A FRAMEWORK FOR MODELLING THE RIPPLE EFFECT OF CROWDING AT PUBLIC TRANSPORT FACILITIES

Nam Huynh
SMART Infrastructure Facility
Faculty of Engineering and Information Sciences
University of Wollongong Australia
Northfields Avenue, NSW 2522, Australia
nhuynh@uow.edu.au

Johan Barthelemy
SMART Infrastructure Facility
Faculty of Engineering and Information Sciences
University of Wollongong Australia
Northfields Avenue, NSW 2522, Australia
johan@uow.edu.au

ABSTRACT

Many studies can be found successfully applying sophisticated and powerful pedestrian simulation packages for crowd management at public transport stations. However, there are gaps in understanding the ripple effect of crowding and passenger demands at a station to the crowdedness at stations further down a transit line. This paper reports a computational framework that links crowd simulations at transit stations of an urban network. Inherently the framework allows for the simulation of the transferring of mobility characteristics of passengers along a transit line as they travel from one station to another and its impacts on crowdedness at these stations. We also report on a proof of concept demonstrating the application of the framework to a small hypothetical train network. The crowd simulation software used in this study is MassMotion. However, the framework should be adaptable to any crowd simulation package.

Keywords: Crowd simulation, Pedestrian, Transportation, Public transit, MassMotion.

1 INTRODUCTION

Simulation of pedestrian movements has been widely recognised and increasingly used as a powerful tool for effective crowd management in both normal and emergency situations at public places. Pedestrian modelling and simulation therefore has been the subject of many studies over the last two decades. The focus of many of these studies has been on reproducing as realistically as possible the navigation (i.e. route choice) and movements (i.e. collision avoidance) of pedestrians through space. Approaches used in these studies can be classified into macroscopic, microscopic and hybrid. The macroscopic approach treats the crowd as a continuum entity, the dynamics of which is often described by models of fluid mechanics. The approach was pioneered by Henderson (1971). The microscopic approach models each pedestrian as an individual entity whose behaviours are determined by physical and social forces, which describe his/her interactions with other pedestrians and with the environment. The Helbing's social force model (Helbing and Molnar 1995) has been widely adapted in many studies following this approach. Comparisons and evaluations of approaches for modelling crowd dynamics can be found in studies by by Zheng, Zhong, and Liu (2009) and by Bellomo and Dogbe (2011). Hybrid models have been proposed to address the high computational demand in simulating huge crowds individually. A review of existing hybrid models for pedestrian simulation can be found in the study by Ijaz, Sohail, and Hashish (2015). More studies recently have focused on developing computational architectures that can cope with large-scale crowd modelling, including those by Lozano et al. (2007) and by Fernandez et al. (2010).

SpringSim-ANSS 2017, April 23-26, Virginia Beach, VA, USA

Thanks to such high research interest on the different facets of the topic, the last decade has seen the arrival of many packages, both commercial and open source, dedicated to crowd simulation. They are sophisticated and yet efficient enough for dependable analysis of large crowd movements in complex environments. Review of a wide range of crowd simulation packages can be found in studies by Kuligowski, Peacock, and Hoskins (2010) and by Challenger, Clegg, and Robinson (2009).

In fact, many studies have applied these packages for crowd management and/or facility management at public transport stations (Galiza et al. 2009; Wen 2013). However there has been very little research investigating these issues from a train service line perspective. In other words, there lacks an understanding of what platform crowding will be like at station along a service line as a result of the transferring of passenger mobility characteristics from previous stations. These accumulative effects may become significant where mobility-reduced individuals comprise a considerable portion of the passengers. Such understanding (and even better, predictive capability), combined with the knowledge of nominal passenger demand along a train service line, informs planners more accurately the real demand a train would face at the stations, and thus facilitates the building of a more robust and reliable timetable for the line.

This paper reports a framework that efficiently links the simulation model of passenger movements at stations along service lines of an urban train network. A primary task of the framework is coordinating the execution of simulation of passenger movements at each of the stations following the chronological order of events that happen at these stations. In this study these events are the arrival of trains at any one of the stations. This approach allows the composition of passengers on board a train at a station to be determined explicitly (by means of the crowd simulation of their movements within the station) and to be fed into the simulation of the passenger movements at a subsequent station down the line. As a result, this framework is among very few of its kind, if not the first, that enable the simulation of (i) the transferring of mobility characteristics of passengers as they travel from one station to another along a train line, (ii) the impacts of such dynamics on passenger build up at these stations, and (iii) the effects the resulting crowdedness has on the movement of passengers at these stations.

While somewhat similar studies can be found elsewhere (for example by Srikukenthiran 2015), the contribution of this framework lies in the detailed description of an algorithm used for coordinating the simulation of passenger movements at each station, as well as the fact that it was developed from the perspective of a user of crowd simulation softwares. The latter point is critical because the framework does not require access to the source codes of the softwares. Importantly the Python implementation of the framework, which is available for download at https://github.com/smart-facility/LinkinCrowd, greatly enhances its reproducibility.

The remaining of the paper details the architecture of the framework, including descriptions of its main elements and a flow chart for preparing, executing crowd simulations at a station and feeding results to the simulation at the next station. We also report on a proof of concept that applies the proposed framework to simulate the passenger movements within and across five hypothetical train stations, serviced by four train lines and twelve train services. It should be noted that the proof of concept serves purely as a demonstrative application of the framework. The framework was designed with a focus on scalability and transferability, thus should be capable to effectively handle the size and complexity of a real urban train network. Please also note that while the software MassMotion was used to simulate passenger movements within a train station, the proposed framework is adaptable to any crowd simulation software.

2 FRAMEWORK FOR LINKING CROWD SIMULATION AT TRAIN STATIONS

The architecture of the framework includes a class diagram (a Unified Modelling Language diagram) that describes relations between its primary elements (Figure 1) and a flow chart of its primary computational steps (Figure 2).

2.1 Class diagram of main elements in the framework

The urban train network, represented by class ‘RailNetwork’, comprises a number of train lines and a number of train stations. Attributes of each train line (represented by class ‘TrainLine’) include the line name, the list of stops the train line serves (i.e. the route), the timetable of the line, and the list of train services running on this line. In this framework, each stop is represented by class ‘Stop’ and is defined as a platform at a station. The timetable of a train line is represented by a two-dimensional integer array, each row of which contains the arrival time of train services at the corresponding stop along the route of this train line.

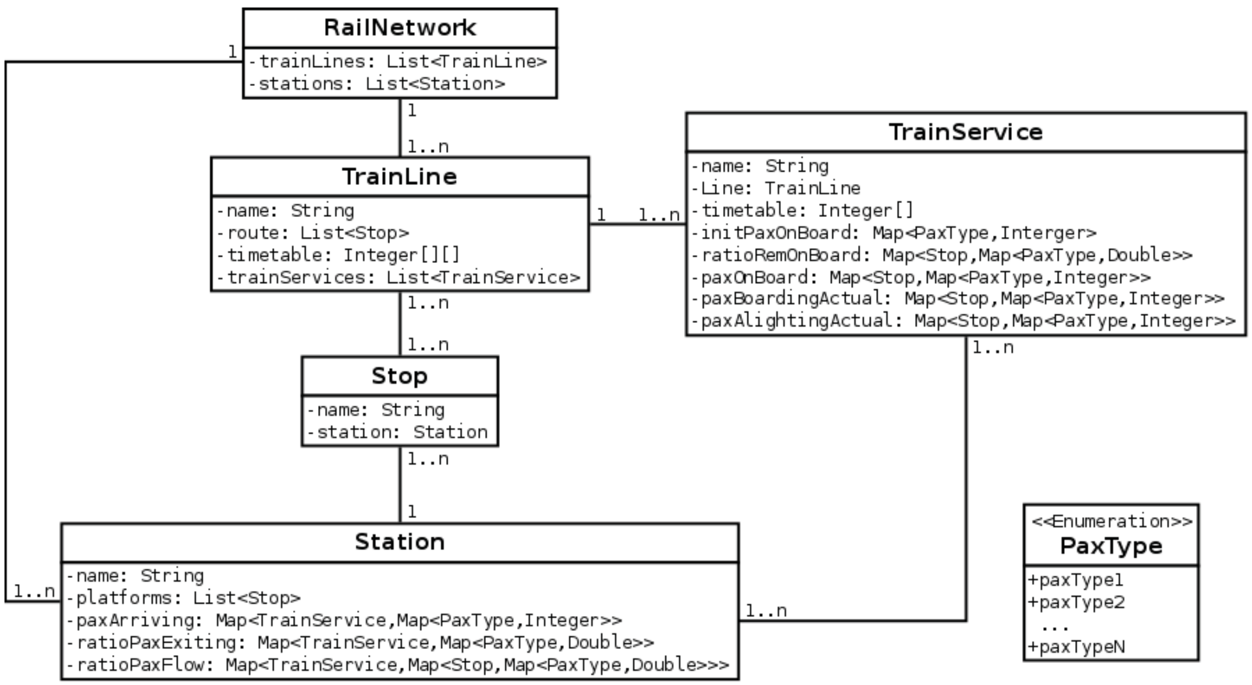

Figure 1: Class diagram of primary elements in the framework.

The number of train services, represented by class ‘TrainService’, belonging to a train line is determined by the number of columns in the train line timetable. Consequently, each column defines the timetable of a train service. Attribute ‘initPaxOnBoard’ is a collection of key-value pairs representing the initial number of passengers (the value) of each passenger type (the key) on board a train service before it enters the simulation. The integer values in this attribute are zero if the first stop of a train line is included in the simulation, and can be non-zero otherwise (e.g. in cases where only a part of the train network is investigated). Attribute ‘ratioRemOnBoard’ is a collection of key-value pairs informing the proportion of passengers not alighting from a train service (the value) at a given stop (the key). The value of this attribute at each stop is a collection key-value pairs detailing the percentage of passengers remaining on board the train (the value) of each passenger type (the key). Structured similarly, attributes ‘paxOnBoard’, ‘paxBoardingActual’ and ‘paxAlightingActual’ represent the number of passengers of each passenger type currently on board, successfully boarding, and successfully alighting from a train service at a stop, respectively. Attributes ‘paxBoardingActual’ and ‘paxAlightingActual’ are to be determined from the crowd simulation of passenger movements at the station to which the stop belongs.

A station, which is represented by class ‘Station’, comprises a number of stops (i.e. platforms). Attribute ‘paxArriving’ of a station represents the number of passengers of each passenger type arriving at the station to board a particular train service. Attribute ‘ratioPaxExiting’ indicates the proportion of passengers of each type who alight from a train service, walk to the gate of the station and exit the simulation. Attribute 'ratioPaxFlow' indicates the proportion of passengers of each type who alight from a train service and walk to (then wait at) each of the other stops (platforms) at the station for boarding a connecting train.

A passenger type, declared within the enumeration 'PaxType', represents passengers having the same set of mobility attributes. Passenger types are used primarily in defining activity patterns of passengers (e.g. arriving at a station and boarding a train) and as a reference to their mobility attributes during the simulation of their movements at a station. Therefore, the attributes of a passenger type are not required to be explicitly declared within the framework, but must be defined in the crowd simulation model for each station. Different pedestrian simulation softwares define mobility attributes of an agent (or passenger) differently. In MassMotion, these attributes include both physical properties and behavioural properties of a passenger. Physical properties include body radius and desired speed and acceleration distributions. Behavioural properties reflect the passenger's preferences to certain geometric conditions of the environment which eventually affect the passenger's route choice. Please refer to the MassMotion software manual (2016) for more detail definition of these mobility characteristics.

Please note in the diagram in Figure 1, the '+' symbols simply represent (public) elements defined in the enumeration 'PaxType'; whereas the '-' symbols represent private attributes of each of the classes. Classes in the diagram are linked with each other by association relationships.

2.2 Flowchart of primary computational steps

As mentioned earlier an event in this study is the arrival of a train service at any station in the network. Therefore, the identification of the next event in Step 1 of the flowchart shown in Figure 2 relies on the timetable of train services arriving at each of the stations and is described in detail below.

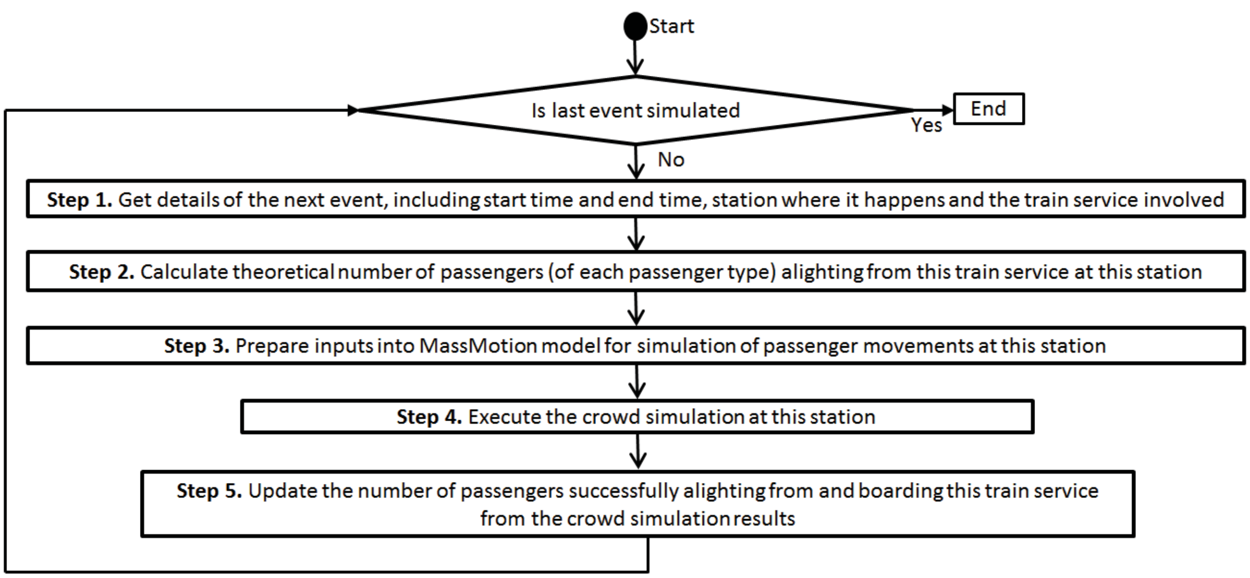

Figure 2: Flowchart of main computational steps.

Let TS^k denote the timetable of train services arriving at a station k. Each row in TS^k is an array containing the time train services from one train line arriving at one of the platforms at the station. These rows are drawn directly from attribute 'timetable' of the relevant train lines. Please note that time values in each array are in ascending order and that the length of these arrays could be different because the number of train services arriving at a platform could be different from one line to another.

Start time of the next train arrival event is determined by equation (1).

$$TS^{k_{min}}_{i_{min},j_{min}} = \min_{k,i,j}\left(TS^{k}_{i,j}\right) \tag{1}$$

where $1 \leq k \leq N_{station}$, $1 \leq i \leq length(TS^k)$, $I_i^k \leq j \leq length(TS_i^k)$

In equation (1), $N_{station}$ is the number of stations in the train network; $length(TS^k)$ and $length(TS_i^k)$ is the total number of rows and the length of the i^{th} row, respectively, in the timetable of station k. I_i^k is the index of the next arrival time available for consideration in the i^{th} row in the timetable of station k. The matrix I is initialised by $I_i^k = 1, \forall k, i$, and then is updated at each iteration by equation (2).

$$I_{i_{min}}^{k_{min}} = j_{min} + 1 \tag{2}$$

$TS_{i_{min},j_{min}}^{k_{min}}$ is the time when the simulation of passenger movements at station k_{min} is resumed, which runs until the next event at this station, which is determined by equation (3).

$$\min_{i,j}\left(TS_{i,j}^{k_{min}}\right) \tag{3}$$

where $1 \leq i \leq length(TS^{k_{min}})$, $I_i^{k_{min}} \leq j \leq length(TS_i^{k_{min}})$.

The train service that triggers this arrival event and the stop it arrives at at station k_{min} can be identified by querying for time $TS_{i_{min},j_{min}}^{k_{min}}$ in attribute 'timetable' of each train line that serves station k_{min}.

The theoretical number of passengers alighting from this train service at this stop (Step 2 in the flowchart) is determined from its attributes 'paxOnBoard' and 'ratioRemOnBoard' as in equation (4).

$$paxAlight^{l,m,n} = paxOnBoard^{l,m,n}\left(1 - ratioRemOnBoard^{l,m,n}\right) \tag{4}$$

where $paxOnBoard^{l,m,n} = paxOnBoard^{l,m-1,n} + paxBoardingActual^{l,m-1,n} - paxAlightingActual^{l,m-1,n}$

$paxAlight^{l,m,n}$ is the number of passengers of type n alighting from train service l at stop m. Similar interpretation applies to $paxOnBoard^{l,m,n}$, $ratioRemOnBoard^{l,m,n}$, $paxBoardingActual^{l,m-1,n}$, and $paxAlightingActual^{l,m-1,n}$. Please note that $paxOnBoard^{l,m,n}$ equals to $initPaxOnBoard^{l,m,n}$ if m is the first stop and that $paxBoardingActual^{l,m-1,n}$ and $paxAlightingActual^{l,m-1,n}$ are determined from the simulation of passenger movements at the previous stop.

Preparing inputs into the MassMotion model for simulation of passenger movements at station k_{min} in Step 3 of the flowchart involves modifying csv files associated with the timetable feature in MassMotion. This feature allows for importing of agent schedules and/or events to be simulated from a series of (programmatically) created csv files. Primary information to be updated in these csv files includes

- Attributes of the event, including its start time $TS_{i_{min},j_{min}}^{k_{min}}$ and the duration (i.e. dwell time of the train at this station)
- Agent schedules, including information on the when, where, how many, and of which profile (i.e. passenger type) agents are created within the model and their destination. Passengers who alight from train service l are created at the arrival platform when the train arrives. Their destination is either the gate of the station or each of the remaining platforms at the station (where they will board a connecting train). The number of alighting passengers heading to the station gate and exiting the simulation is determined by multiplying $paxAlight^{l,m,n}$ by $ratioPaxExiting^{l,n}$. The number of alighting passengers heading to each of other platforms at the station is determined by multiplying $paxAlight^{l,m,n}$ by the corresponding value in attribute 'ratioPaxFlow' of the station. Passengers who arrive at the station to board train service l are created at the station gate and may follow a predefined time-dependent arrival distribution. Their destination is the arrival platform of the train. Their number is informed by corresponding values in attribute 'paxArriving' of the station.
- Agent activities, e.g. waiting on a platform for and boarding a connecting train.

The simulation starting time and end time (which were determined from equations (3) and (4)) are also updated in the project file MassMotion model before it is executed (Step 4) from within the framework. Complete descriptions of elements of the timetable feature of MassMotion, as well as the theoretical background behind its algorithms for agents' navigation and collision avoidance when simulating their second by second movements can be found in the MassMotion software manual (2016).

The number of passengers successfully boarding and alighting from train service l is extracted from MassMotion output files (Step 5) and is saved to attributes 'paxBoardingActual' and 'paxAlightingActual'. They will be used in deciding the number of passengers on board this train in preparing inputs for the simulation at the next stop.

The steps in Figure 2 will be iteratively executed until $\mathrm{I}_\mathrm{i}^\mathrm{k} = \mathrm{length}\left(\mathrm{TS}_\mathrm{i}^\mathrm{k}\right)\ \forall \mathrm{k}, \mathrm{i}$, i.e. all arrival trains have been considered.

3 PROOF OF CONCEPT OF THE FRAMEWORK

The proposed framework was applied to simulate passenger movements within and across five hypothetical train stations serviced by four train lines. Please note that this proof of concept purely serves as a demonstrative application of the framework and does not reflect the operations of any real-life train network.

3.1 Model development and assumptions

The geometry of the stations is shown in Figure 3. Please note that the simulation of passenger movements within a train carriage is not in the current scope the framework. Therefore, detailed geometry of train carriages is not included in Figure 3. Passengers after successfully boarding a train (i.e. passing its gates onto the train floors) exit the simulation. Escalators are available in pairs at stations S2, S4 and S5. Escalators of a pair travel in opposite directions.

Table 1: Timetable of train lines in the proof of concept.

line1	P2S1	3:03:00	3:08:00	3:13:00	3:18:00
	P2S2	3:08:00	3:13:00	3:18:00	3:23:00
	P2S3	3:12:00	3:17:00	3:22:00	3:27:00
line2	P1S3	3:01:00	3:04:00	3:07:00	3:10:00
	P1S2	3:05:00	3:09:00	3:12:00	3:15:00
	P1S1	3:07:00	3:11:00	3:15:00	3:18:00
line3	P2S4	3:01:00	3:04:00	3:07:00	3:10:00
	P4S2	3:04:00	3:08:00	3:11:00	3:14:00
	P2S5	3:10:00	3:14:00	3:17:00	3:20:00
line4	P1S5	3:00:00	3:05:00	3:10:00	3:15:00
	P3S2	3:05:00	3:11:00	3:16:00	3:21:00
	P1S4	3:13:00	3:18:00	3:23:00	3:28:00

Timetable of the train lines is shown in Table 1. Each train line has four train services travelling pass three stops. The arrival and departure of a train are not simulated explicitly but via the opening of train gates at its arrival time and closing them at its departure time. The dwell time is assumed to be 45 seconds for all trains arriving at any platform.

With regards to station attributes, this proof of concept assumes that a total of 50 passengers of each type arrive at a station for boarding any train within 6 minutes prior to its arrival at the station. At all stations except for S2, half of the passengers of each type who alight from any train go to the station gate and exit the simulation while the other walk to the other platform and wait for a connecting train. At station S2, 55% of the alighting passengers (from any train) of each type walk to the station gate and exit while 15% of them walk to each of the other three platforms for a connecting train.

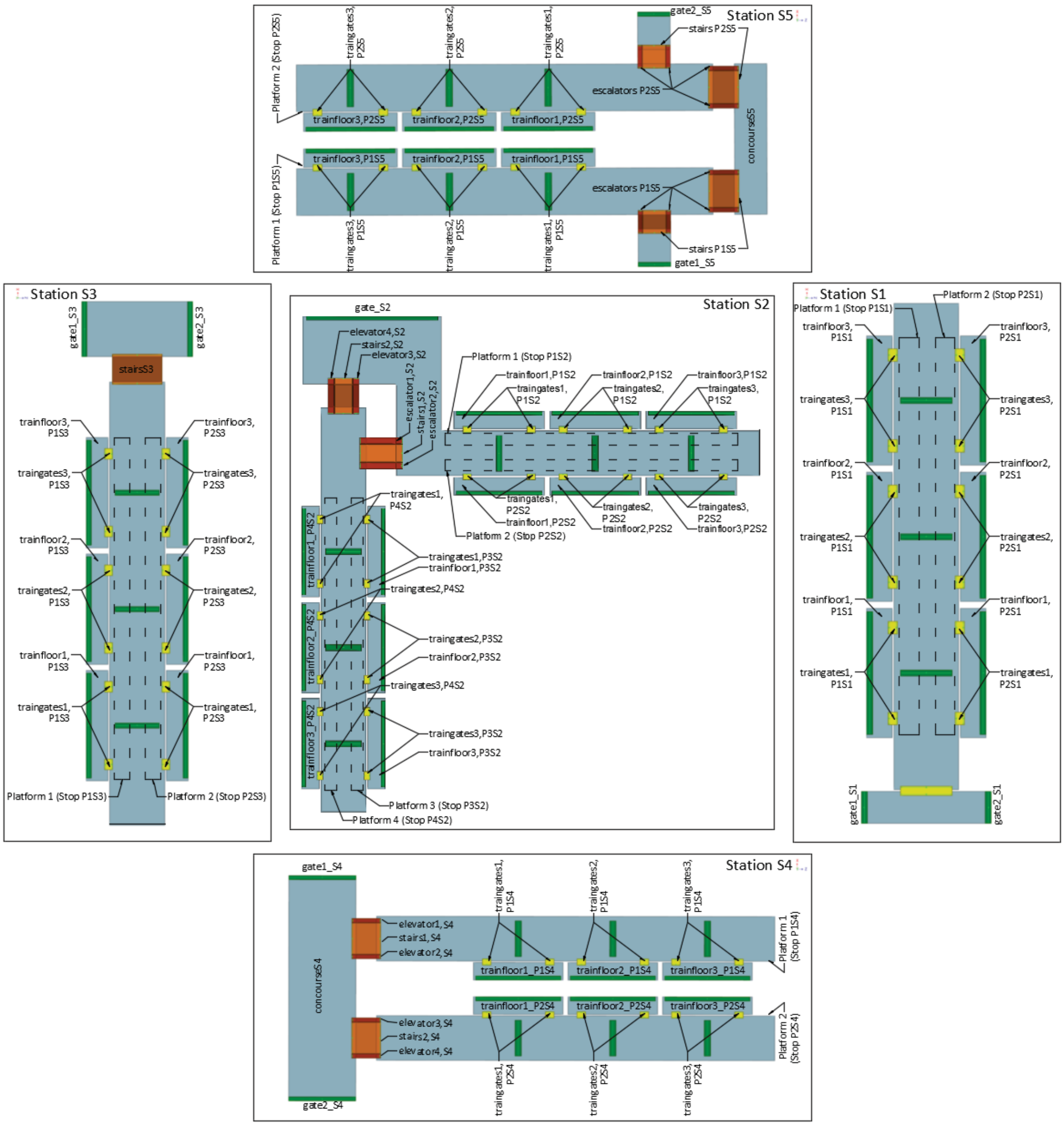

Figure 3: Layout of train stations in the proof of concept.

With regards to train service attributes, the proof of concept assumes that the number of passengers of each type initially on board of any train service is 100. It also assumes that half of the passengers of each type on board of any train service alight upon its arrival at any stop.

With regards to passenger activities, while waiting for boarding a connecting train, passengers spread relatively equally across a platform. They also give way to alighting passengers for about half of the dwell time before boarding. Each boarding passengers is given a token (i.e. a ticket) for the train which they plan to board. The proof of concept makes a simple assumption that passengers who wait for a connecting train will board the next one arriving on their platform.

Table 2: Main mobility characteristics of the three passenger types.

		slimFastActiv	DefaultProfile	fatSlowPassiv
Body radius (m)		0.15	0.25	0.4
Uniform speed distribution	min value (m/s)	1.5	0.65	0.5
	max value (m/s)	3	2.05	1.5
Vertical distance cost (uniform distribution)	min value	0.5	0.75	2
	max value	1	1.25	3

A summary of mobility characteristics that differentiate three passenger types considered in this proof of concept is shown in Table 2. The specified speed distribution is used to assign a desired speed to a passenger when moving freely on a flat open surface. The vertical distance cost is used to differentiate passengers in terms of their preferences when doing route choice. For example, 'fatSlowPassiv' passengers will see stairs as costlier than elevators compared to 'slimFastActiv' passengers and thus would try to avoid them. Please note the above mobility characteristics and their values were chosen purely for demonstrative purposes. Apart from the parameters in Table 2, MassMotion uses a wide range of other parameters to characterise the movements and preferences of an agent. The proof of concept assumes MassMotion's default values for these parameters. Please refer to the MassMotion software manual (2016) for their complete descriptions.

3.2 Simulation results

The order in which the MassMotion model of passenger movements is executed at each of the stations is shown in Table 3. This is the result from applying the algorithm described in Section 2.2 to the timetable of train lines in Table 1.

Table 3: Execution order of Massmotion model of each station.

Index	Triggerred by a train arriving at	Simulation at station	Simulation start time	Simulation end time	Paused by a train arriving at
1	P1S5	S5	3:00:00	3:04:59	P1S5
2	P1S3	S3	3:01:00	3:03:59	P1S3
3	P2S4	S4	3:01:00	3:03:59	P2S4
4	P2S1	S1	3:03:00	3:06:59	P1S1
5	P4S2	S2	3:04:00	3:04:59	P1S2
6	P1S3	S3	3:04:00	3:06:59	P1S3
7	P2S4	S4	3:04:00	3:06:59	P2S4
8	P1S2	S2	3:05:00	3:07:59	P2S2
9	P3S2	S2	3:05:00	3:07:59	P2S2
10	P1S5	S5	3:05:00	3:09:59	P1S5
11	P1S1	S1	3:07:00	3:07:59	P2S1
12	P1S3	S3	3:07:00	3:09:59	P1S3
13	P2S4	S4	3:07:00	3:09:59	P2S4
14	P2S2	S2	3:08:00	3:08:59	P1S2
15	P4S2	S2	3:08:00	3:08:59	P1S2
16	P2S1	S1	3:08:00	3:10:59	P1S1
17	P1S2	S2	3:09:00	3:10:59	P4S2
18	P2S5	S5	3:10:00	3:13:59	P2S5
19	P1S5	S5	3:10:00	3:13:59	P2S5
20	P1S3	S3	3:10:00	3:11:59	P2S3
21	P2S4	S4	3:10:00	3:12:59	P1S4
22	P1S1	S1	3:11:00	3:12:59	P2S1
23	P4S2	S2	3:11:00	3:11:59	P1S2
24	P3S2	S2	3:11:00	3:11:59	P1S2
25	P1S2	S2	3:12:00	3:12:59	P2S2
26	P2S3	S3	3:12:00	3:16:59	P2S3
27	P2S1	S1	3:13:00	3:14:59	P1S1
28	P2S2	S2	3:13:00	3:13:59	P4S2
29	P1S4	S4	3:13:00	3:17:59	P1S4
30	P4S2	S2	3:14:00	3:14:59	P1S2
31	P2S5	S5	3:14:00	3:14:59	P1S5
32	P1S5	S5	3:15:00	3:16:59	P2S5
33	P1S2	S2	3:15:00	3:15:59	P3S2
34	P1S1	S1	3:15:00	3:17:59	P2S1
35	P3S2	S2	3:16:00	3:17:59	P2S2
36	P2S3	S3	3:17:00	3:21:59	P2S3
37	P2S5	S5	3:17:00	3:19:59	P2S5
38	P2S1	S1	3:18:00	EOT	N/A
39	P2S2	S2	3:18:00	3:20	P3S2
40	P1S1	S1	3:18:00	EOT	N/A
41	P1S4	S4	3:18:00	3:22	P1S4
42	P2S5	S5	3:20:00	EOT	N/A
43	P3S2	S2	3:21:00	3:22	P2S2
44	P2S3	S3	3:22:00	3:26:59	P2S3
45	P2S2	S2	3:23:00	EOT	N/A
46	P1S4	S4	3:23:00	3:27	P1S4
47	P2S3	S3	3:27:00	EOT	N/A
48	P1S4	S4	3:28:00	EOT	N/A

Simulation of passenger movements is first simulated for station S5 because it has the first train arriving across all network, which is at its platform 1 (stop P1S5) at 3:00:00. As another train arrives at the same platform at 3:05:00, the simulation is paused at 3:04:59. This is because in general the total number and composition of passengers that will alight from this new train is unknown, which may be determined from the simulation of its previous stop or may be from a set of initial values (which is in this case). The next simulations take place at station S3 and station S4 because both have a train arriving at the same time at 3:01:00. The simulation at both stations is paused at 3:03:59 because there is one train arriving at platform 1 of station S3 (stop P1S3) and one train arriving at platform 2 of station S4 (stop P2S4) at 3:04:00. Subsequent executions of the passenger movement simulation at each station and their triggering event (as described in the remaining of Table 3) can be explained in the same manner. 'EOT' in Table 3 denotes end time of the simulation triggered by the arrival of the last train at any station. For example, the last train arrives at station S1 is at platform 2 at 3:18:00, which triggers the simulation of passenger activities at this station. The end of this simulation cannot be determined from the arrival time of the next train but by a predefined value, which is set as 3:30:00 for this proof of concept. This is also the end time of the last simulation at other stations.

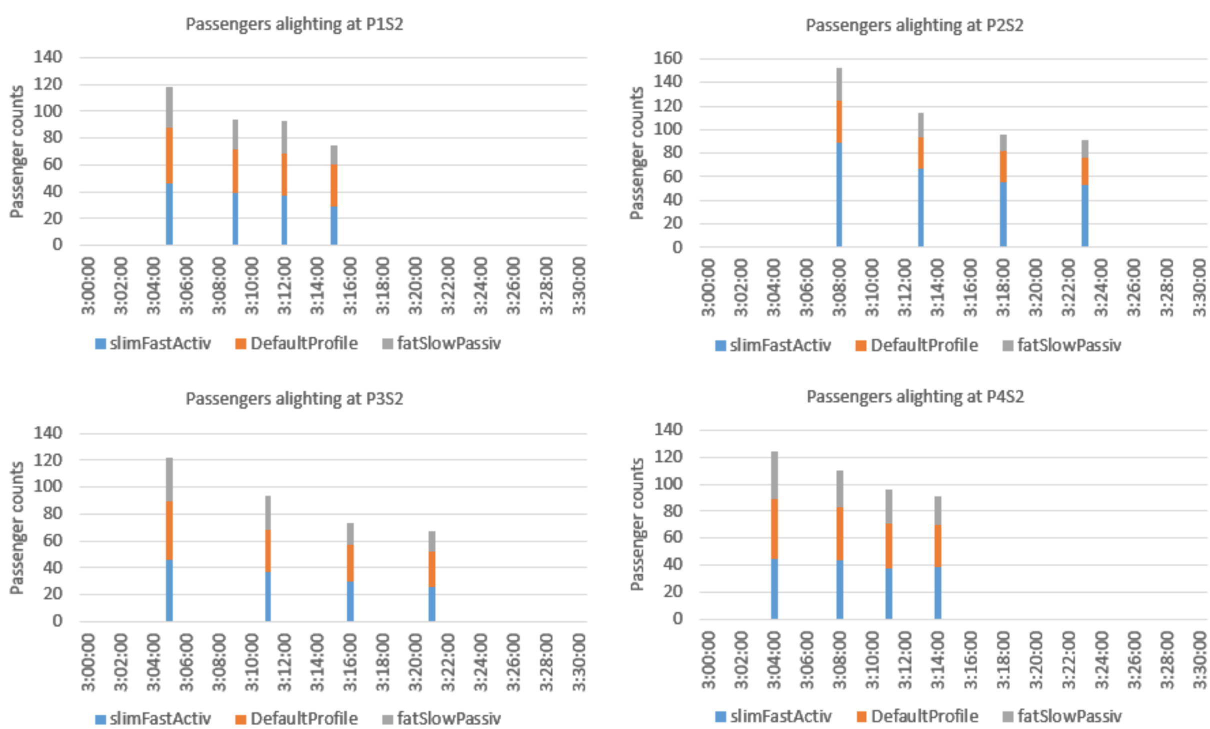

Figure 4: Passengers alighting at station S2 in baseline scenario.

In this proof of concept, station S2 serves as an interchange rail station which all train lines pass and where passengers arrive for transferring to a connecting train. Therefore, its simulation results would best demonstrate the impacts on the crowdedness and passenger experience at a station due to passenger demand and activities from other parts of the network, which the framework proposed in this study aims to achieve. To better illustrate and provide a comparative understanding of the trans-station effects of passenger demand and activities to crowdedness at station S2, we simulated another scenario (referred to as '90%Slim' from now on) of the proof of concept. Differences between this new scenario and the original settings are summarised below. It shall be noted that none of the changes was applied to attributes of station S2 nor attributes of train services relating to it.

- The total number of passengers arriving at stations S1, S3, S4 and S5 for boarding any train service remains 150. However, 90% of them are of passenger type 'slimFastActiv'. The remaining 10% are equally split between the other two passenger types.

- The total number of passengers initially on board a train service arriving at stations S1, S3, S4 and S5 remains 300. However, 90% of them are of passenger type ‘slimFastActiv’. The remaining 10% are equally split between the other two passenger types.

The total number of passengers successfully alighting from a train at each platform at station S2 throughout the simulation of the proof of concept is shown in Figure 4 and Figure 5. Please note that passengers alighting at platforms P1S2, P2S2, P3S2 and P4S2 are from trains on lines ‘line1’, ‘line2’, ‘line3’ and ‘line4’, respectively. The higher proportion of ‘slimFastActiv’ passengers arriving at station S2 from all train lines in scenario ‘90%Slim’ compared to the ‘baseline’ scenario verifies that the framework satisfactorily manages and transfers passenger characteristics across stations of a train network.

More interestingly, the total number of passengers alighting at S2 from each train line is consistently and considerably higher in scenario ‘90%Slim’ compared to the ‘baseline’ scenario. This could be attributed to more passengers successfully boarding at the previous stops as well as successfully alighting at S2 (because there are more of them who are agile in scenario ‘90%Slim’). In fact, simulation results also show that the number of passengers successfully boarding a train at S2 is higher in scenario ‘90%Slim’ than in the baseline, and most of them are ‘slimFastActiv’, as demonstrated in Figure 6.

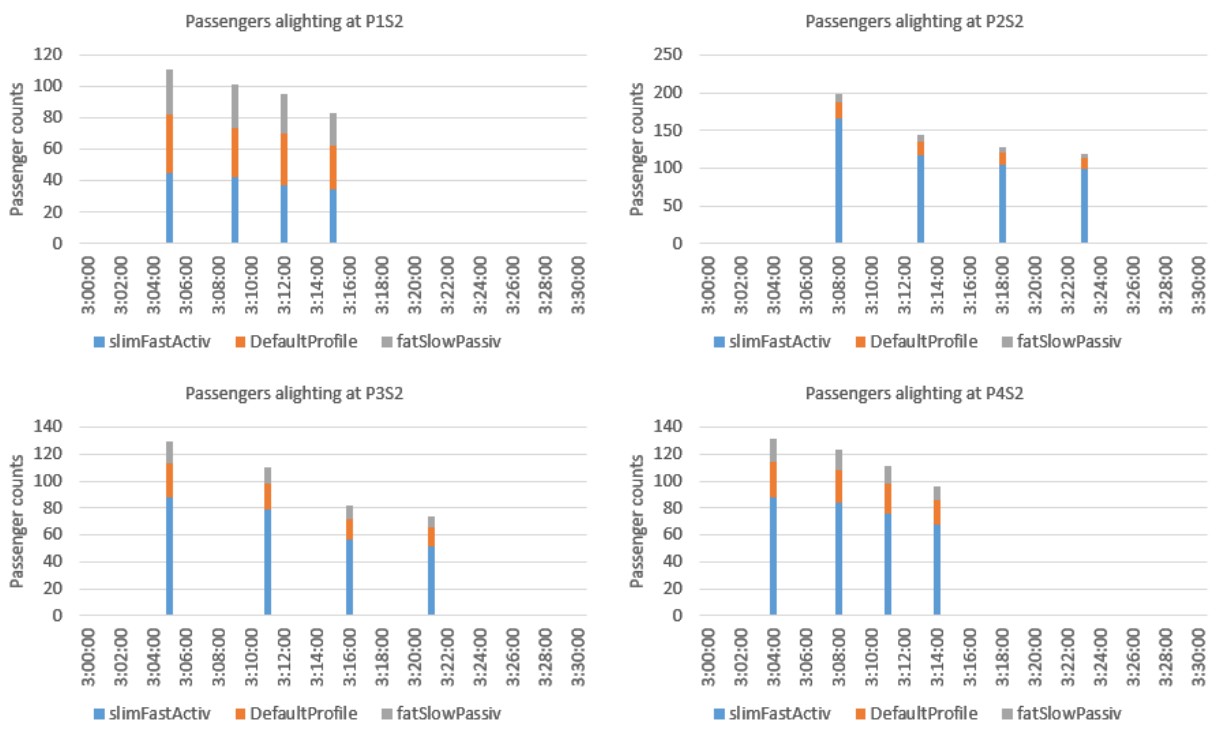

Figure 5: Passengers alighting at station S2 in scenario ‘90%Slim’.

Staircases and escalators were introduced into the presented model partly to enhance the sophistication of the proof of concept. More importantly they help highlight the capability of the framework in picking up the ripple effect of passengers’ behaviours and attributes to the built environment in different settings. More specifically, one implication of having more ‘slimFastActiv’ passengers at S2 is that the stairs at the station tend to be more occupied. Analysis of the number of passengers using stairs and escalators shows that 38.6% of passengers chose stairs over escalators in scenario ‘90%Slim’ whereas the ratio is only 29.3% in the baseline scenario.

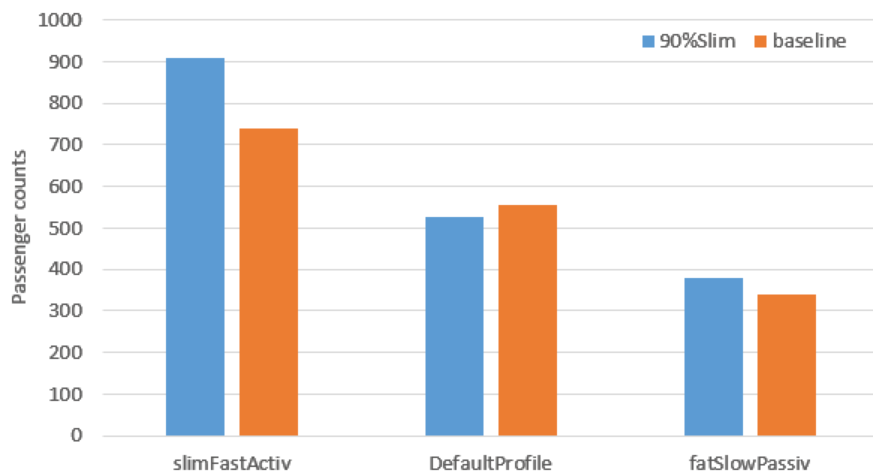

Figure 6: Passengers boarding at station S2.

4 CONCLUSIONS

This paper has presented a computational framework that links crowd simulation at transit stations. The proof of concept that applied the framework to simulate the passenger movements within and across a number of stations of a hypothetical urban train network illustrated the framework's capability to capture the ripple effects of crowdedness and passenger demands at stations along a transit line.

At the heart of the framework is an algorithm for coordinating the simulation of passenger movements at each of the stations, ensuring that results from the simulation at a station (e.g. the number of passengers successfully boarding and alighting) are properly and timely fed into the simulation at subsequent stations. Inherently, the algorithm for such coordination relies on a given timetable of train lines servicing the stations being simulated. However, the timetable can be revised on-the-fly to reflect changes in arrival time of a train at a station (e.g. from a transit line operation model). While such on-the-fly modification of the timetable is not yet in the current implement of the framework, only minor changes would be required to make this possible. The framework can then be helpful in capturing the mutual impacts of passenger build up and train operations and any amplification of delay at stations further down the line. Another suggestion for future development is the inclusion of the geometry of train carriages into the crowd model of a station, providing a seamless simulation environment of passenger movements. Apart from more realistically representing the train station environment, such inclusion would allow for more accurate simulation and estimates of the number of passengers boarding and alighting from a train. The class diagram of the framework would be revised to include carriage layout as an attribute of train services.

Among challenges in applying the framework to simulating passenger movements at stations of an urban train network is the availability of data required as input into the framework. Most challenging would be the recording of the flow of passengers alighting from each train service to destinations within a station (e.g. station exits and other platforms). While automatic ticketing systems (e.g. the Opal card system deployed in New South Wales, Australia) can help partly with the quantities, they are unable to inform the behavioural and physical attributes of the passengers. Recent advents in image processing algorithms which efficiently extract passenger flow from video records from security cameras at train stations may provide a way forward for such a challenge.

REFERENCES

Bellomo, N., and C. Dogbe. On the Modelling of Traffic and Crowds, A Survey of Models, Speculations and Perspectives. SIAM Reivew, Vol. 53, 2011, pp. 409-463.

Challenger, R., C. W. Clegg, and M. A. Robinson. Understanding Crowd Behaviours: Simulation Tools. U.K. Cabinet Office, 2009.

Fernandez V., F. Grimaldo, M. Lozano, and J. M. Orduna. Evaluating JASON for Distributed Crowd Simulations. In Proceedings of the 2nd International Conference on Agents and Artificial Intelligence 2010, pp. 206-2011. DOI:10.5220/0002729802060211

Galiza, R.Z., I. Kim, L. Ferreira, and J. Laufer. Modelling Pedestrian Circulation in Rail Transit Stations Using Micro-simulation. In: Proceedings of the 32nd Australiasian Transport Research Forum, Auckland, New Zealand, 2009.

Helbing, D., and P. Molnar. Social Force Model for Pedestrian Dynamics. Physcial Review E, Vol. 51, 1995, pp. 4282-4286.

Henderson, L. F. The Statistics of Crowd Fluids. Nature, Vol. 229, 1971, pp. 381-383.

Ijaz, K., S. Sohail, and S. Hashish. A Survey of Latest Approaches for Crowd Simulation and Modelling Using Hybrid Techniques. 17th UKSIM – AMSS International Conference on Modelling and Simulation, Cambridge, UK, 2015.

Kuligowski, E. D., R. D. Peacock, and B. L. Hoskins. Technical Note 1680, A Review of Building Evacuation Models, 2nd Ed. National Institute of Standards and Technology, U.S. Department of Commerce, 2010.

Lozano, M., P. Morillo, D. Lewis, D. Reiners, and C. Cruz-Neira. A Distributed Framework for Scalable Large Scale Crowd Simulation. Lecture Notes in Computer Science, Vol. 4563, 2007, pp. 111-121.

MassMotion User Manual, http://www.oasys-software.com/media/Manuals/Latest_Manuals/MassMotion_Flow.pdf. Accessed August 1, 2016.

Srikukenthiran, S. Integrated Microsimulation Modelling of Crowd and Subway Network Dynamics for Disruption Management Support. PhD Thesis, University of Toronto, Canada, 2015.

Wen, K.C. A Dynamic Simulation of Crowd Flow in Taipei Railway and MRT Station by Multi-Agent Simulation System. Urban Planning and Design Research, Vol. 1, 2013, pp. 59-68.

Zheng, X., T. Zhong, and M. Liu. Modelling Crowd Evacuation of a Building Based on Seven Methodological Approaches. Building and Environment, Vol. 44, 2009, pp. 437-445

AUTHOR BIOGRAPHIES

NAM HUYNH is an Honorary Research Fellow at the SMART Infrastructure Facility, University of Wollongong Australia. He has vast experience in modelling and simulation for transport planning and operations research. He has been chief investigator in various commercial research projects in urban transport planning and rail logistics in collaboration with both government agencies and private sectors in New South Wales (Australia). His research interests include the use computational modelling for urban transport planning, operations research, and intelligent transport systems (ITS). His email address is nhuynh@uow.edu.au.

JOHAN BARTHELEMY is an Associate Research Fellow at the SMART Infrastructure Facility, University of Wollongong Australia. He has vast experience in agent based modeling. During his PhD he developed the foundations of a parallelized micro-simulation platform for population and mobility behaviour and applied it to the Belgian context. At SMART he applies his experiences in agent-based simulations and high performance computing to develop new models and improve the scalability of existing ones. His email address is johan@uow.edu.au.