Analyzing Congestion Dynamics in Mass Rapid Transit using Agent-Based Modeling

Briane Paul V. Samson, Cymon Nicollo P. Marcaida, Eloisa Anne B. Gervasio, Ravi D. Militar, and Jerome Ibanez
Center for Complexity and Emerging Technologies, De La Salle University, Manila, Philippines
{briane.samson,cymon_marcaida,eloisa_gervasio,ravi_militar,jerome_ibanez}@dlsu.edu.ph

ABSTRACT

Despite being designed to accommodate only 300,000 passengers, Manila's Metro Rail Transit or MRT3 is currently serving more than 500,000 passengers daily. This over-utilization has resulted into worn down coaches and facilities, and most importantly, passenger delays. In this paper, we present an agent-based model for a Philippine mass rapid transit, particularly MRT3. The agent-based model uses passenger, train, and station agents, and allows the user to run different operational scenarios. Multiple scenarios were simulated, each having different parameters to differentiate their effects to the passenger congestion dynamics of MRT3. It was observed that passenger waiting time is at its minimum when the train speed and number of trains deployed is at their maximum values, 80kph and 20 trains, respectively. However, maintaining their default speed of 50kph and decreasing headway between trains to 1 minute will lead to a drastic increase in passenger waiting and journey times, negatively affecting passenger congestion at the platforms.

CCS CONCEPTS

•**Computing methodologies → Modeling and simulation;** *Agent /discrete models;*

KEYWORDS

agent-based modeling, multi-agent systems, congestion dynamics, mass rapid transit

ACM Reference format:
Briane Paul V. Samson, Cymon Nicollo P. Marcaida, Eloisa Anne B. Gervasio, Ravi D. Militar, and Jerome Ibanez. 2016. Analyzing Congestion Dynamics in Mass Rapid Transit using Agent-Based Modeling. In *Proceedings of Philippine Computing Science Conference, Cebu City, Philippines, March 2017 (PCSC'17),* 6 pages.

1 INTRODUCTION

Traffic congestion due to poor urban planning and overpopulation have caused great inconvenience to the people of Metro Manila, forcing them to use other modes of transportation to get around the city [5]. One of the more popular alternative modes of public transportation is the Metro Rail Transit or MRT3, one of Manila's rail systems. The said line runs for 16.8 km, adjacent to the Epifanio de los Santos Avenue or EDSA, and passes through the city's major leisure and business areas. Originally planned to serve only a maximum 300,000 passengers per day, the MRT3 currently serves more than 500,000 passengers daily, resulting into long queues and disgruntled passengers [6]. Many people blame the Department of Transportation (DOTr), with most passengers saying that the department has done poorly in addressing the plight of the passengers using the MRT3 every day.

Agent-based modeling is a simulation modeling technique that uses smart and decision making bodies called agents that perform real world actions in a particular set of environments [2]. Through the years, the use of agent-based models have increased as it allows the simulation of emergent phenomena with little cost and no risk. Using the MRT3's daily passenger volume data and hourly passenger distribution data, a diffusion agent-based model was created to simulate the operations of the MRT3 during the morning rush hour. Key parameters of the model were then modified to simulate different scenarios to see which positively or negatively impact the average waiting time and average journey time of passengers.

The model was created using NetLogo (see Figure 3), an open source modeling language designed specifically for the creation of agent-based models. It is based on the Logo programming language and represents agents as "turtles" and the world as "patches" [7]. It also comes with an array of analysis tools in the form of graphs and charts that visualize the state of the model in real time.

In the following section, we give an overview of related works on creating models for public transportation. In Section 3, we discuss in depth the model's architecture and agent while in Section 4 we discuss the validation done for the model and its behaviors, the scenarios tested, and their results. In Section 5 and 6, the researchers share their conclusion and future work.

2 MODELS ON PUBLIC TRANSPORTATION

Public transportation can be considered as a complex system as it is always changing and dynamic. There are several types of public transportation that people use on a day-to-day basis. Some own cars and drive every day to work, some use buses while some use trains and trams. One research [1] was focused mainly on the multi-modality of the trams, trains and buses. Traveling can be done through a variety of ways. Some can use the bus exclusively, while some use a combination of the three. The study focuses on finding a possible route given the combination of ways a person can travel to a certain place. Despite the use of trams which deploy a train-like road network, the focus was mainly on roads.

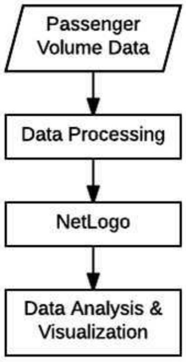

Figure 1: The architectural design of the model and the interacting modules

The work of [8] focuses on Intelligent Dispatching and Systems Design for planning. It mainly goes through the process of monitoring the buses and analyzing the passenger flow. After that, actualizing the automatic dispatch and command on public transportation, ensuring timely vehicle operations, accelerating the operation speed and improving the bus system is performed.

Agent-based models have been thought of as the most accurate of all models in terms of simulating real life events and scenarios. This is because the agents in the model are usually programmed to act either dynamically, by using artificial intelligence to make its own decisions given the scenarios in the environment that it is placed on, or it follows a carefully structured heuristic rule-based approach. A recent study on the Singapore rail system [3] created an agent-based model that investigated overloading in train coaches and overcrowding in platforms. Their model used commuter, train, and station agents that simulate daily operations of a single rail line composed of 29 stations. It was then validated by generating actual trips and travel times using anonymous smart card data. A succeeding work [4] created a model for all Singapore rail lines.

3 DETAILED DESIGN

In this section, we discuss the architectural design, its limitations, and the agents used in the model.

The study used three modules shown in Figure 1. At the beginning of each simulation, the Data Processing module uses the passenger volume data and processes it to be used by the model developed in NetLogo. After the simulation is finished running in NetLogo, the Data Analysis and Validation module analyzes the raw simulation data and generates reports that stakeholders can use for decision-making.

3.1 Data Processing

The model runs on three files: line information, origin-destination probability distribution, and hourly passenger distribution. The line information file simply contains the names of the stations and their coordinates in the world. The passenger probability distribution, on the other hand, determines the origin and destination station of the passengers that will be generated in the model. The hourly passenger distribution is similar to the passenger probability distribution file but instead of containing probabilities, the hourly passenger distribution contains that actual number of passengers that will be generated by the model per hour.

3.1.1 Distribution of Passengers per Origin-Destination. Data for the passenger volume for the month of July 2014 has been provided by the Department of Transportation. For each origin-destination pair of stations, there is data on the total number people that used the station based on the type of ticket, either SJ (Single Journey) or SV (Stored Value).

The data was processed further to generate a file that can be used by the model in NetLogo. First, the distribution of passengers going to the different stations was computed. For each day in July, the percentage of the total number of SJ and SV passengers going to a station from an origin station was derived. For example, if there were 3,400 SJ passengers and 2,578 SV passengers that went to North Avenue station from Ayala station on July 5, and Ayala station had a total of 9,829 total number of passengers that day, the percentage of passengers that entered Ayala station and exited North Avenue station is 60.82%.

After all percentages has been computed from the raw file, they will be stored in a CSV file.

3.1.2 Hourly distribution of passengers. Next, the hourly rate of passengers per station was computed. For the purpose of the study, only 4 time frames included in the AM peak hours were considered. It starts from 6:30AM to 10:30AM. Since the time frames start and end at the 30-minute mark (e.g. 7:30, 10:30), the available Entry rate was divided into half. For example, the Entry rate for North Avenue at 6:00AM-6:59AM is 5,986. Since AM peak begins at 6:30AM, 5,986 will be divided by 2. This will be added to the half of the Entry rate of 7:00AM-7:59AM. Once the passenger count is computed per hour, a Poisson distribution will be computed to maintain stochasticity.

Depending on the time frame being simulated, the appropriate rate will be used for each passenger origin and destination.

3.2 Modeling and Simulation Module

The model in this study was designed and created in NetLogo. It is mainly composed of three types of agents, namely: stations, trains, and passengers.

The passenger agents start at its origin station. It will initially spawn at a specific point on the model, depending on the passenger's origin station. Shown in Figure 2, it will enter the station and then board the platform. Once in the platform, it will wait for a train to arrive and stop at the station. If the train is not full, the passenger will ride the train and journey through the tracks until it has arrived at its destination. It will wait for a train if otherwise. A tournament selection algorithm [4] was adapted to determine whether a passenger will be able to ride a train at the platform. Once at the destination, the passenger will automatically alight the train and eventually exit the station. The model keeps track of the different states that the agent has gone through and the travel time.

Like the passenger agent, train agents also navigate through the world but are only limited to follow the rail track plotted in the model. Trains are either generated or reused, depending if there are trains waiting to be deployed from the line depot which is located at the northernmost part of the line, and navigate through the rails.

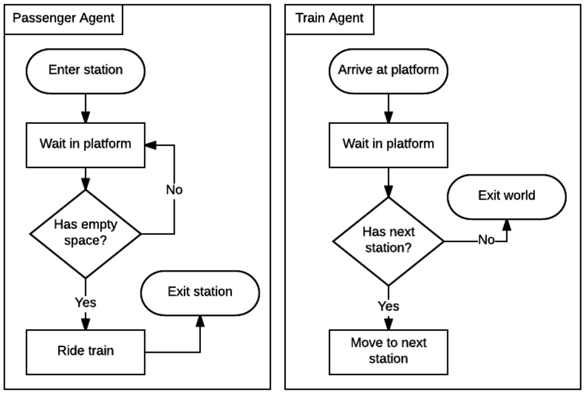

Figure 2: The different actions performed by the train and passenger agents in the model

Sensors placed in the tracks guide which direction the train should go to and which coordinate in the world it should stop in.

Once a train is generated into the model, it is assigned an initial station where it will start its journey. All north-bound trains are generated at Taft Avenue station while south-bound trains start at North Avenue station. Once it arrives at a station, it will stop at the sensor for a few ticks, depending on how the model was initialized. While waiting at the station, passengers can freely enter the train as long as it is not yet full. Every time a train agent is docked in a station, the previous and next stations are tracked to guide its movement. The train will move to the station after it is done waiting for passengers.

When the train reaches the Taft Avenue station, it will turn around heading north-bound until it reaches the last station of the line which is the North Avenue station. Once it reaches the last station, the train will continue heading north until it reaches the end of the rail where it will be hidden from the simulation to be reused later if needed. In case the train will not be reused, it will still remain in the model. Skip trains are outside of the model's scope.

Unlike passenger and train agents, station agents do not move around the model and are assigned a fixed coordinate in the world. Although it does not perform any actions, it is still programmed to interact with the train and passenger agents. There are 26 stations in the model representing the north-bound and south-bound platforms of the 13 MRT3 stations.

The world was manually plotted by changing the colors of the patches in the model to recreate a linear layout of MRT3. The distance between stations was made close to actual. Figure 3 shows how the model looks like in NetLogo. The black area shows the world. The green squares represent the stations and are labeled accordingly on the left. The train tracks are represented by the white lines. The left and right lines model the south-bound and north-bound stations, respectively. Each passenger is generated on the left or right of its initial station, depending on its destination. Passengers are assigned different colors depending on their origin station. They move towards the stations to simulate passengers entering the stations and walking towards the platform. To observe platform congestion, the station platforms change color to red if it has reached at least 70% of its capacity.

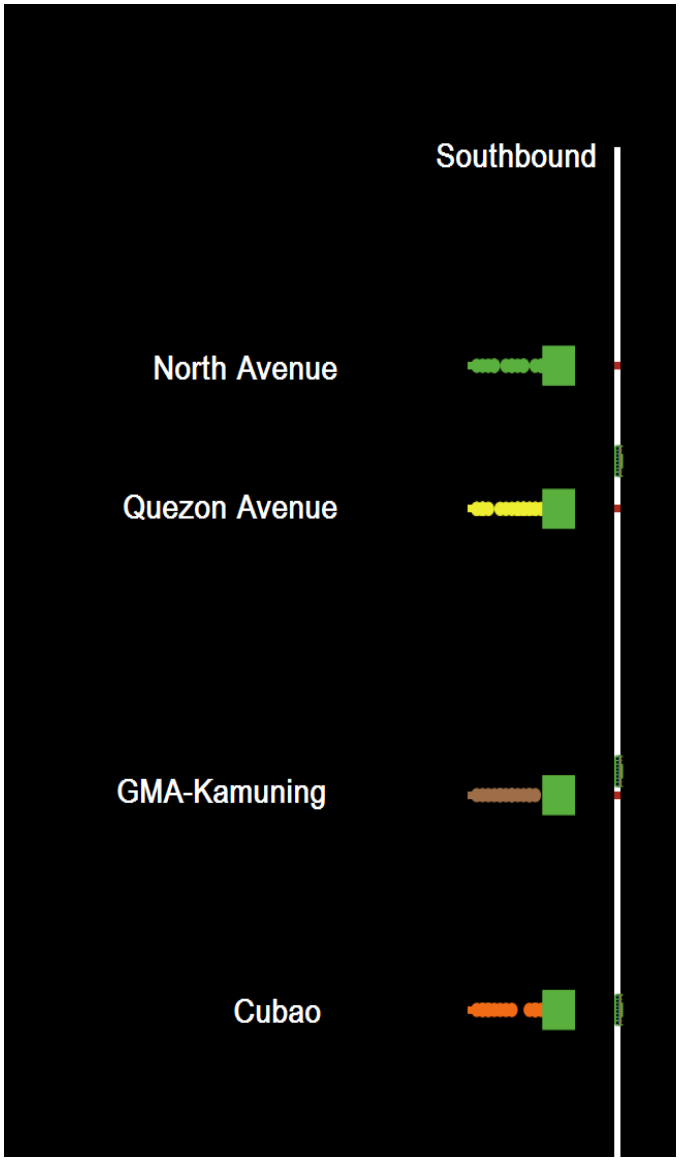

Figure 3: The MRT3 model

3.3 Analysis Module

While running a simulation, a number of metrics are measured and shown to the user.

3.3.1 Passenger-related Analysis. The analysis module for the passenger focuses on a particular agentsfi travel time. The travel time of all of the agents in the model are added and then averaged to come up with an average waiting time. This is an important component in the analysis module because it will help users to know if changing variables in the model have any significant impact on a passengerfis travel time.

3.3.2 Train-related Analysis. The analysis module for trains keeps track of the number trains deployed as well as how congested they are. Trains turn red once it 70% of its capacity has been occupied. Else, it will stay green. Moreover, the number of trains going north and south are tracked as well as the number of trains docked in a station or moving through the rails.

3.3.3 Station-related Analysis. The module keeps track of the congestion rate in a station. Like the trains, a station is congested if it is filled up to 70% of its capacity. It will turn red and will only go back to green once the number of passengers in a station has lessened. Moreover, the station related analysis monitors also keeps track of the number of people in a particular station.

Congestion rate monitors for each of the stations were not implemented because they have their own way of notifying the users if a station is congested. However, it was decided that the overall congestion rate is to be tracked to see how congested the line is as a whole.

4 RESULTS AND ANALYSIS

The model was validated through four methods: passenger distribution validation, passenger state checking, train state checking, and stochastic validation.

4.1 Passenger Distribution Validation

For the passenger distribution validation, the researchers wanted to see whether the model generates similar passenger distribution percentages when compared to the actual passenger distribution percentages. The behavior of the passenger agents were validated by checking if all of the agents follow the correct process assigned to them. Likewise, the train agents were validated by checking whether or not they follow the correct process and whether or not the model generates the correct number of trains specified by the user.

Comparing Figure 4 and Figure 5, it can be seen that only the passenger distribution at Shaw Boulevard matched at 8.5%. The rest of the passenger distribution percentages do not match but are fairly close to each other. Some stations, however, have a slightly large discrepancy with the maximum difference being 5% at Taft Avenue and Guadalupe. The researchers expected that there will be some percentages from the model that will not match the percentages in real life because the hourly passenger distribution count that the model reads from were derived from the yearly averages and not from the daily MRT ridership.

4.2 Passenger State Checking

For the passenger state checking, the number of passengers that have successfully exited the model were counted. Passenger success is determined by checking whether all passenger agents performed all four events as shown in Figure 2. Once a simulation is finished the number of successful passengers that exited the model is then compared to the number of passengers generated during the entire simulation. If the values are equal, it means that all of the passengers successfully went through all four steps. After 21 runs in all scenarios, all passengers generated followed the correct series of steps and were able to successfully exit the model.

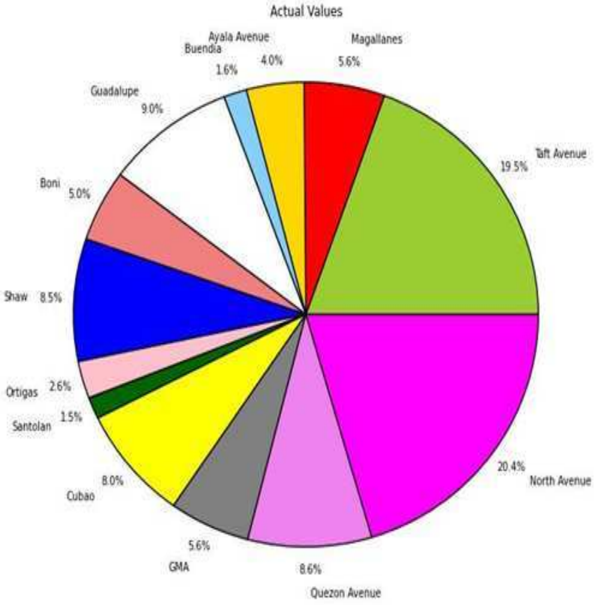

Figure 4: Actual Values

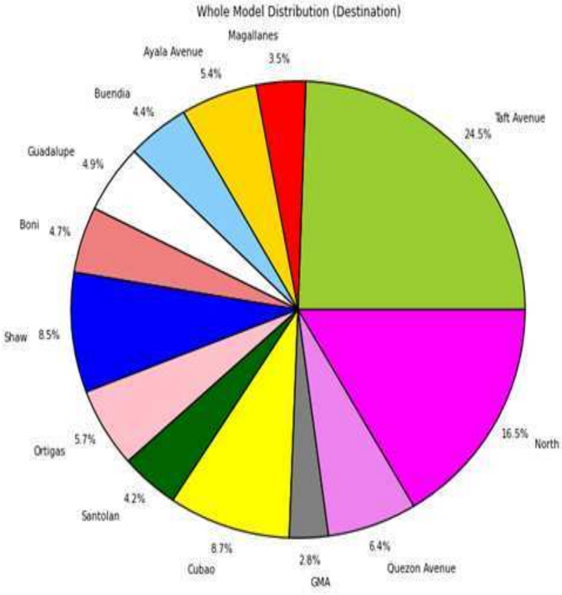

Figure 5: Whole Model Distribution (Destination)

4.3 Train State Checking

The train state checking was done by checking whether or not the model generates the same number of trains specified by the user. Moreover, the trains in the model should not exit the model when they reach their final stations. Instead, the trains should loop towards the next line just like how the trains behave in real life. To do the train validation, we ran the model 21 times and counted the

Variable	Scenario 1 (Default)	Scenario 2 (Max Train Speed)	Scenario 3 (Min Train Interval)
Train Speed	50 km/h	80 km/h	50 km/h
Train Waiting Time	1 tick (20 secs)	1 tick (20 secs)	1 tick (20 secs)
Train Time Interval	10 ticks (200 secs)	10 ticks (200 secs)	3 ticks (60 secs)
Train Required	20 trains	20 trains	20 trains
Maximum Train Capacity	1,182 passengers	1,182 passengers	1,182 passengers

Table 1: Scenario Configurations

	Result 1 (Default)	Result 2 (Max. Train Speed)	Result 3 (Min. Train Interval)
Avg. Journey Time	96.10 ticks (32.03 mins)	73.13 ticks (24.38 mins)	163.09 ticks (54.36 mins)
Avg. Waiting Time	23.04 ticks (7.68 mins)	14.50 ticks (4.83 mins)	115.72 ticks (38.57 mins)

Table 2: Average Journey and Waiting Times after running each scenario

number of trains deployed. Once the simulation is done, it is then compared to the number of trains set by the user. Results show that the model generates the required number of trains in every run regardless of setting.

Aside from checking whether it has gone through all stations, we also validated the trainsfi looping behavior. The only stations that a train can loop to are penultimate stations, the Taft Avenue and North Avenue stations. The number of trains that made a loop were then compared to the number of trains generated by the model. As a result, all of the trains generated followed a looping behavior in all runs regardless of setting.

4.4 Stochastic Validation

After validating the model, the researchers were curious as to what affects the waiting time and journey time of the passengers. First, the researchers ran the model using its default settings. The default settings were derived from the day-to-day settings of the MRT and were confirmed by an MRT3 official. After the model was run, the average journey time and the average waiting time of the passengers was recorded. The model was then run three times with different settings to come up with six non-default runs. The default scenario results served as a baseline to compare the other results to as shown in Table 1.

There are additional six scenarios that the researchers have tested, but two out of six scenarios stood out as it provided the lowest and highest average journey and waiting time, as shown in Table 2. One scenario has the maximum train speed and the other has the minimum train interval. After testing each of the two scenarios three times, to ensure the accuracy of the results, the researchers found out that if train speed was increased to 80 kilometers per hour, the average waiting time of each passenger would be decreased by almost 3 minutes and the average journey time of each passenger would be decreased by 8 minutes.

On the other hand, if the headway between trains would be decreased to 1 minute, both the average journey and waiting times would drastically increase. Since there are only a limited number of trains available, there will come a point when all of the 20 deployed trains are serving only either the north-bound passengers or the south-bound passengers while the passengers on the other side wait much longer because there are no trains available on their side. The slow speed setting in this scenario also contributed to the increase in average journey and waiting times, thus aggravating the congestion in the platforms. So unless the number of available trains increase, it would be counter-intuitive to lessen the headway between trains.

5 CONCLUSION

We were able to develop an initial model of the MRT3 that mimics that movement of passengers during rush hour. We used the July 2014 data provided by the then DOTC to derive the distribution of passengers per origin-destination pair and the hourly distribution during rush hours. After developing the model, we observed the effects of changing certain parameters like train speed and headway between trains to the average journey and waiting times of the passengers.

The passenger waiting time is at its minimum when the train speed and number of trains deployed is at their maximum values, 80kph and 20 trains, respectively. Like the waiting time, the passengersfi journey time is also at its lowest when the number of trains are increased. However, it would be counter-intuitive to suddenly lessen the headway between deployed trains unless the number of available trains increase. Majority of the people who use the MRT claim that the way to make their trips more pleasant is to increase the trains and this test confirms their opinion. This implies that the waiting time and journey time of passengers will be significantly decreased if more trains are deployed. The researchers expect that the waiting time and journey time will be shorter when this is combined with a shorter train interval.

Although the model is successful in recreating the MRT during the AM rush hour, it is nowhere near perfect. The fact that the model only simulates the AM rush hour poses limitations that can only be solved once the other time slots have been put into picture. Moreover, the model lacks certain dynamics such as queuing which could have made the simulations more realistic. Finally, it is important to note the model might be prone to over-fitting since the researchers only had limited data to use for testing.

6 FUTURE WORK

The researchers plan to improve the system by porting it to another programming language, for instance in Python, which will allow the system to expand its capabilities such as including weather and maintenance variables which the researchers believe has also an effect in the passenger congestion dynamics, implementation of queuing theory, and an intuitive tournament system that pits passengers with each other in order to identify which agent will be able to ride the train.

Platform congestion dynamics and train congestion dynamics should be implemented since the model deals with congestion on the MRT as a whole. Congestion inside the platform has a direct effect as to which passengers will be able to board the train while the amount of congestion in the train could have implications as to how long it will take for a passenger to enter or exit the train. It would very beneficial if both the platform and train congestion dynamics be incorporated to provide a more realistic simulation of model.

Furthermore, extending the simulation to other mass rapid transit lines or to other modes of transportation such as PNR, LRT, jeepneys, buses, UV express, and ferries which will allow the users to have an overview of the mass transportation system in a specific area or district.

ACKNOWLEDGMENTS

The authors would like to thank Atty. Michael Sagcal from DOTC for providing the MRT3 data.

The authors would also like to thank Mr. Stanley Tan and Dr. Nelson Marcos for sharing their technical expertise.

REFERENCES

[1] M. Abed, H. Mnasser, and F. Gargouri. 2013. Towards an intelligent information system of public transportation. In *Advanced Logistics and Transport (ICALT), 2013 International Conference on.*

[2] E. Bonabeau. 2002. Agent-based modeling: Methods and techniques for simulating human systems. *Proceedings of the National Academy of Sciences* 99, suppl 3 (2002), 7280–7287. DOI: http://dx.doi.org/10.1073/pnas.082080899

[3] E. Legara, C. Monterola, K. Lee, and G. Hung. 2014. Critical Capacity, Travel Time Delays and Travel Time Distribution of Rapid Mass Transit Systems. *Physica A 406* (February 2014).

[4] E. Legara, C. Monterola, N.B. Othman, and V. Selvam. 2015. A Data Driven Agent-Based Model of Congestion and Scaling Dynamics of Rapid Transit Systems. *Journal of Computational Science* 10 (September 2015), 338–350.

[5] H. Lidasan, I. Espada, and M. De Leon. 2009. A Needs Assessment of Transport Planning and Traffic Management of Local Cities: The Case of the Philippines. *Journal of the Eastern Asia Society for Transportation Studies* 7 (January 2009).

[6] T. Valderama. 2014. Messed-up mass transport system. (2014). Retrieved July 27, 2014 from http://www.manilatimes.net/messed-up-mass-transport-system/74431/l

[7] U. Wilensky. *NetLogo 6.0 User Manual.*

[8] M. Yikui and D. Jun. 2009. Fundamentals of intelligent public transportation dispatching systems planning. In *Computing, Communication, Control, and Management (CCCM), 2009 ISECS International Colloquium on.*