EasyChair Preprint

№ 2065

Enabling Crowd Simulations To Handle Ripple Effects Across Transport Networks

Nam Huynh, Johan Barthelemy and Pascal Perez

December 1, 2019

Enabling Crowd Simulations To Handle Ripple Effects Across Transport Networks

Nam Huynh[1], Johan Barthélemy[1] and Pascal Perez[1]

[1] SMART Infrastructure Facility, University of Wollongong, NSW 2522, Australia
pascal@uow.edu.com

Abstract. Pedestrian modelling and simulation has been vastly researched over the last decades, resulting in the emergence of a wide range of software packages that are sophisticated and yet computationally efficient enough for reliable reproducing and investigations of pedestrian movements through complex environments. Many studies can be found successfully using these powerful tools for crowd management and/or facility 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 a number of transit stations of an urban network. Inherently the framework allows for the simulation of the transfer of passengers along a transit line as they travel from one station to another and its impacts on crowdedness at these stations. The framework should be adaptable to any crowd simulation package. The open source Python implementation of the framework and of the proof of concept would help enhance their reproducibility and be useful to both researchers and practitioners.

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

1 Introduction

Pedestrian simulation 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 and movements of pedestrians through space. Approaches used 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. This was pioneered by Henderson [1,2] and Hughes [3,4]. The microscopic approach models each pedestrian as an individual entity whose behaviors 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 [5] has been widely adapted in many studies following this approach. Comparisons and evaluations of approaches for modelling crowd dynamics can be found in [6-8]. Hybrid models have been proposed to address the high computational demand

in simulating huge crowds. A review of existing hybrid models can be found in [9]. A wide range of commercial crowd simulation packages is detailed in [10,11].

While many studies have successfully applied these packages for crowd management at public transport stations [12-15], there has been 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 transfer of passenger from previous stations. These accumulative effects may become significant where mobility-reduced individuals comprise a considerable portion of the passengers. Such understanding, combined with the knowledge of the nominal passenger demand along a train service line, informs planners more accurately about the real demand at the stations. Consequently, this facilitates the building of a more robust and reliable timetable.

This paper reports a framework that 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 arrivals of trains at any one of the stations. This approach allows the composition of passengers boarding a train at a station to be determined explicitly 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 enables the simulation of (i) the transferring 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. Importantly the Python implementation of the framework, which is available for download at https://github.com/smart-facility/LinkinCrowd, greatly enhances its reproducibility. The commercial software MassMotion was used as part of the framework to account for the simulation of pedestrians at trains stations. The Python implementation therefore was written to interact with specific features within MassMotion. The framework itself however should be adaptable to any crowd simulation software.

2 Framework for Linking Crowd Simulation at Train Stations

The class diagram in Figure 1 describes relations between primary 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.

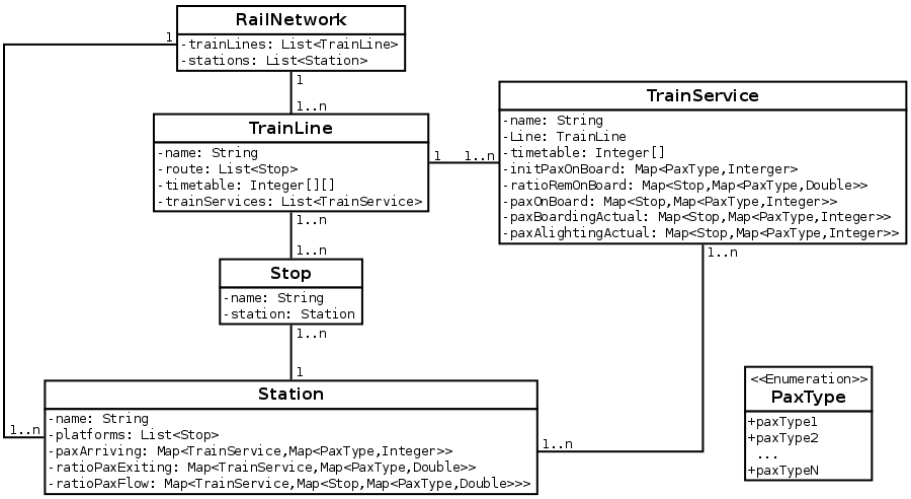

Fig. 1. Class diagram of primary elements in the framework.

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 software defines mobility attributes of an agent (or passenger) differently. In MassMotion, for example, these attributes include both physical properties and behavioral properties of a passenger. Physical properties include body radius and desired speed and acceleration distributions. Behavioral properties reflect the passenger's preferences to certain geometric conditions of the environment which eventually affect the passenger's route choice.

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.

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.

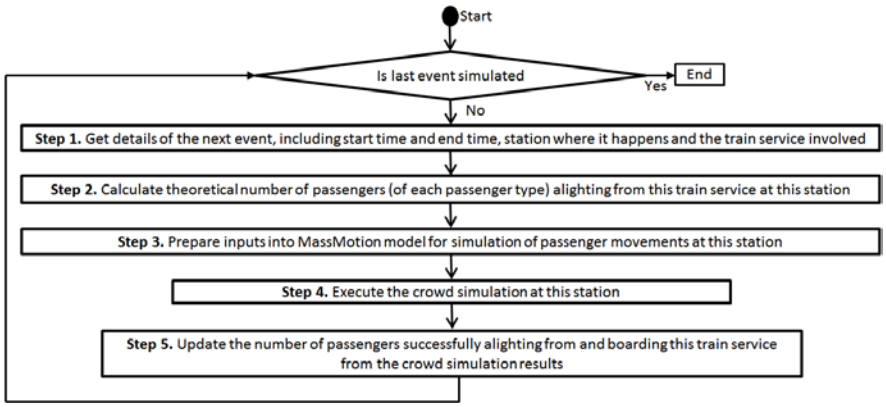

Fig. 2. Flowchart of main computational steps.

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

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

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

In equation (1), $N_{station}$ is the number of stations in the train network; $length(TS^k)$ and $length\left(TS_i^k\right)$ is the total number of rows and the length of the ith

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

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).

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

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 created csv files. Primary information to be updated in these csv files includes

- Attributes of the event, including its start time $TS_{i_{min},j_{min}}^{k_{min}}$ $TS_{i_{min},j_{min}}^{k_{min}}$ 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 [16]. 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.

Steps in Figure 2 are iteratively executed until $I_i^k = length(TS_i^k) \forall k, i$, i.e. all arrival trains are considered.

3 Conclusions

This paper has presented a computational framework that links crowd simulation at transit stations with each other. 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 de-

ployed in New South Wales, Australia) can help partly with the quantities, they are unable to inform the behavioral 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

1. Henderson, L. F. The Statistics of Crowd Fluids. Nature, Vol. 229, 1971, pp. 381-383.
2. Henderson, L. F. On The Fluid Mechanic of Human Crowd Motion. Transportation Research, Vol. 8, 1975, pp. 509-515.
3. Hughes, R. L. A Continuum Theory for the Flow of Pedestrians. Transportation Research Part B: Methodological, Vol. 36, 2002, pp. 507-536.
4. Hughes, R. L. The Flow of Human Crowds. Annual Review of Fluid Mechanics, Vol. 35, 2003, pp. 169-183.
5. Helbing, D., and P. Molnar. Social Force Model for Pedestrian Dynamics. Physcial Review E, Vol. 51, 1995, pp. 4282-4286.
6. 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.
7. 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
8. Zhou, S., D. Chen, W. Cai, L. Luo, M. Y. H. Low, F. Tian, V. S. H. Tay, D. W. S. Ong, and B. D. Hamilton. Crowd Modelling and Simulation Technologies. ACM Transactions on Modelling and Computer Simulation, Vol. 20, 2010, pp. 20:1-20:35.
9. 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.
10. 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.
11. Challenger, R., C. W. Clegg, and M. A. Robinson. Understanding Crowd Behaviours: Simulation Tools. U.K. Cabinet Office, 2009.
12. 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.
13. King, D., S. Srikukenthiran, and A. Shalaby. Using Simulation to Analyze Crowd Congestion and Mitigation at Canadian Subway Interchanges Case of Bloor-Yonge Station, Toronto, Ontario. In: Transportation Research Record: Journal of the Transportation Research Board, No. 2417, Transportation Research Board of the National Academies, Washington, D.C., 2014, pp. 27–36.
14. Hoy, G., E. Morrow, and A. Shalaby. Use of Agent-Based Crowd Simulation to Investigate the Performance of Large-Scale Intermodal Facilities – Case Study of Union Station in Toronto, Ontario, Canada. In: Transportation Research Record: Journal of the Transportation Research Board, No. 2540, Transportation Research Board of the National Academies, Washington, D.C., 2016, pp. 20–29.

15. 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.
16. MassMotion User Manual, http://www.oasys-software.com/media/Manuals/Latest_Manuals/MassMotion_Flow.pdf, last accessed 01/08/2016.