ASIM-Workshop STS/GMMS 2013. *ASIM-Workshop STS/GMMS*; 2013 Feb; Düsseldorf. Vienna, Austria: ARGESIM/ASIM Pub. 57-64.

[14] Osogami T, Imamichi T, Mizuta H, Morimura T, Raymond R, Suzumura T, Takahashi R, Idé T. Research Report IBM Mega Traffic Simulator. Technical Report, 2012.

[15] 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.

[16] 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.

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

[18] 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.

[19] Ullrich O, Lückerath D. An Introduction to Discrete-Event Modeling and Simulation. *SNE*. 2017, 27(1).