Understanding the Importance of Efficient Visitor Flow within Tokyo Skytree

Eirik Haraldsen, Karl Østr°adt, Reggie Davidrajuh

Electrical Engineering and Computer Science, University of Stavanger,
Stavanger, Norway.
es.haraldsen@stud.uis.no; km.ostradt@stud.uis.no; reggie.davidrajuh@uis.no

***Abstract* - Tokyo Skytree is the tallest freestanding tower in the world. The tower is a popular tourist attraction with two observational decks at an altitude of 350 and 450 meters. This paper aims to understand the importance of the efficient flow of visitors within the Tokyo Skytree. We utilize a discrete mathematical model based on Petri Nets and the tool GPenSIM to simulate the visitor flow. Through simulation, we explore how the flow of visitors change by increasing and decreasing the elevator specifications. The results show that the specifications of the elevators play a crucial role in efficient visitor flow. The model and its implementation presented in this paper are specific to Tokyo Skytree. However, the idea presented in this paper can be easily adapted to model any other tall buildings too.**

Keywords - Tokyo Skytree, Visitor Flow, Petri Nets, Elevator, model, GPenSIM.

I. INTRODUCTION

Tokyo Skytree is a combined broadcasting and observation tower located in the Sumida ward, Tokyo. Tokyo Skytree was completed in 2011 and is currently the tallest tower in the world [1]. The tower has become a popular tourist attraction with over 6 million annual visitors [2]. Visitors are able to enjoy the stunning view over Tokyo city from two different observation decks. The first deck, Tembo deck, is found at 350m and spans across three levels. The three levels include a restaurant and cafe, souvenir shop and a spectacular 360- degree panoramic view of the city. The second deck, Tembo Galleria, is found at 450m and is often referred to as “the world’s highest skywalk”. Tembo Galleria includes a spiral ramp along with the tower that ends up in a more conventional 360-degree observation space.

Toyoko is the venue for the 2020 Summer Olympics (officially known as the XXXII Olympiad Games, also known as Tokyo 2020), scheduled to be held from 23 July to 8 August 2021. Since this event draws huge attendees, we believe the information this paper carries will be interesting to various actors. However, this paper’s main focus is mathematical modeling - the modeling, simulation, and performance analysis with Petri Nets.

II. PROBLEM DEFINITION

The flow of visitors is critical when it comes to the efficiency of the tourist attraction. A bottleneck may stagnate the flow and potentially cause flow congestion. An inefficient flow negatively affects visitors’ satisfaction, which may reduce the number of recurrent visitors and damage the venue’s reputation.

In this paper, we investigate how the elevator specifications impact the flow of visitors. It will be interesting to explore the elevator’s capacity and speed to avoid operational inefficiencies due to congestion. We also aim to identify other potential bottlenecks by changing the deck capacity and amount of visitors. The flow of visitors is modeled using the Petri net; the Petri net model is implemented and simulated with GPenSIM.

A. Observation Decks

Tourists can access two observational decks such as the Tembo Deck and the Tembo Galleria. Tembo Deck spans across floor 350, 345, and 340. There is a cafe accessible from floor 350 and 340. Floor 345 hosts a restaurant and a souvenir shop. In Tembo Galleria, visitors walk up a slope from floor 445 up to floor 450. This slope is called the “air-walk”. Floor 450 is the highest floor a visitor can reach. At an altitude of 450 meters, the view over Tokyo is quite breath-taking. Each floor has a panoramic view of Tokyo as the outer walls are all windows. From both observational decks, landmarks such as Sens¯o-ji Temple or Tokyo Tower can be spotted. On a clear day, even Mt. Fuji is visible from 106 km away [3].

B. Ticket Options

Tokyo Skytree is the most-attended observational experience in the world with between six and seven million visitors annually [2]. Each visitor can choose between two types of admission tickets, Tembo deck and combo ticket. The Tembo Deck ticket is the cheapest option, and only grants access to the Tembo Deck floor. The combo ticket is slightly more expensive, and grants access to both observational floors.

After its opening in 2011, the leading group of visitors was domestic tourists from all over Japan. One of the strategies for attracting more foreign visitors was introducing a fast lane ticket in 2015. The fast lane ticket is only available to foreign tourists and can be purchased without waiting in line. The ticket also includes a separate prioritized elevator queue. Tobu Tower Skytree Co., the tower operator, started conducting sampling surveys of domestic and foreign visitors in 2013. The surveys show an increase of foreign visitor from 6.8% in 2013 to 24.4% in 2018 [4].

C. Elevators

An essential factor in profit maximization for tourist attractions is the flow of visitors. The most significant issue for any observational attraction is how to move people vertically efficiently. The tower operators installed four high-speed, large-capacity elevators to minimize flow stagnation.

Each elevator is capable of transporting 40 people to the Tembo Deck in approximately 50 seconds. The elevators reach a top speed of 36 km/h, making them the fastest large-capacity elevators in Japan [5]. The elevators pick up new visitors on 4th floor and transport them to the top of the Tembo Deck on floor 350. From this floor, it moves down to floor 340 to pick up any departing tourists. The elevator moves down to the exit on the 5th floor and then returns to the entrance on the 4th floor. A separate pair of elevators move similarly between the Tembo Deck and Tembo Galleria observational floors.

D. Process

Given that there are approximately 6.4 million annual visitors and that the attraction is open 365 days a year, there are roughly 17500 daily visitors. All visitors have to buy a ticket and wait in queue for the elevators. It is also safe to assume that foreign visitors always choose the fast lane option. This accounts for 4270 of the daily visitors, considering that 24.4% of visitors are foreign tourists. Visitors in the fast lane are prioritized when boarding passengers into the elevators. Boarding the elevators occurs on the 4th floor and transport visitors to floor 350. The visitors are free to buy beverages at the cafe and walk around the circular floor, although it is preferred to walk in a clockwise direction.

Visitors with the combo ticket can take the elevator up to Tembo Galleria on floor 445, from which they walk up the "air walk" to floor 450. It is preferred to walk in a clockwise direction to the elevators that go down to floor 345. Visitors that did not have tickets to Tembo Galleria take the stairs or escalators down to floor 345, combining the flow of visitors from Tembo Deck and Tembo Galleria. On floor 345, the visitors have the option to enter the souvenir shop and the restaurant. Anti-clock-wise is the only available direction one can walk. An escalator goes down to floor 340 from where the visitors have one final opportunity to visit a cafe before walking clockwise around the floor to stand in queue for the exit elevator. The visitors exit the attraction on the 5th floor. The elevator then picks up more visitors on the 4th floor.

A detailed overview of all the floors in the tower is presented in figure 1.

III. LITERATURE REVIEW

A thorough literature study reveals no similar works on modeling visitor flow on an observational (tall) tower, focusing on elevators. The majority of similar works focus on optimizing visitor flow in large, complex venues like airports, museums, amusement parks, etc.

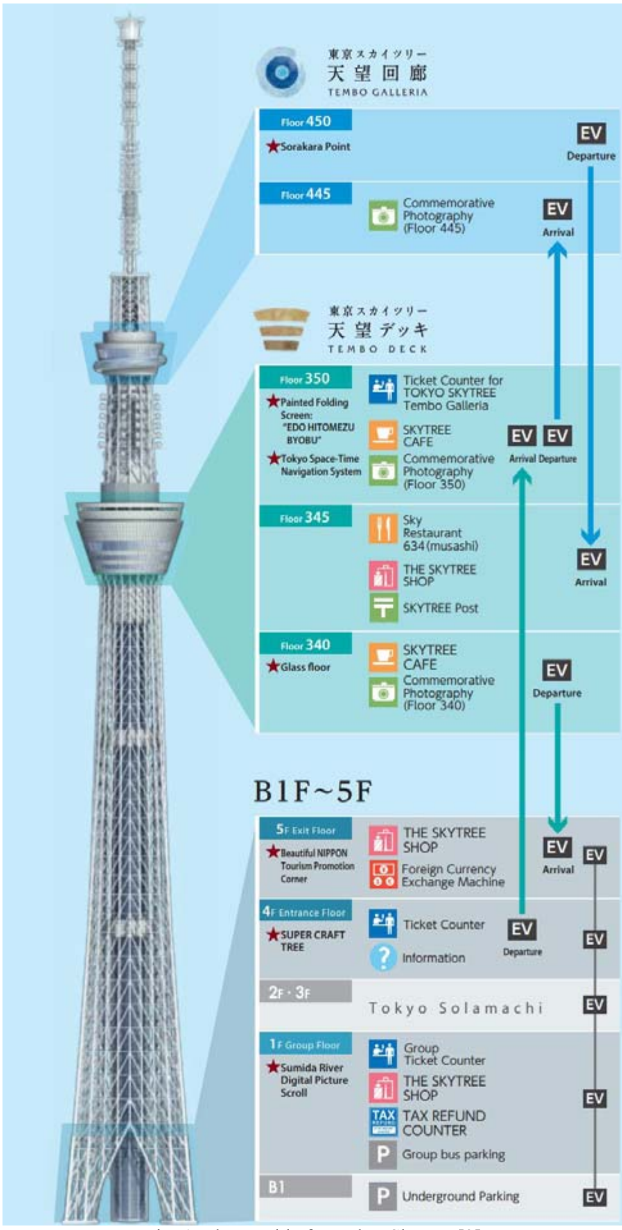

Fig. 1. Floor guide for Tokto Skytree [3]

DOI 10.5013/IJSSST.a.22.01.07 ISSN: 1473-804x online, 1473-8031 print

For example, [6] presents a methodology to improve elevator performance by applying machine learning. Crites et al. [7] present a similar approach, focusing on safety and control issues. Similarly, Ho and Fu [8] uses a learning algorithm to optimize elevator performance.

Some works use Petri Nets (as in the case of our paper) for elevator modeling. Ahmad et al. [9] present a model of an elevator system for control aspects. Lin and Fu [10] focus on optimal scheduling of passenger traffic. Cho et al. [11] are about optimal control of a group of elevators.

In summary, by the literature study, we found some papers focusing on simulating elevator passengers. However, these works tend to focus on optimizing the elevator algorithm to secure the best possible passenger flow and safety control. Davidrajuh [12] presents a toolkit based on GPenSIM, for modeling an elevator; though [12] is only for a single elevator, it served as an inspiration to start the modeling approach of our paper.

IV. PETRI NETS

Due to space limitations, the basics of Petri Nets is described in this paper. For a thorough study on Petri nets, the following textbook is suggested [13].

A. GPenSIM

GPenSIM (General-purpose Petri Net Simulator) is developed by the third author of this paper [14], and is freely available on the website [15]. GPenSIM supports distributing the model logic between the Petri Net ("hard wiring") and processor files ("soft coding") [14]. Hence, all the behaviors that cannot be (or difficult to) put on the Petri Net model are coded in the processor files. This two-way division makes modeling of large systems easier.

In GPenSIM, the Petri Net details (the places, the transitions, and the arcs) are coded in the Petri Net Definition file (PDF). The additional logical conditions are divided into preconditions (to start firing of transitions) and post-actions (after firing completes). And the pre-conditions are coded in the preprocessor files and the post-actions in post-processors files. For further details, the interested reader is referred to [14] and [15].

GPenSIM supports modular model building, which is inevitable for modeling large real-life discrete-event systems [16], [17]. GPenSIM proposes the use of "Petri Modules" for developing large Petri Net models [16] [17].

V. METHOD AND DESIGN

Modeling the Tokyo Skytree in its entirety is a complex and time-consuming task. Hence, the following subsection presents the basic assumptions.

A. Basic Assumptions

Some assumptions were made to simplify the Petri net model and make sure a useful result is generated. We assume the number of daily visitors stays fixed throughout the year. This assumption is made as the data we have is only about the number of annual visitors. In reality, the daily number of visitors will vary depending on the day of the week, holidays and season. We also assume all foreign tourist will buy the fast lane ticket, though the ticket is slightly more expensive.

B. Modular Approach

Since the problem is large and complex, a monolithic model would become too big. Hence, a modular model was developed. The Petri net was divided into multiple modules to get a better overview of the complete system. Each module is based on a different component of the tourist attraction, and is responsible for performing a single task. The arrival module is a generator that simulates the arrival of visitors at a fixed interval. The elevator module is responsible for moving visitors between different floors. The deck module simulates how visitors move through the deck and spend time at different locations. Additionally, we implement a data collection module to log information throughout the simulation. Some modules are connected using an inter-modular connection module, while other modules are detached from the rest of the system.

VI. IMPLEMENTATION

This section presents an overview of the entire model. At first, the purpose of the inter-modular communication (IMC) module is presented, followed by the presentation of all other modules.

A. Overview and the IMC Module

Figure 2 shows the structure of the Tokyo Skytree model. Each block in the figure refers to a separate module, where each module can contain many places and transitions. The four places (pWaiting, pLeave, pTransit1, pTransit2) are a part of the IMC module and is responsible for communication between modules. Figure 2 also shows how the arrival and data collector modules are detached from the system. The place pWaiting is a symbolic place where visitors are waiting to enter the elevator on different floors. Similarly, the place pLeave is a symbolic place for visitors leaving the elevator on different floors. The two transit places are used to simulate the elevator closing the doors and moving to another floor.

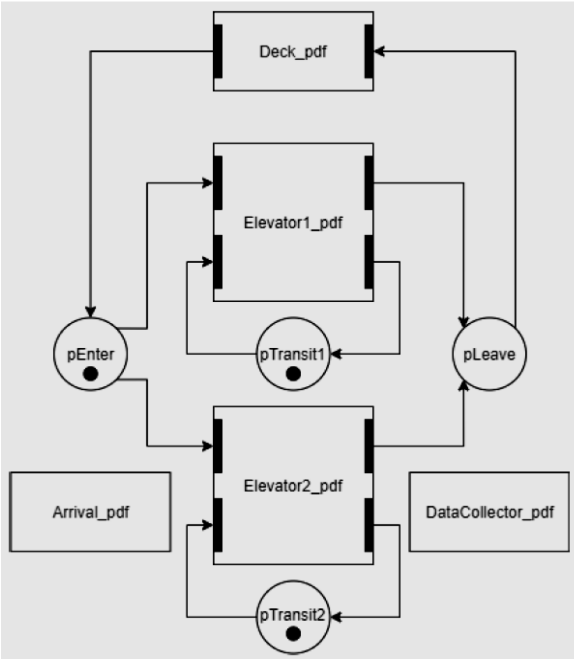

Fig. 2. IMC module

B. Arrival Module

Figure 3 shows the implementation of the arrival module. The model consist of one place (pArrival) and one transition (tArrival), and is responsible for the generation of visitors. The place pArrival is initialized with one token, making tArrival enabled at all times. The transition tArrival fires at a fixed time interval of 5 seconds, simulating the arrival of 720 visitors an hour. Rather than producing one token for each visitor, the arrival module only increments a global arrival counter each time it fires. This strategy was deployed to reduce the total run time of the simulation. The absence of input and output tokens makes this module completely detached from the rest of the system. The post-processor file of tArrival is responsible for maintaining the correct ratio of slow and fast lane tickets. A flag is used to assign every fourth arriving visitor to have a fast lane ticket.

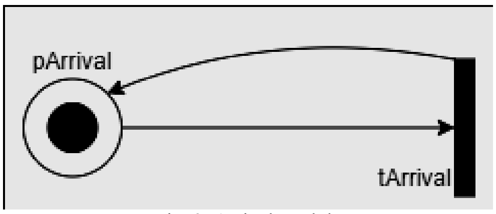

Fig. 3. Arrival module

C. Elevator Module

Elevators are the only sensible mean of transporting visitors to Tembo Deck at an altitude of 350 meters. At the entrance floor, visitors in the fast lane have priority over the slow lane when boarding the elevators. When taking the elevator down, there is no such priority. The elevator has a capacity of 40 people. It takes 50 seconds to reach floor 350, and it is assumed that the return trip takes the same amount of time.

Figure 4 shows how the elevator module is implemented. The module has two places (pInCartX, pReadyX) and five transitions (tEnterX, tLeaveX, tReadyX, tCloseXDoor, tOpenXDoor). The 'X' is replaced by an integer, denoting which elevator the place or transition belongs to. E.g. tEnter1 and tEnter2 belongs to two different elevators. Both elevators have an identical structure and pre/post processing files. It is therefore sufficient to only explain how one of the elevators works.

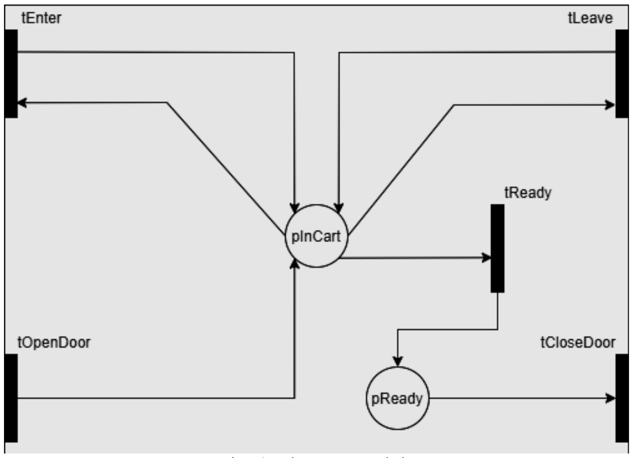

Fig. 4. Elevator module

The transition tEnterX, when fired, moves visitors from the queue into the elevator. Multiple people are moved at once; thus, it can only fire if there are zero visitors in the elevator. Furthermore, the transition can only fire at the entrance floors, never on exit floors. An elevator priority has been implemented to ensure that visitors only enter one elevator at a time when the elevators are on the same floor. tEnterX consumes a token from pInCartX and pWaiting and puts a new token back into both places.

When fired, transition tLeaveX move visitors out of the elevator to the deck or the exit, depending on which floor the elevator is at. This transition cannot fire at a floor where visitors enter the elevator. Tokens are consumed from pInCartX and puts a token into pInCartX and pLeave.

On entrance floors (floor 4 and 340), the elevator is ready to move once visitors are inside. On exit floors (floor 5 and 350), the elevator can only move to the next floor if emptied. If the elevator has been idle at an entrance floor for too long, it automatically closes and moves to the next floor. When tReadyX fires, the elevator priority is given to the other elevator to be filled up. The token in pInCartX is consumed and put into pReadyX. This deactivates tEnterX and tLeaveX, preventing visitors from entering or leaving while the elevator is ready to move.

DOI 10.5013/IJSSST.a.22.01.07 ISSN: 1473-804x online, 1473-8031 print

D. Deck Module

When visitors step off the elevator they arrive at the southern side of the Jumbo Deck floor. The visitors move clockwise through the floor, stopping at each cardinal direction, to enjoy and take pictures of the stunning view over Tokyo. Figure 5 shows the implementation of the jumbo deck module. The module consist of one input transition (tDeckEntrance), one output transition (tDeckExit), four places (pSouth, pWest, pNorth, pEast) and three transitions (tSouthWest, tWestNorth, tNorthEast) connecting the places. An additional transition (tSink) was implemented to avoid accumulation of excess tokens.

Each place in the module refers to a different cardinal section of the Jumbo Deck floor and is initialized with one token. A visitor counter and a map of arrival times are implemented for each place in the module. The visitor counter tracks how many people are currently present, while the map tracks the specific arrival times. The key of each entry in the arrival map is a timestamp, and the value is equal to the number of people arriving at that time. These two variables are used for data collection and precondition logic.

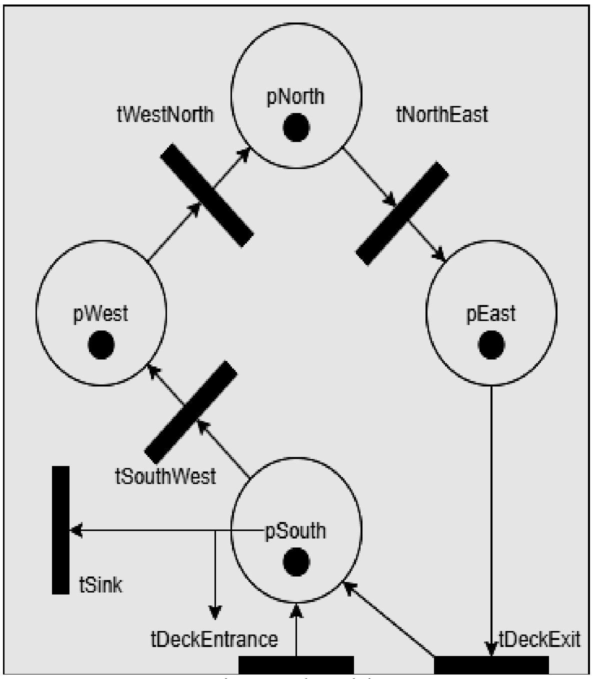

Fig. 5. Deck module

With the current implementation, a visitor spends roughly one hour on this floor. After spending fifteen minutes in the final location, the visitor makes his way to the exit. The output transition tDeckExit uses the same logic as the cardinal transitions to identify when visitors should move. The transition increases a deck exit counter by the number of people moving towards the exit. The elevator module uses this counter to determine how many people should be moved from the deck to the exit on the 5th floor.

E. Data Collector Module

The data collectors' structure is identical to the Arrival module, although it has an entirely different purpose. As its name suggests, the module collects data throughout the simulation. Data collection is achieved by letting tData fire at a fixed interval.

The module is completely detached from the other modules. Conceptually, the module can be thought to be observing or surveilling the simulation and logging the system's state. In retrospect, this allows for measuring the average number of visitors in a location for any given timeframe.

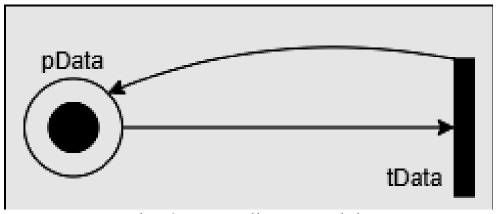

Fig. 6. Data collector module

F. Complete Model and Reproducibility

Due to brevity, this paper only highlights the important parts of the model and the implementation. For reproducibility, the interested reader can see more details and the complete implementation details (GPenSIM code) from the website [18].

VII. TESTING, ANALYSIS AND RESULTS

In this section, we first present our sources for data collection and adapt these parameters for our simulation. Second, the results from a sample run are presented and discussed. Finally, we present the visitor flow simulation results when different parameters are changed.

A. Data Used for Testing

The main sources of information used in this paper are from the official Tokyo Skytree website, a translated version of a Japanese newspaper article, and Toshiba, the company responsible for installing the elevators. Our other sources include the global attractions attendance report by TEA/AECOM and tourist information services [19].

B. Sample Run

A sample run tries to simulate Tokyo Skytree during open hours with real parameters. We assume that the tourist attraction is open from 8:00 to 22:00 with the last admission at 21:00. Thus the simulation should run for 14 hours, and visitors cannot enter the queue or the elevator going up after 13 hours. The time unit is in seconds, so the simulation runs for 60 * 60 * 14 time units. An additional 15 minutes of simulation time is added to simulate visitors lining up in queue before opening.

```
Parameters used for this simulation:
Simulation time in hours:       14
Elevator capacity:      40
Deck capacity:  1000
Close door delay:       25
Automatic close delay:  30
Logging interval in minutes:    10
--------------------
Number of visitors and departed:
Total visitors: 9361
Departed:       9317
--------------------
Number of people in queue:
Slow queue:     0
Fast queue:     0
--------------------
Number of people in the building:
In south:       0
In west:        0
In north:       0
In east:        44
In exit:        0
--------------------
Average number of people in each location:
South mean:     167.376
West mean:      163.012
North mean:     167.376
East mean:      163.012
```

Fig. 7. Output from sample run

The capacity of the elevators is set to 40 people, and the deck capacity is set to 1000 people. The firing time of tCloseXDoor is set to 25 seconds to simulate that it takes 50 second going up (see VI-C). The doors automatically close after 30 seconds and then proceeds to the next floor. Logging takes place every 10 minutes. These parameters denote the default parameters. For the sample run, we present the output and three plots that describe the flow of visitors.

Figure 7 shows the output from running the simulation with default parameters. The output shows that there were a total of 9361 visitors, of which 9317 has departed. This is slightly higher than the expected 8750 average visitors per day. Additionally, only 44 visitors were remaining in the tower at the end of the simulation. Evidently, our implementation is able to simulate a day at Tokyo Skytree using real parameters.

Monitoring where the visitors are located is vital to discover where flow congestion occurs. Figure 8 does not indicate any signs of flow congestion. One can detect congestion if one location constantly has much more visitors than in the next location(s). Each location in the deck has, on average 160-170 visitors each hour. If these numbers were vastly different, then it would likely be because of congestion.

The length of the queues indicates the efficiency of the elevators. Figure 9 shows the size of each queue throughout the day. The blue line displays how many visitors entered the fast lane, and the orange line shows how many entered the slow lane. The fast lane is constantly low because they are prioritized.

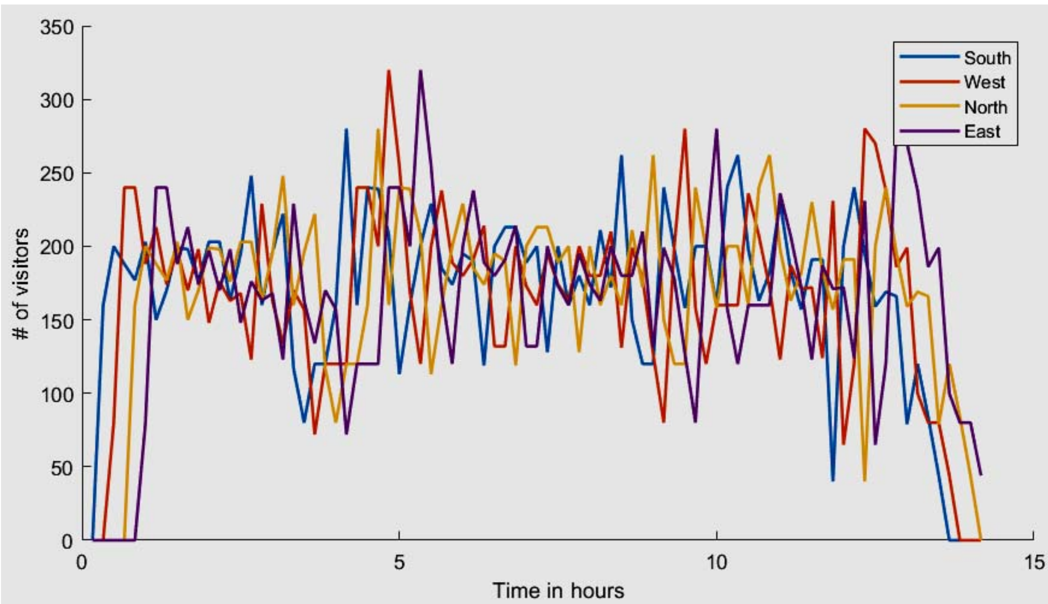

Fig. 8. Number of visitors at each location in the deck

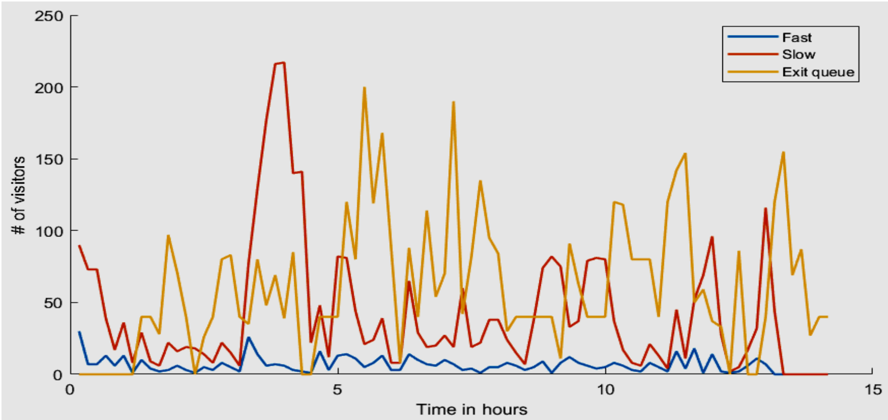

Fig. 9. Number of visitors in each queue

Visitors in the slow lane only enter an elevator if there is enough space available after selecting visitors from the fast lane. This is why the length of the slow lane fluctuates significantly more than the fast lane. The spike from hour 3 to hour 4 likely happened because the deck capacity was reached, not allowing visitors in the fast and slow lane to enter the elevator. The yellow line shows the length of the exit queue at the deck. The exit queue fluctuates based on the flow at the deck and arrival times. At hour 5, the elevators have nearly transported every visitor that stood in the fast and slow lines up to the deck. It takes approximately one hour to go through the deck. This is likely why the exit queue spikes one hour after the spike in the slow queue.

Figure 10 shows, with the blue line, how many visitors are up in the deck and the exit queue. The grey line marks the deck capacity at 1000 visitors. The number of visitors typically hovers around 800, but it never exceeds the capacity.

C. Varying the Elevator Capacity

In our first experiment, we investigate how elevator capacity affects the flow of visitors. Three simulations were conducted. The simulations used an elevator capacity of 20, 40 and 60, respectively. All other parameters were set to their default values. It is clear that an elevator capacity at 20 is not able to keep up with the queues. There were noticeably fewer visitors at the deck on average. With an elevator capacity at 40 (same as sample run), the deck capacity could be reached, and the queues did not continuously grow. The deck capacity was not reached when the elevator capacity was increased to 60 visitors. Additionally, the queues were generally shorter. The elevators are efficient, and there is hardly any queue at the bottom. This indicates excellent flow. People get to the elevator quickly, spends an hour in the deck and leaves swiftly. This scenario would likely be able to support more daily visitors.

VIII. DISCUSSION, CONCLUSION ND FUTURE WORK

From the results in section VII-B, it is evident that our model makes a solid attempt at simulating a day in Tokyo Skytree. The arrival module generates the average daily visitor count at Tokyo Skytree. Most of the visitors depart the building by the end of the day. This shows that our model was able to simulate the scenario at the tourist attraction.

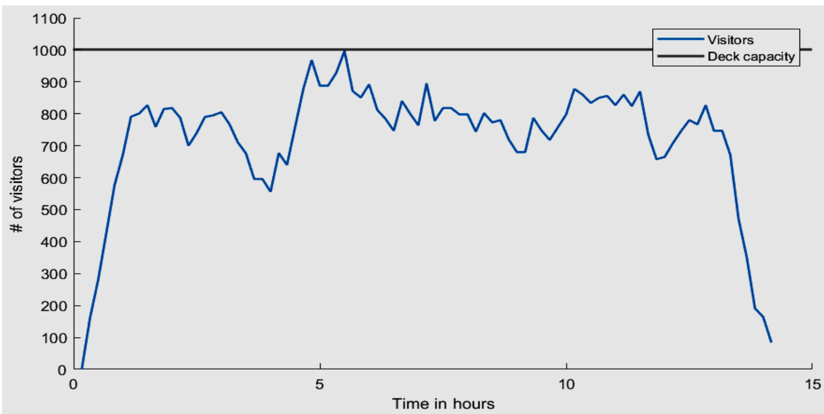

Fig. 10. Total number of visitors at the deck

By changing the simulation parameters, we can explore the importance of the elevator specifications. The results in section VII-C show that elevators with half the capacity would not be able to handle the average amount of traffic. This proves that the decision to install Japans fastest large-capacity elevators was essential to the tourist attraction's success.

Limitations of the Model: Currently, the model assumes that the visitors are individuals and arrive one by one. This is not very realistic as visitors can also arrive in groups and thus enter and leave the elevators and enjoy the venue in groups. Thus, the model needs to be extended to cope with the group arrivals. Also, as clearly stated under the assumptions, every fourth visitor is considered a foreigner and uses the fast lane. However, during the Olympics event, this assumption can become wrong as most visitors could be foreigners.

Further work: First of all, the Petri net modules presented in this paper do not strictly adhere to the modular Petri net definitions given in [16]. Hence, some enhancements are to be done to the modules. Secondly, the model does not include the details of the cafe/restaurant and the souvenir shop. Without changing the existing model, the additional details (such as cafe and restaurant) can be wrapped into separate modules; and these newer modules can then be joined with the existing model. Surely, expanding a model by incorporating additional modules with a minimal change to the existing model is the main benefit of the modular Petri Net.

REFERENCES

[1] Council on Tall Building and Urban Habitat, "100 tallest freestanding towers by height to architectural top," https://www.skyscrapercenter.com/buildings?list=tallest-towers, Tech. Rep., 2020, accessed on 10 Feb 2021.

[2] Themed Entertainment Association (TEA), "Theme index and museum index: The global attractions attendance report," https://www.teaconnect.org/images/files/TEA328381804/190528.pdf , Tech. Rep., 2020, accessed on 10 Feb 2021.

[3] Tokyo Skytree, "Tokyo skytree traditional techniques and forefront technologies from japan," http://www.tokyoskytree.jp/others/pdf/skytree english.pdf?201904, Tech. Rep., 2020, accessed on 10 Feb 2021.

[4] S. Kawamura, "Tokyo skytree welcomes surge of inbound tourists 7 years after opening," https://mainichi.jp/english/articles/20190522/p2a/00m/0na/009000c, Tech. Rep., 2020, accessed on 10 Feb 2021.

[5] ——, "Tokyo skytree," https://www.toshibaelevator.co.jp/elv/infoeng/projects/japan/01.html, Tech. Rep., 2020, accessed on 10 Feb 2021.

[6] I. D. Touretzky, M. Mozer, M. Hasselmo et al., "Improving elevator performance using reinforcement learning," in Advances in neural information processing systems. Citeseer, 1996, vol. 8, pp. 1017–1023.

[7] R. H. Crites and A. G. Barto, "Elevator group control using multiple reinforcement learning agents," Machine learning, vol. 33, no. 2, pp. 235–262, 1998.

[8] Y.-W. Ho and L.-C. Fu, "Dynamic scheduling approach to group control of elevator systems with learning ability," in Proceedings 2000 ICRA. Millennium Conference. IEEE International Conference on Robotics and Automation. Symposia Proceedings (Cat. No. 00CH37065), vol. 3. IEEE, 2000, pp. 2410–2415.

[9] F. Ahmad, I. Fakhir, S. A. Khan, and Y. D. Khan, "Petri net-based modeling and control of the multi-elevator systems," Neural Computing and Applications, vol. 24, no. 7, pp. 1601–1612, 2014.

[10] C.-H. Lin and L.-C. Fu, "Petri net based dynamic scheduling of an elevator system," in Proceedings of IEEE International Conference on Robotics and Automation, vol. 1. IEEE, 1996, pp. 192–199.

[11] Y. C. Cho, Z. Gagov, and W. H. Kwon, "Timed petri net based approach for elevator group controls," in Proceedings 1999 IEEE/RSJ International Conference on Intelligent Robots and Systems. Human and Environment Friendly Robots with High Intelligence and Emotional Quotients (Cat. No. 99CH36289), vol. 2. IEEE, 1999, pp. 1265–1270.

[12] R. Davidrajuh, "Developing a toolbox for modeling and simulation of elevators," International Journal of Simulation–Systems, Science & Technology, vol. 20, no. S1, pp. 1.1–1.6, 2019.

[13] J. L. Peterson, Petri net theory and the modeling of systems. Prentice Hall PTR, 1981.

[14] R. Davidrajuh, Modeling Discrete-Event Systems with GPenSIM. Cham: Springer International Publishing, 2018.

[15] GPenSIM, "General-purpose Petri net simulator," http://www.davidrajuh.net/gpensim, Tech. Rep., 2019, accessed on 20 July 2020.

[16] R. Davidrajuh, "Extracting petri modules from large and legacy petri net models," IEEE Access, vol. 8, pp. 156 539–156 556, 2020.

[17] ——, "A new modular Petri net for modeling large discrete-event systems: A proposal based on the literature study," Computers, vol. 8, no. 4, p. 83, 2019.

[18] Code, "Complete code for skytree simulation," http://www.davidrajuh.net/gpensim/Pub/2021/Skytree.pdf, Tech. Rep., 2020, accessed on 10 Feb 2021.

[19] TEA and AECOM, "The world's premier infrastructures," http://aecom.com/, Tech. Rep., 2020, accessed on 10 Feb 2021.