*CSE 561: Modeling & Simulation Theory and Application*

**Amusement Park optimal ratio of fastpass buyers**

Jisoo Lee Ying Yu Wu

Computer Science PhD Student Computer Science Masters Student

School of Computing and Augmented Intelligence

Arizona State University, Tempe, AZ, USA, 85281

Due Date: 11-03-2023

**Contents**

1. Introduction

1.1 Objectives

1.2 Background

1. System Description

2.1 Configuration and Variables

1. Modeling Approach

3.1 Visitor

3.2 Generator

3.3 Attraction

3.4 Transducer

3.5 Monitor

1. Experimental Setup
2. Results
3. References

**Team Members Tasks (% effort is preliminary)**

| **List of tasks** | **Name1** | **% Effort** | **Name 2** | **% Effort** |
| --- | --- | --- | --- | --- |
| 1. Introduction and background | Jisoo | 80 | Ying Yu | 20 |
| 2. System description | Jisoo | 20 | Ying Yu | 80 |
| 3. Modeling approach | Jisoo | 50 | Ying Yu | 50 |
| 3.1 Visitor | Jisoo | 50 | Ying Yu | 50 |
| 3.2 Generator | Jisoo | 50 | Ying Yu | 50 |
| 3.3 Attraction | Jisoo | 50 | Ying Yu | 50 |
| 3.4 Transducer | Jisoo | 50 | Ying Yu | 50 |
| 3.5 Monitor | Jisoo | 50 | Ying Yu | 50 |
| 4. Experimental setup | Jisoo | 80 | Ying Yu | 20 |
| 5. Results | Jisoo | 80 | Ying Yu | 20 |
| 6.References | Jisoo | 50 | Ying Yu | 50 |

Abstract

This project aims to model the validity of FastPass tickets in an amusement park using the DEVS-Suite Simulator. The primary objective is to uncover the relationship between the sales rate of these FastPass tickets and the waiting times of buyers. Various variables, such as the number of rides, their durations, visitor flow, and the popularity of rides, will influence this model simulation. This will create disparities in waiting times between buyers of the two types of tickets. Ultimately, this model will provide valuable insights for amusement park management, allowing them to adjust ticket quantities or prices in order to regulate the sales rate effectively.

# 1 Introduction

# 1.1 Objective

The objective of this project is to design and simulate a model that elucidates the relationship between the sales rate of FastPass tickets, commonly introduced in places like amusement parks, and the reduction in waiting times.

## 1.2 Background

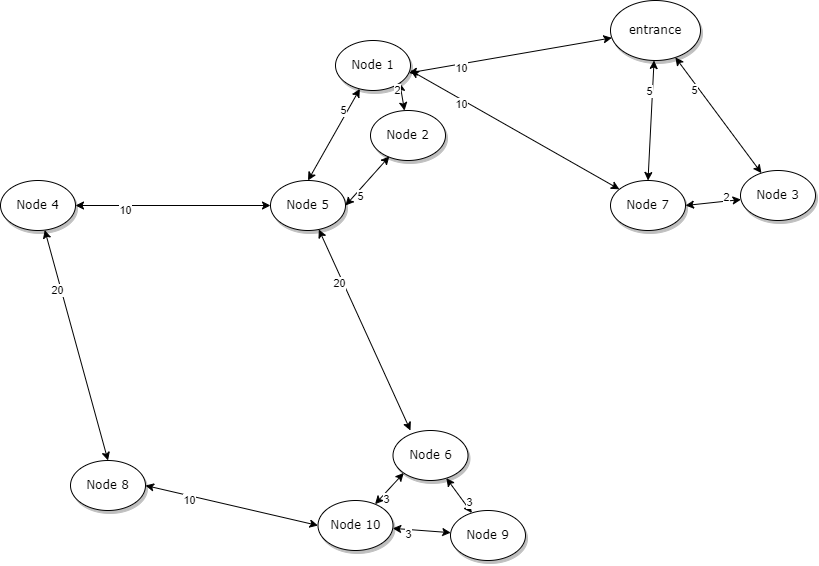
The FastPass system was created by the Walt Disney Company to enhance customer access to attractions in Disney theme parks. After the onset of COVID-19, in 2021, Disney discontinued this system in Disneyland. However, they continue to operate a similar system under a different name, “Genie[1]”, allowing customers who pay an additional fee to access attractions quickly. Another popular theme park, Universal Studios, also sells tickets with similar functionality under the name "Express Pass tickets[2]". Many amusement parks, in a similar fashion, use systems where customers who pay extra for FastPass tickets can bypass regular queues and gain expedited access to attractions.

The way these FastPass tickets work is by allowing the buyers to enter a separate queue when using the amusement rides. Therefore, the significance of purchasing these tickets is that only a small proportion of the visitors to the amusement park purchase them. If all visitors to the amusement park were to purchase these tickets, unless there are no tickets with higher priority, all visitors would have to wait as long as regular visitors do.

# 2 System Description

We will assume that FastPass is effective when the average waiting time for regular ticket buyers is twice as long as the average waiting time for FastPass ticket buyers. We will perform several experiments with different ratios and variable ranges to see how the waiting time changes depending on the ratio. Consequently, through simulation, we can determine to what extent we adjust the ratio of FastPass ticket buyers in order to achieve a targeted gap in waiting times. Considering the complexity of amusement park flow tracking, we make some assumptions to limit the problem statement.

Figure 1 illustrates the amusement park map and passenger flow. In this map, each Node represents an attraction, and the distances between two attractions are denoted as time intervals, measured in minutes. Visitors will enter the amusement park via the entrance since the park's opening time and make their attraction selections based on a formula that takes into account both the priority order list and the distance. Visitors will be assigned to random order of attractions, and they will remain in their chosen line once they choose it. There are only two lines in every attraction: general line and FastPass line. There is no higher priority line for elderies or disable people. The only way visitors can access is via the designated paths on the map, and they will always opt for the shortest path between their location and their chosen attraction. Visitors will leave the park immediately either upon experiencing all the attractions in their queue or at the park closing time. All FastPass buyers can ride the attraction first compare to normal visitors.



**Figure 1: amusement park map**

## 

## 2.1 configuration & variables

There are variables that change depending on the ratio of FastPass ticket buyers.

| Variables | definition |
| --- | --- |
| Number of amusement park attractions | 10 |
| Number of visitors | 2000-3000 (50-70 per group) |
| Park opening time | 8:00 am |
| Park last entrance time | 2:59 pm |
| Entrance time for each tourist | People group will enter the park at 10-minute intervals. |
| Distance (time) between attractions (minutes) | We assume that people walk at a certain rate and it takes a specific time between two attractions. The time between attractions will be set to different constant numbers mentioned in Figure 1. |
| Priority order list | Each tourist will have a priority order list for attractions. Here is an example.   | priority order | Node | | --- | --- | | 1st | 5 | | 2nd | 6 | | 3rd | 5 | | 4th | 3 | | 5th | 7 | | 6th | -1 | | 7th | 9 | | 8th | 4 | | 9th | 9 | | 10th | 8 |   The range of the number of attractions in the queue is between 0 and 10. The value of -1 in the queue is used to represent the scenario where a visitor will ride fewer than 10 attractions. When a node is assigned the value -1, the visitor proceeds to the next ride in the queue.This priority list will be set randomly. Visitors can revisit attractions. |
| Duration for each attraction | Each amusement ride operates for a specific duration. The machines operate with a high degree of precision, and with each attempt, they run for the exact same duration of time.   | Node | Duration (minutes) | | --- | --- | | 1 | 10 | | 2 | 5 | | 3 | 3 | | 4 | 4 | | 5 | 5 | | 6 | 5 | | 7 | 15 | | 8 | 8 | | 9 | 4 | | 10 | 6 | |
| Capacity of each attraction | Each attraction has their own capacity.   | Node | Capacity | | --- | --- | | 1 | 10 | | 2 | 30 | | 3 | 15 | | 4 | 40 | | 5 | 50 | | 6 | 50 | | 7 | 20 | | 8 | 30 | | 9 | 30 | | 10 | 50 | |
| Ratio of FastPast Buyer  (Input) | [10,20,30] (%) |
| Waiting time for each attraction  (Ouput) | Waiting time for each attraction will be different depending on the ratio of FastPass ticket buyers   | Node | FastPass waiting time (minutes) | normal waiting time (minutes) | | --- | --- | --- | | 1 | F1 | N1 | | 2 | F2 | N2 | | 3 | F3 | N3 | | 4 | F4 | N4 | | 5 | F5 | N5 | | 6 | F6 | N6 | | 7 | F7 | N7 | | 8 | F8 | N8 | | 9 | F9 | N9 | | 10 | F10 | N10 | |

**Table 1: variables and their definitions**

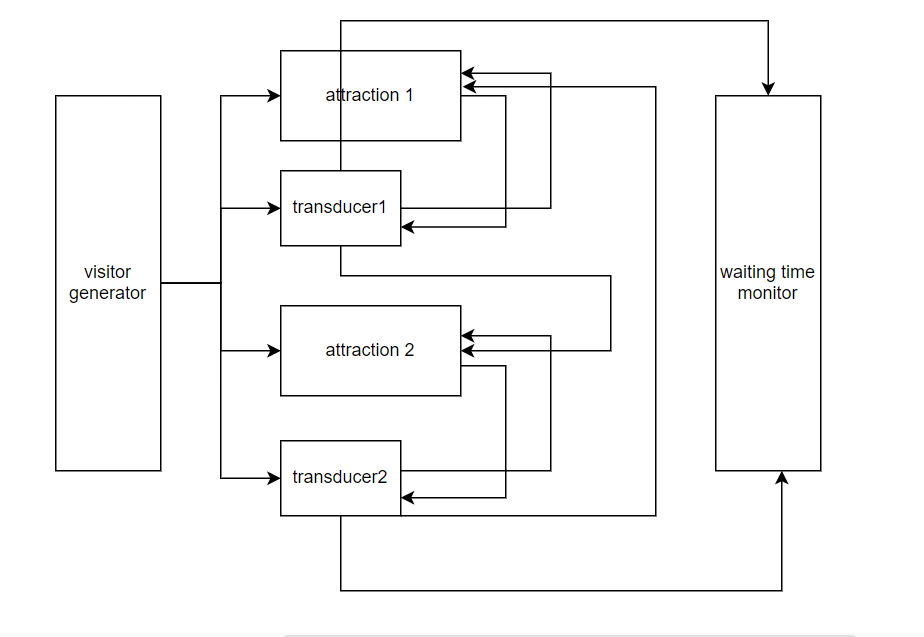
# 3 Modeling Approach

In this project, our goal is to implement a coupled model system where each component of the amusement park serves as an atomic component, and visitors become a job entity. These components are interconnected, creating an integrated model. The objective is to simulate the entire system, all with the aim of achieving our simulation goals.

This model consists of four parts. First, there's the generator, which generates visitors. Then, there are multiple processors that manage the various activities occurring at the attraction. Additionally, there's a transducer that tracks the time-related events generated by these processors. Finally, there's a monitor that processes the generated output to present it in the desired format.

Since visitors have their prioritized lists of rides to visit, our model is designed so that each visitor uses one ride and then proceeds to the next ride in their list. In reality, visitors move between rides, but in the simulation model, all visitors generated by the generator are initially sent to all rides. They then complete their tasks by comparing their priorities with the respective ride, and this information is shared with other rides through associated transducers.

Once the most recently generated visitors have completed all the rides on their priority lists at each processor, each transducer eventually sends the total waiting time and the number of visitors of the two types to the monitor. The monitor receives this information, calculates the average values, and then displays how significant the difference is between the two waiting times.

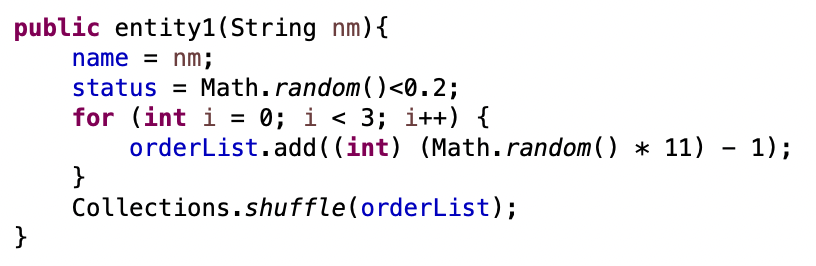


**Figure2 : Amusement park system abstraction**

In the following subsections, each subsection describes and specifies the model, including the implementations that have been completed.

## 3.1 Visitor

Generator generates entities called “visitor”. Each visitor has their name, ticket type and a list of priority. Constructor for creating instances defines what they will have. They gives each instance String type of name, boolean type of status and a list that has random integer values from -1 to 9. By changing the probability when they assign status value, we can adjust the ratio of Fastpass ticket buyers.



**Figure3 : Visitor Entity**

## 3.2 Generator

The generator will generate a group of people every 10 minutes. Each group has the number of people specified in the configuration within the given range. When they generate people, they use for loop. After appending each entity to the message, they return all at the same time. By using the variable “index”, we make them enable to count the number of entire visitors.



**Figure4 : Outputs of Generator**

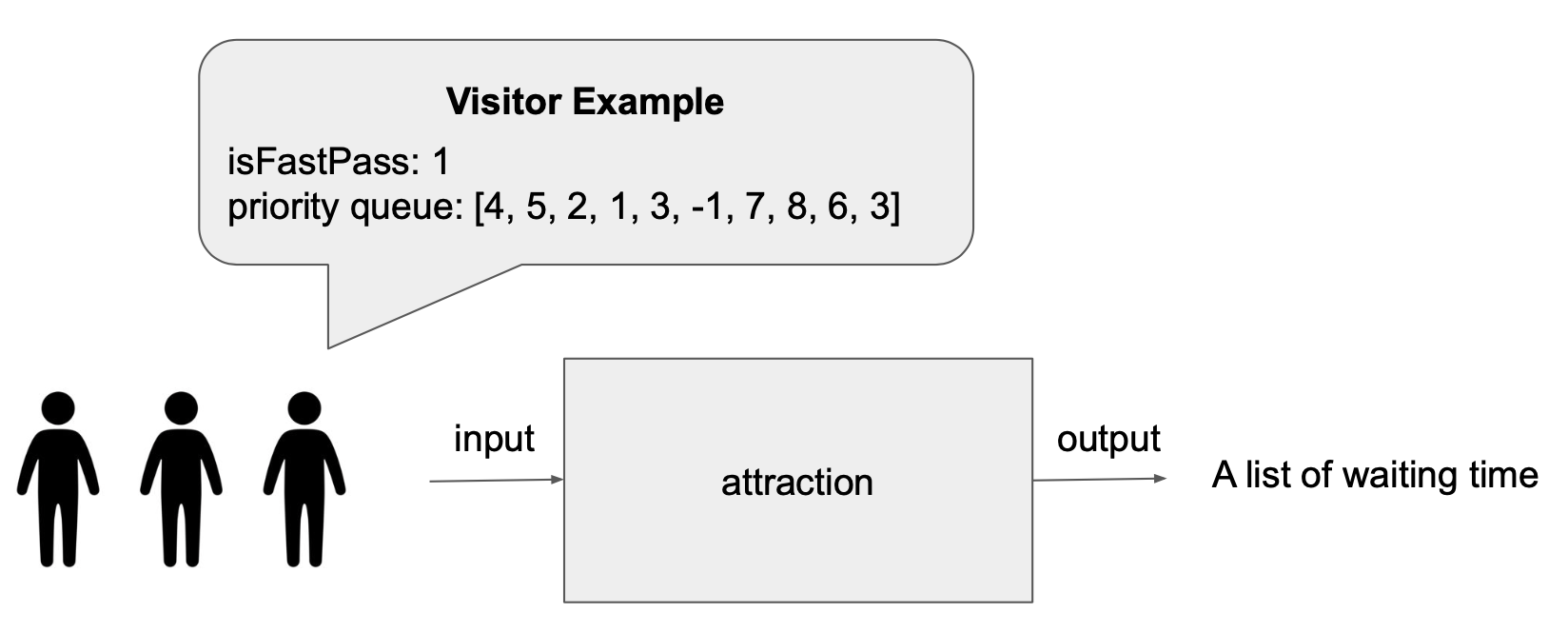
## 

## 

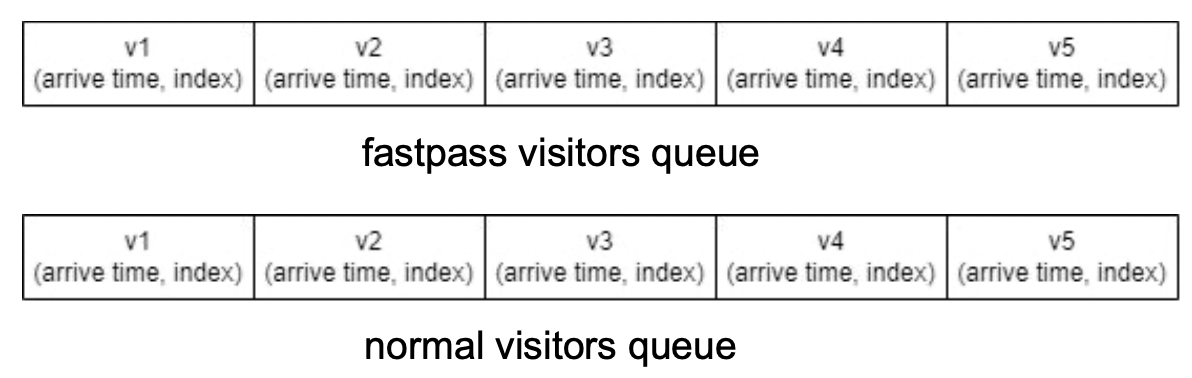
## 3.3 Attraction

In the project, each ride at the amusement park is implemented as a processor component in the model. The system has as many processors as there are rides, and each processor interacts with generators and transducers, by exchanging their inputs and outputs.

The operation works as follows: When a visitor entity is created and sent to a processor, they are initially categorized into different groups based on their first-priority ride. Then, if the group corresponds to the processor, which is the ride they are assigned to, they are placed into two different queues within the processor according to a predefined algorithm. Figure 6 shows how queues work as two waiting lines. The initial state of the processor is "Loading." Subsequently, as the count reaches the capacity of the ride, the entities are removed from the queues, and the processor's status is changed to "Run." The status changes back to "Loading" after the duration of the ride, at which point the Transducer immediately conveys information about the visitors who used the previous ride. The Transducer passes this information to other processors, and this process results in the removal of elements from the visitors' priority lists. In the end, when all visitors' priority lists become null, the final waiting time is calculated.



**Figure5 : Attraction processor component diagram**

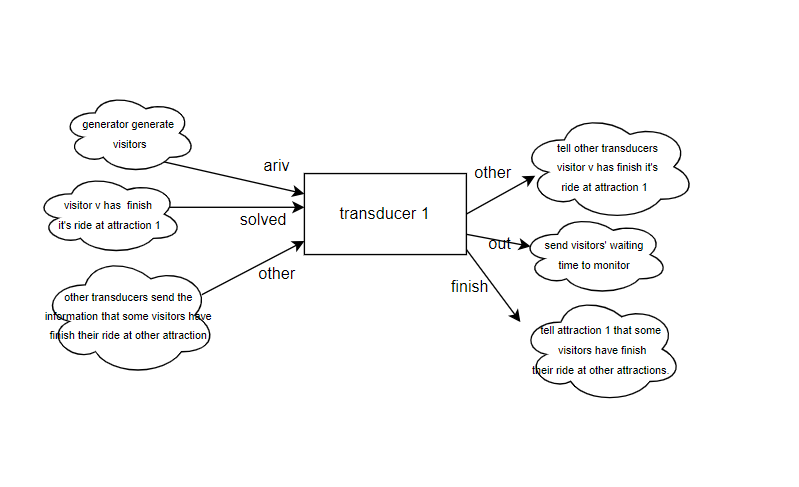


**Figure6 : Queues in Processor**

## 

## 3.4 Transducer

* Every time the generator generates output, every transducer will receive visitors from the generator through the “ariv” port.
* When a visitor arrives at an attraction managed by the transducer during their visit, the transducer records the arrival time of the visitor. When the visitor finishes a ride at this attraction, the attraction will send its outputs to the transducer through the “other” port. At this point, the transducer receives the output from the attraction through its “solved” port, thereby generating waiting time.
* After the transducer receives the information that some visitors have finished their ride at this attraction through the “other” port, it will send this information to the rest of the attractions through the “finish” port.
* In the end, it will send waiting time for every visitor to monitor through the “out” port.



**Figure7: Transducer**

## 3.5 Monitor

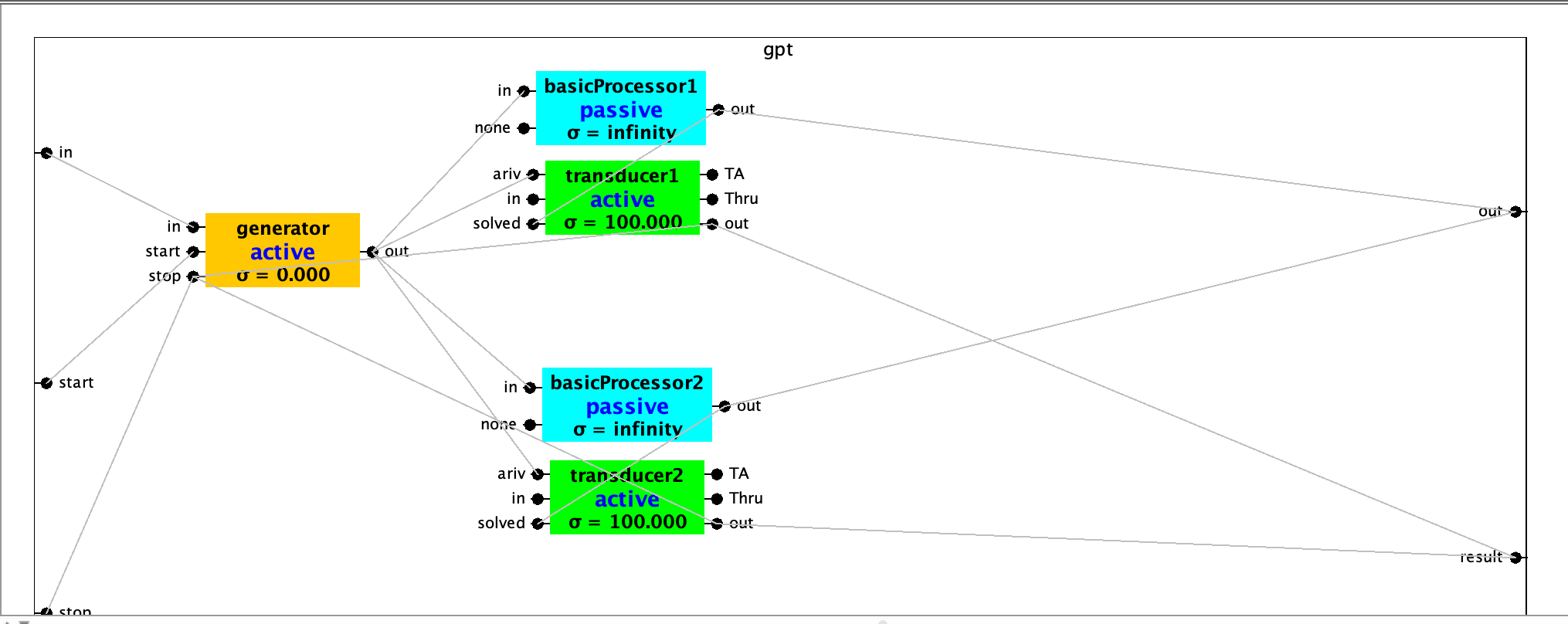
Monitor will receive waiting time from every transducer. it will calculate the final average waiting time for both FastPass visitors and normal visitors. Ultimately, it provides information to determine the fastpass ticket sales ratio by analyzing the difference in waiting times between the two types of visitors.

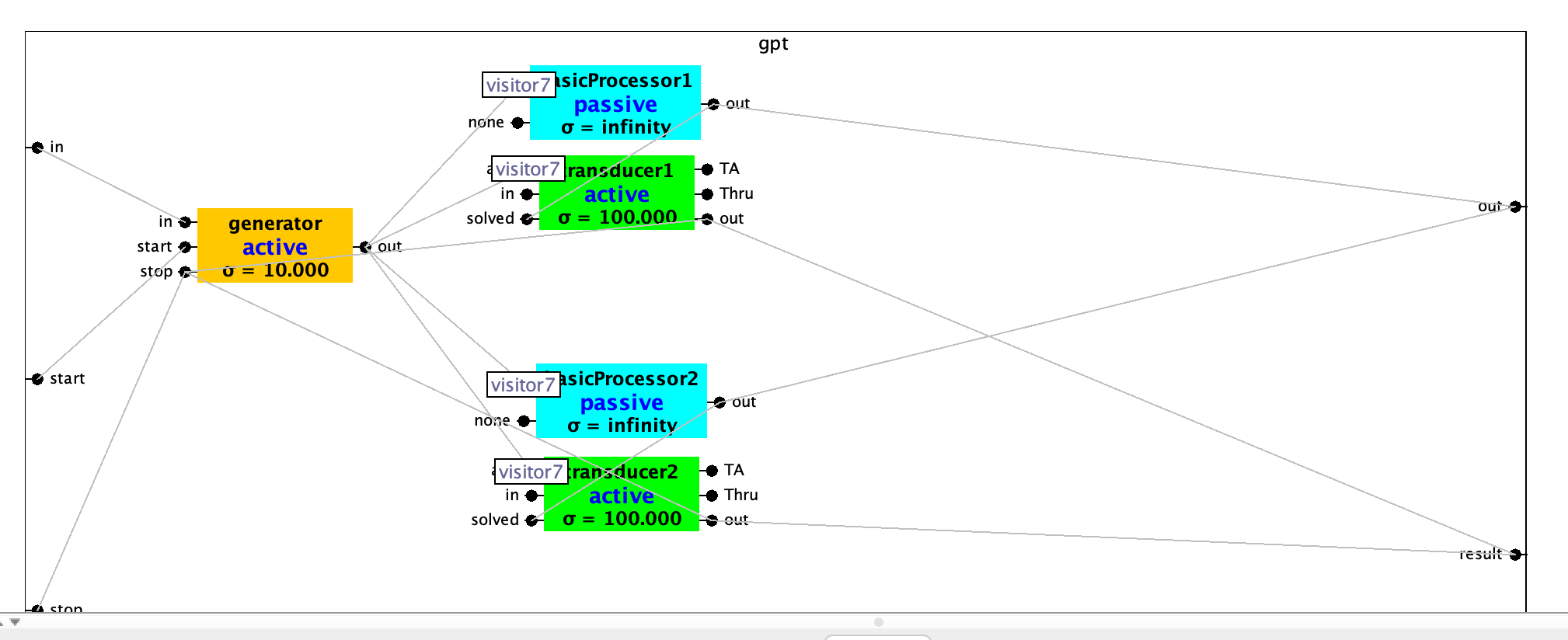
# 

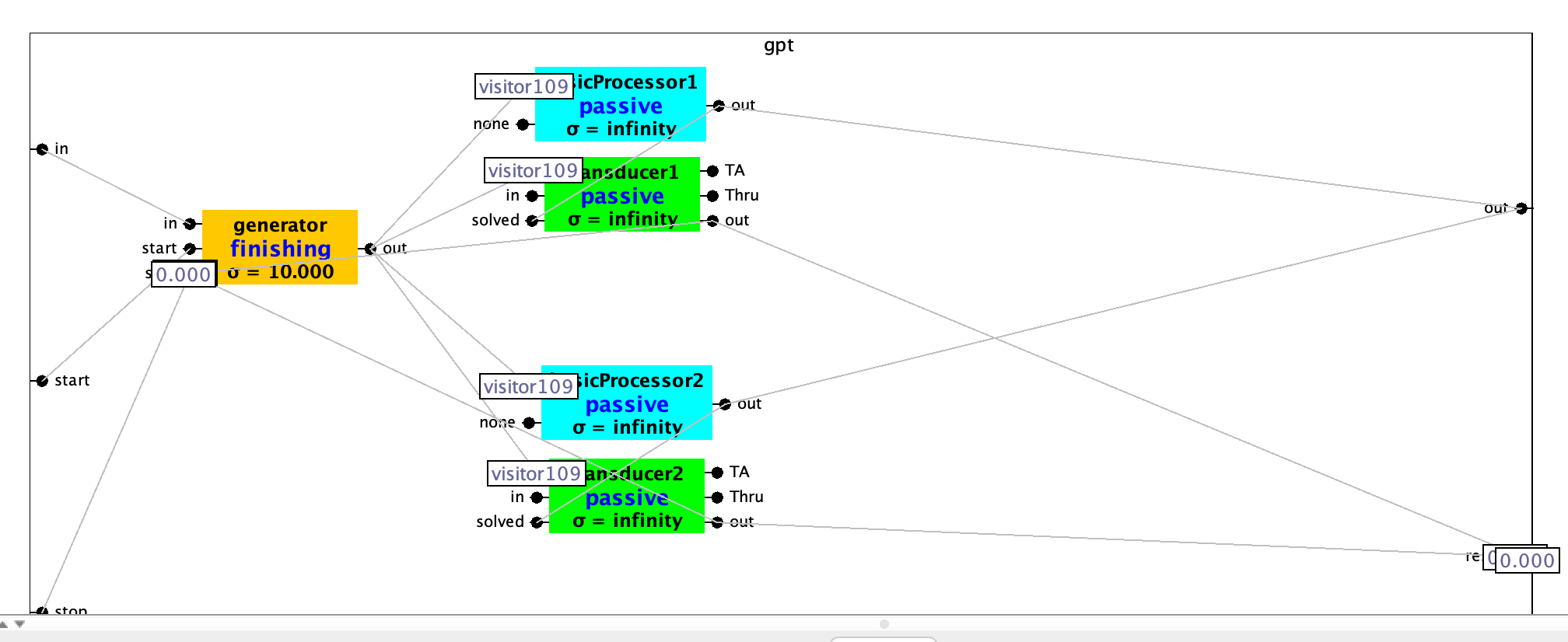
# 4 Experimental Setup

The simulation experiments will be conducted three times for each of the three different fastpass ticket purchase ratios. Other variables are set as defined in the configuration. Each time, as the number of visitors is randomly determined, and as their priorities change, running multiple simulations for a single configuration will help obtain valid results. The fastpass ratios are set to 10%, 20%, and 30%, as specified in the configuration above. By observing the resulting differences in the final average waiting times, it will aid in determining the "optimal fastpass ticket sales ratio" targeted in this project.

The diagram below provides a simplified overview of how our model operates in the DEVS simulator. Visitors generated by the generator are passed to each processor and transducer, with the final step being the delivery of the Transducer's output to the ultimate Results. More processors and transducers will be added for the complete implementation of the simulation.







**Figure 8: DEVS simulator view**

# 5 Results

Currently, the model has been implemented with a minimal set of atomic components, and the implementation of components is in prograss except for the generator. The experiment's results will be released once all components are added and the model implementation is completed.

# 

# 6 References

[1]<https://disneylandtourguide.com/disney-genie-the-new-replacement-for-fastpass-at-disneyland-and-walt-disney-world/>

[2]https://www.universalorlando.com/web/en/us/tickets-packages/express-passes