AMC Roosevelt Field 8 FlexSim Model Report

BY: Youssef Elmougy & Sari AlTawabini

Model Assumptions:

System Start Time - 8:50am (1 hour before first movie starts)

First customer groups arrival time – 9:20am

First movie start time – 9:50am

Last movie end time – 1:30am

System Stop Time – 2:30am (1 hour after last movie finishes)

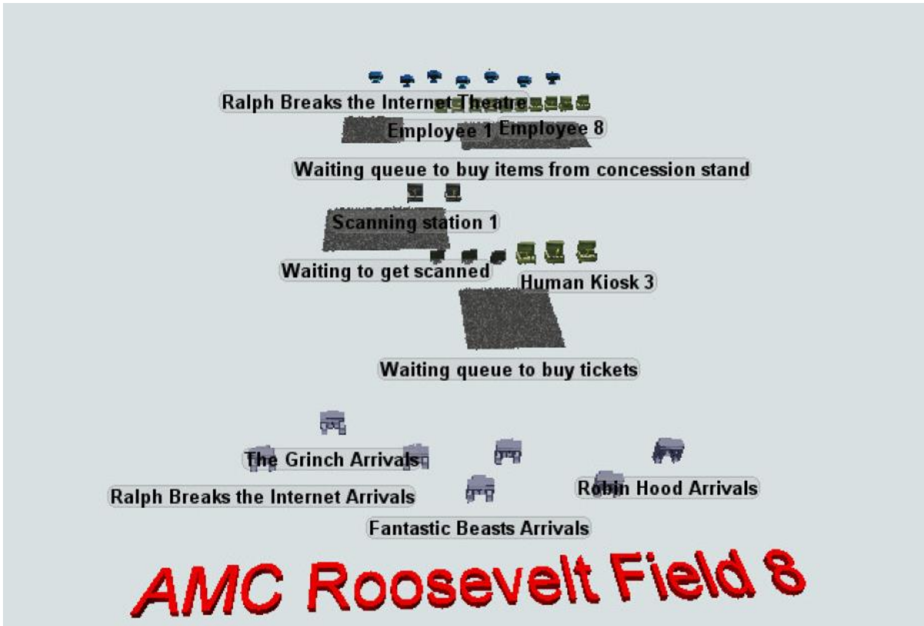
Model Settings:

Time Units - Minutes

Movies	**Genre**
Ralph Breaks the Internet	Family
The Grinch	Family
Creed II	Drama
Fantastic Beasts	Drama
Instant Family	Drama
Robin Hood	Action
Widows	Drama

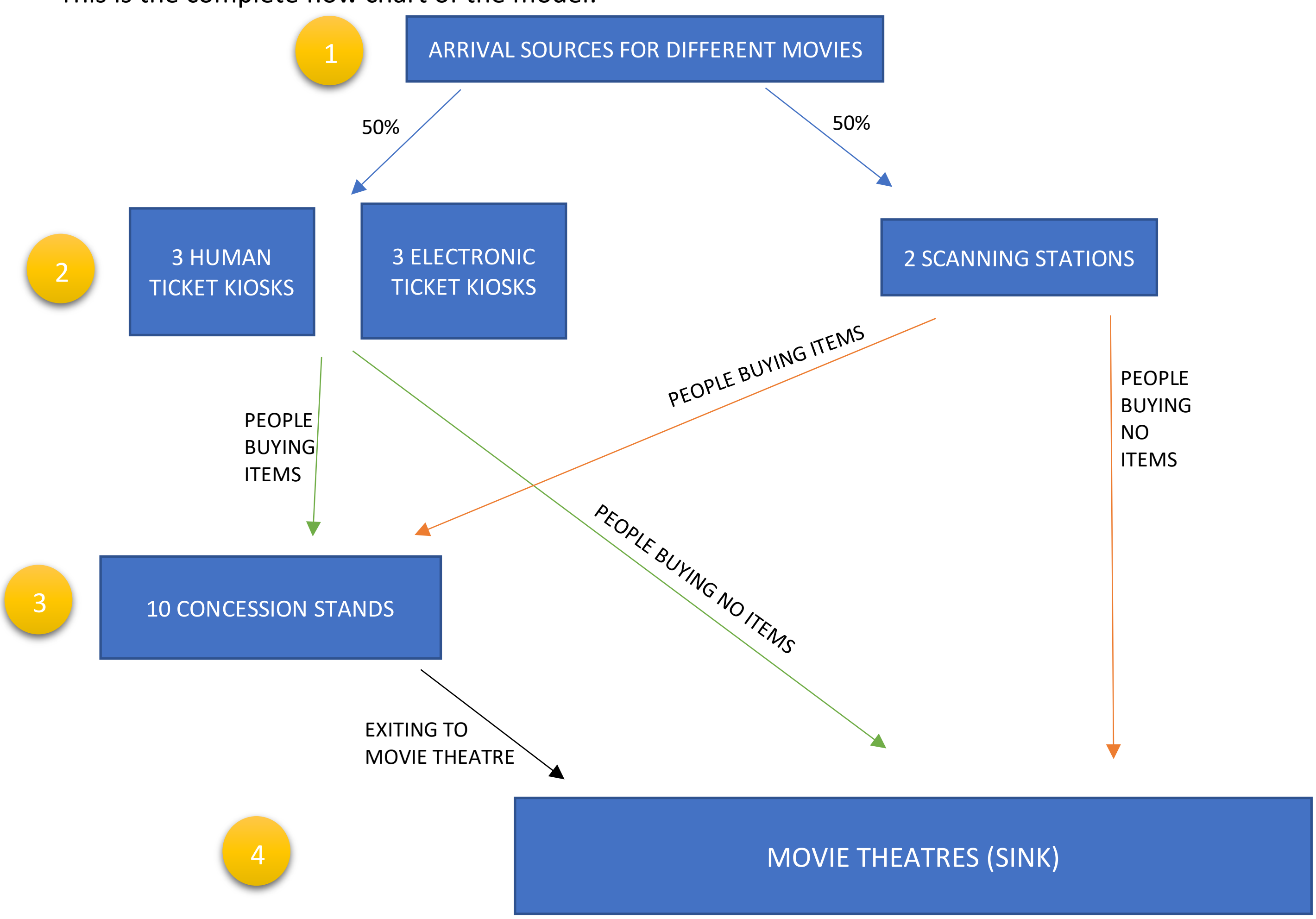
All numbers are approximated due to experimental purposes.

Our project is a model of the AMC Roosevelt Field 8 cinema. All the values calculated and utilized in this model are accurately approximated to mimic the real-life scenario. This report explains the realistic model in simplistic steps.



Picture of model

This is the complete flow chart of the model:



Breaking down the flow chart into steps:

1. Arrivals – The demand of customers arriving at the movie theatre depended on the movie, the rotten tomatoes score, and the time of the day score. We split every movie into its individual times and calculated the demand of customers for each movie time. The rotten tomatoes scores used are as follows:

MOVIE	ROTTEN TOMATOES SCORE
Ralph Breaks the Internet	88%
The Grinch	58%
Creed II	83%
Fantastic Beasts	39%
Instant Family	82%
Robin Hood	16%
Widows	91%

Having the rotten tomatoes score and the time of day score, we took each movie and calculated the demand for each movie show timing. After that, we took each individual show timing and calculated the amount of people arriving at intervals before and after the movie (30 mins before, 15 mins before, on time, 15 mins late) by multiplying it with percentages provided for the specific genres.

We will explain using the movie Ralph Breaks the Internet as an example (all other movies used the same calculations and logic).

Ralph Breaks the Internet		
Demand		
		Rounded
9:50am	105.6	106
12:55pm	79.2	80
3:05pm	79.2	80
3:50pm	79.2	80
6:00pm	105.6	106
9:00pm	118.8	119

= number of seats * rotten tomatoes score * time of day score
= 150 * 0.88 * 0.8

9:50am movie		
Arrival times		
		Rounded
9:20am	31.8	32
9:35am	42.4	42
9:50am	10.6	11
10:05am	21.2	21

= demand * percentage
= 106 * 0.3

12:55pm movie	
Arrival times	
12:25pm	24
12:40pm	32
12:55pm	8
1:10pm	16

= demand * percentage
= 80 * 0.3

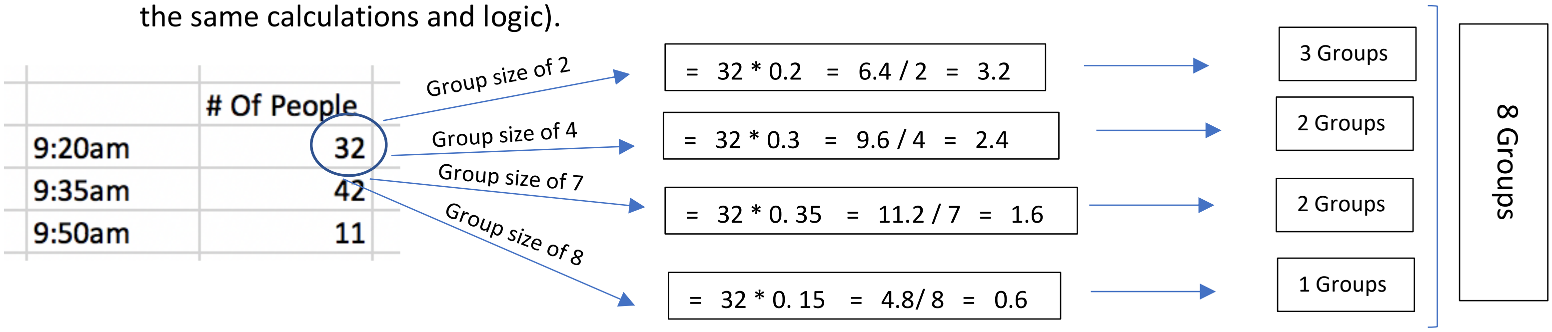
Finally, we accumulated all the arrivals in one table

	# Of People
9:20am	32
9:35am	42
9:50am	11
10:05am	21
12:25pm	24
12:40pm	32
12:55pm	8
1:10pm	16
2:35pm	24
2:50pm	32

3:05pm	8
3:20pm	40
3:35pm	32
3:50pm	8
4:05pm	16
5:30pm	32
5:45pm	42
6:00pm	11
6:15pm	21
8:30pm	36
8:45pm	47
9:00pm	12
9:15pm	24

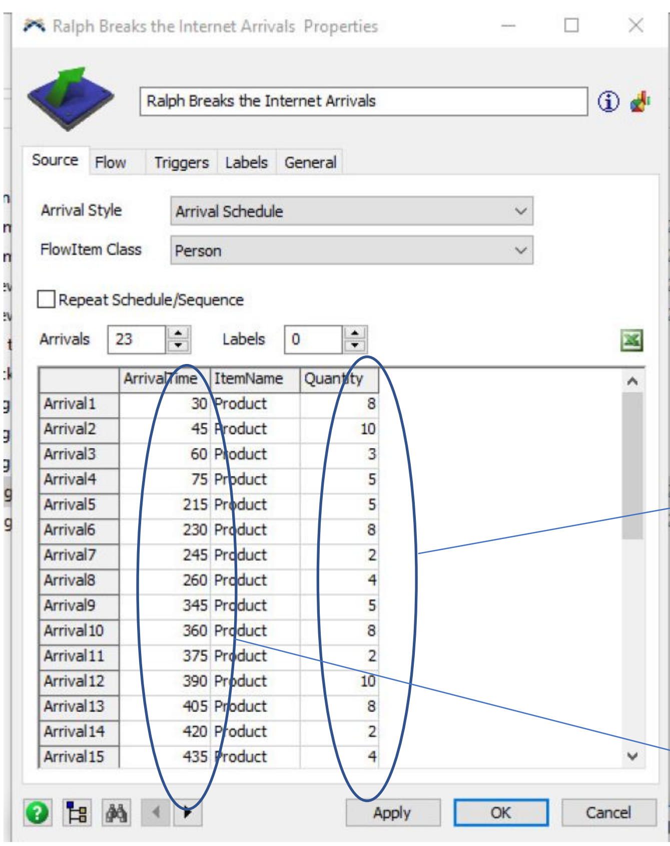
To incorporate the idea of customers arriving in groups, we decided to calculate the number of groups beforehand since we already know the number of customers coming into the movie theatre at a given time. The number of customer groups was calculated by taking the number of customers arriving at the specific times, then multiplying it by the percentages of the different group sizes deferring by genres (2,4,7,8), once we generate the values we divide each value by the specific group size and add up the number of full groups created.

We will explain using the movie Ralph Breaks the Internet as an example (all other movies used the same calculations and logic).



We calculated the group sizes for each number for customers arriving and therefore converting our table containing number of people arriving at specific times into a table containing number of groups arriving at the different times. The following is the amended table with number of groups and a diagram of its implementation in the source for the movie in our model:

(MODEL SETTINGS: Start Time: 8:50am, Time Units: Minutes)



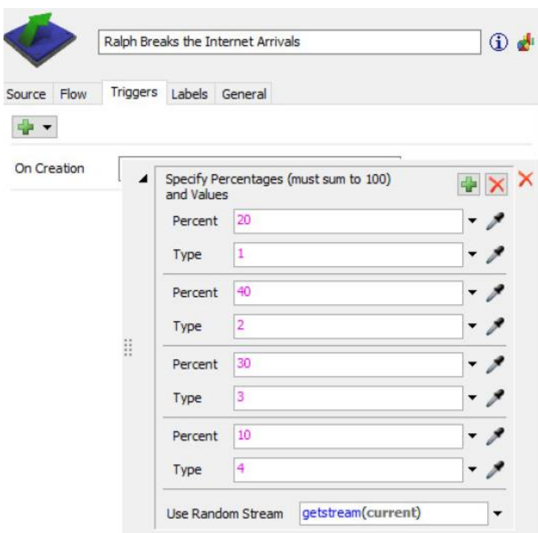
	# Of Groups
9:20am	8
9:35am	10
9:50am	3
10:05am	5
12:25pm	5
12:40pm	8
12:55pm	2
1:10pm	4
2:35pm	5
2:50pm	8
3:05pm	2
3:20pm	10
3:35pm	8
3:50pm	2
4:05pm	4
5:30pm	8
5:45pm	10
6:00pm	3
6:15pm	5
8:30pm	10
8:45pm	12
9:00pm	3
9:15pm	5

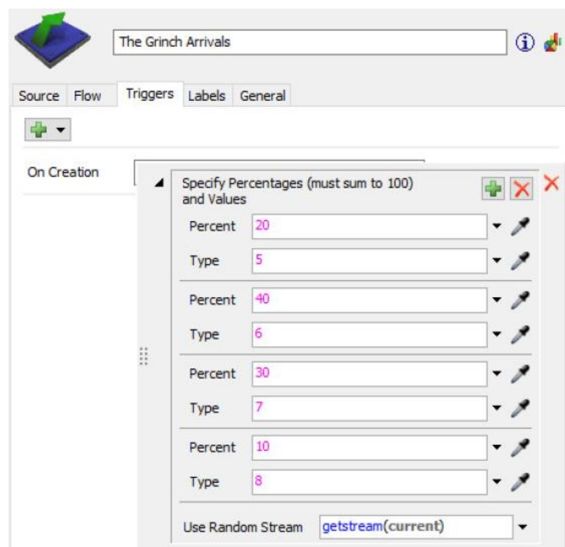
This is the arrival time of the customer groups with reference to 8:50. The first value is the arrival for the groups coming in 30 mins before the first movie timing.

Lastly, at arrivals we created a trigger of item type. This item type is the differentiator of the percentages of groups that buy items at concessions (0,2,4,5). We created a different item type by percentage (depending on genres) for the number of items bought for each specific movie.

We will explain using the movies Ralph Breaks the Internet and The Grinch as examples (all other movies used the same calculations and logic).

Ralph Breaks the Internet	→	20% of groups buying 0 items	→	Item Type 1
	→	40% of groups buying 2 items	→	Item Type 2
	→	30% of groups buying 4 items	→	Item Type 3
	→	10% of groups buying 5 items	→	Item Type 4
The Grinch	→	20% of groups buying 0 items	→	Item Type 5
	→	40% of groups buying 0 items	→	Item Type 6
	→	30% of groups buying 0 items	→	Item Type 7
	→	10% of groups buying 0 items	→	Item Type 8





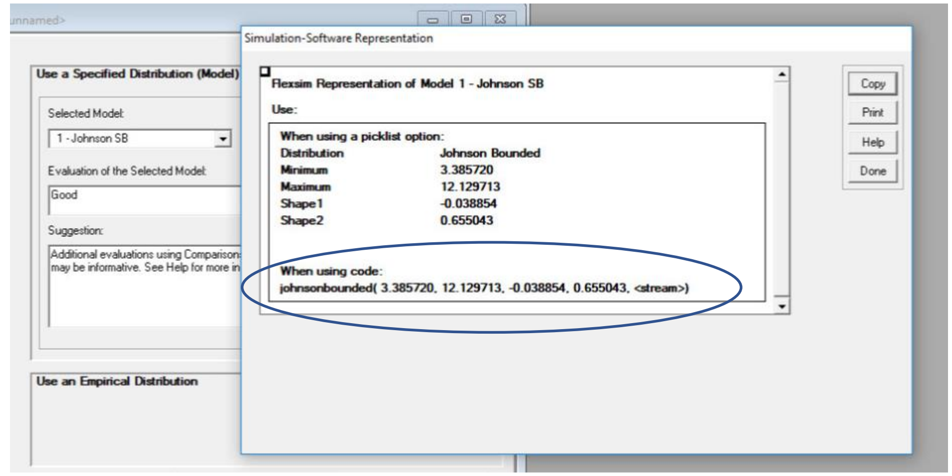
The groups then flow 50% to get their tickets scanned and 50% to buy tickets through either a human or electronic ticket kiosk.

2. Buying tickets or scanning tickets – To buy tickets there are options for the groups to use either one of the three human ticket kiosks or one of the three electronic ticket kiosks. To scan tickets the groups can be processed by one of the two scanning stations. The three human ticket kiosks are operated by an employee each (operator), and the two scanning stations are operated by an employee each (operator). The processing time for the ticket kiosks and the scanning stations were evaluated using Expert Fit. The following is the analysis of the appropriate distributions: (The distributions used in the different processors are circled in the following diagrams)

Human Ticket Kiosk

Relative Evaluation of Candidate Models

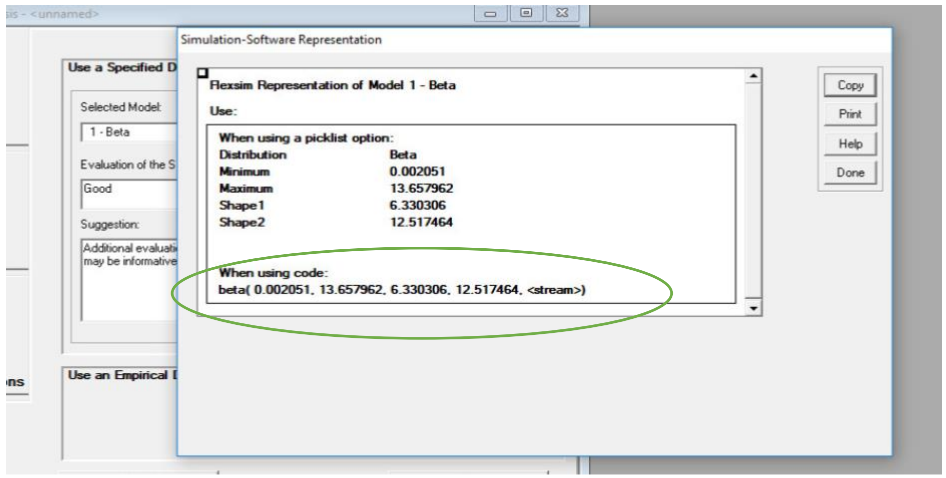
Model	Relative Score	Parameters	
1 - Johnson SB	97.83	Lower endpoint	3.38572
		Upper endpoint	12.12971
		Shape #1	-0.03885
		Shape #2	0.65504
2 - Beta	95.65	Lower endpoint	3.49473
		Upper endpoint	11.99735
		Shape #1	1.10889
		Shape #2	1.05722
3 - Uniform	93.48	Lower endpoint	3.51000
		Upper endpoint	11.98000



Electronic Ticket Kiosk

Relative Evaluation of Candidate Models

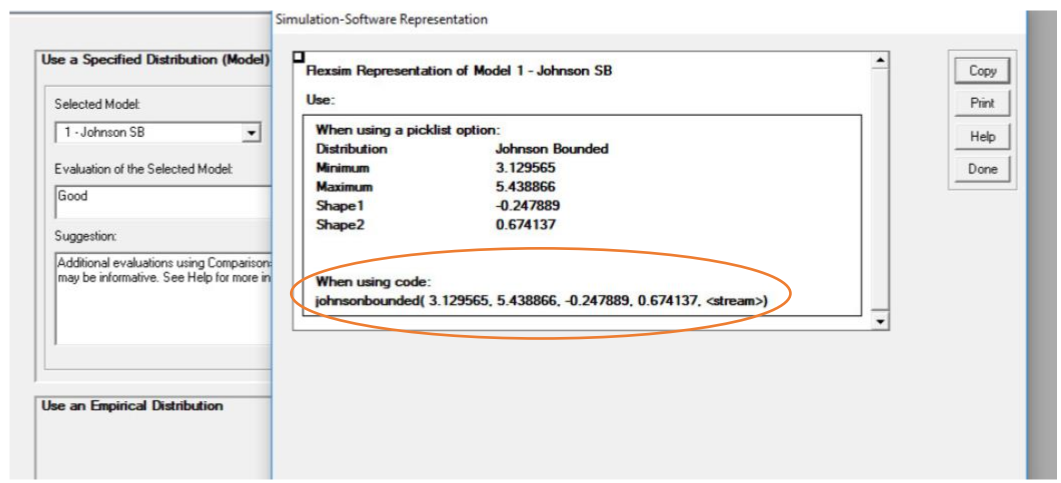
Model	Relative Score	Parameters	
1 - Beta	100.00	Lower endpoint	0.00205
		Upper endpoint	13.65796
		Shape #1	6.33031
		Shape #2	12.51746
2 - Weibull(E)	95.45	Location	0.82274
		Scale	4.22478
		Shape	2.80921
3 - Johnson SB	90.91	Lower endpoint	0.00205
		Upper endpoint	11.75534
		Shape #1	0.86215
		Shape #2	1.80413



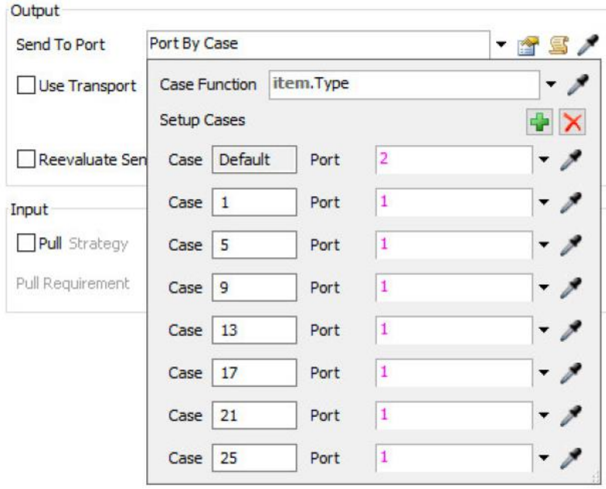
Scanning Station

Relative Evaluation of Candidate Models

Model	Relative Score	Parameters	
1 - Johnson SB	99.04	Lower endpoint	3.12956
		Upper endpoint	5.43887
		Shape #1	-0.24789
		Shape #2	0.67414
2 - Beta	97.12	Lower endpoint	3.15798
		Upper endpoint	5.40684
		Shape #1	1.31916
		Shape #2	1.01025
3 - Weibull(E)	90.38	Location	0.10952
		Scale	4.58826
		Shape	8.35794



The best evaluations in each case were used to produce the most accurate distributions which is subsequently put as the process time for each of the specific stations. After the groups have been processed (either bought tickets or scanned tickets), they flow out into one of either 2 ports. The first flow out port is for the groups of customers that are going straight to the movie theatre without buying any items (0 items). The second flow out port is for the group of customers that are going to buy items at the concession stands (2,4,5 items). This flow out structure is common to all processors at this stage. We used port by case to split each of the customer groups. The customer going to port 1 are the groups with the item type relevant to the 0 items section of each movie. Therefore, following that logic and the explanation in the arrivals section, from the Ralph movie item type 1 should go to port 1, from the Grinch movie item type 5 should go to port 1 and so on. Those only include the customer groups classified as buying 0 items from each movie. The rest of the customer groups (default) go to the second port. The implementation of this in FlexSim is as follows:



The groups going to port 1 are routed to a queue, in this queue the groups are routed to their specific movie theatre by port by case, and the case is item type. For example, for the movie Ralph, its groups have the item types 1,2,3,4 as explained in the arrivals section. The movie the Grinch will have item types 5,6,7,8. And so on.

The groups going to port 2 are routed to a queue as well, in this queue the groups wait in line for the concession stands which is the next stage.

3. Concessions – There are ten concession stands with one employee operating each stand (operator). The process time differs depending on the number of items a customer group is going to buy. The processing time for the different number of items bought were evaluated using Expert Fit. The following is the analysis of the appropriate distributions: (The distributions used in the different processors are circled in the following diagrams)

Buying 2 Items

Relative Evaluation of Candidate Models

Model	Relative Score	Parameters	
1 - Weibull	100.00	Location	0.00000
		Scale	2.87945
		Shape	2.76283
2 - Beta	93.75	Lower endpoint	1.72528 e -4
		Upper endpoint	5.79268
		Shape #1	2.90201
		Shape #2	3.66434
3 - Log-Laplace	85.00	Location	0.00000
		Scale	2.54000
		Shape	2.76878

Use a Specified Distribution (Model)

Selected Model: 1 - Weibull

Evaluation of the Selected Model: Indeterminate

Suggestion: Additional evaluations using Comparison strongly recommended. See Help for mor information.

Flexsim Representation of Model 1 - Weibull

Use:

When using a picklist option:

Distribution	Weibull
Location	0.000000
Scale	2.879454
Shape	2.762828

When using code:

weibull(0.000000, 2.879454, 2.762828, <stream>)

Copy | Print | Help | Done

Buying 4 Items

Use a Specified Distribution (Mo

Selected Model: 1 - Weibull

Evaluation of the Selected Model: Good

Suggestion: Additional evaluations using Compa may be informative. See Help for m

Use an Empirical Distribution

Flexsim Representation of Model 1 - Weibull

Use:

When using a picklist option:

Distribution	Weibull
Location	0.000000
Scale	4.950703
Shape	2.321897

When using code:

weibull(0.000000, 4.950703, 2.321897, <stream>)

Copy | Print | Help | Done

Relative Evaluation of Candidate Models

Model	Relative Score	Parameters	
1 - Weibull	98.68	Location	0.00000
		Scale	4.95070
		Shape	2.32190
2 - Beta	96.05	Lower endpoint	0.00000
		Upper endpoint	10.54394
		Shape #1	2.22535
		Shape #2	3.12867
3 - Erlang(E)	78.95	Location	1.31425 e -4
		Scale	1.10501
		Shape	4

Buying 5 Items

Relative Evaluation of Candidate Models

Model	Relative Score	Parameters	
1 - Beta	97.06	Lower endpoint	1.72528 e -4
		Upper endpoint	18.57721
		Shape #1	1.82191
		Shape #2	2.96534
2 - Weibull	95.59	Location	0.00000
		Scale	7.96499
		Shape	2.00407
3 - Rayleigh	89.71	Location	0.00000
		Scale	7.96193

Use a Specified Distributio

Selected Model: 1 - Beta

Evaluation of the Selected Mo Indeterminate

Suggestion: Additional evaluations using C strongly recommended. See H information.

Use:

When using a picklist option:

Distribution	Beta
Minimum	0.000173
Maximum	18.577212
Shape1	1.821909
Shape2	2.965345

When using code:

beta(0.000173, 18.577212, 1.821909, 2.965345, <stream>)

Since the different number of items bought have different processing time, we used values by case, with the case being item type. Therefore, we grouped all the item types that are considered for each amount and gave it the appropriate processing time. The item types are from the before declaration of items at arrival for each movie (remember that each movie has 4 item types because a customer can buy 0,2,4 or 5 items). The item types for buying only 2 items at concession were 2,6,10,14,18,22,26 (one for each movie) and they used the distribution: weibull(0,2.879454,2.762828, getstream(current)). The item types for buying only 4 items at concession were 3,7,11,15,19,23,27 (one for each movie) and they used the distribution: weibull(0,4.950703,2.321897, getstream(current)). The item types for buying only 5 items at concession were 4,8,12,16,20,24,28 (one for each movie) and they used the distribution: beta(0.000173,18.577212,1.821909,2.965345, getstream(current)). The implementation of this structure in FlexSim is as follows:

Vending Station 1 - Process Time*

```
/***popup:ValuesByCase:valuestr=Time*/
/**Values By Case*/
int case_val = /** \nCase Function: *//***tag:ValueFunc*//**/item.Type/**/;
/** \nCases:\n*/
switch (case_val) {
/***tagex:data*/
    case /**\nCase: *//**/2/**/: return /** Time: *//**/weibull(0,2.879454,2.762828, getstream(current))/**/;
    case /**\nCase: *//**/3/**/: return /** Time: *//**/weibull(0,4.950703,2.321897, getstream(current))/**/;
    case /**\nCase: *//**/4/**/: return /** Time: *//**/beta(0.000173,18.577212,1.821909,2.965345, getstream(current))/**/;
    case /**\nCase: *//**/6/**/: return /** Time: *//**/weibull(0,2.879454,2.762828, getstream(current))/**/;
    case /**\nCase: *//**/10/**/: return /** Time: *//**/weibull(0,2.879454,2.762828, getstream(current))/**/;
    case /**\nCase: *//**/14/**/: return /** Time: *//**/weibull(0,2.879454,2.762828, getstream(current))/**/;
    case /**\nCase: *//**/18/**/: return /** Time: *//**/weibull(0,2.879454,2.762828, getstream(current))/**/;
    case /**\nCase: *//**/22/**/: return /** Time: *//**/weibull(0,2.879454,2.762828, getstream(current))/**/;
    case /**\nCase: *//**/26/**/: return /** Time: *//**/weibull(0,2.879454,2.762828, getstream(current))/**/;
    case /**\nCase: *//**/7/**/: return /** Time: *//**/weibull(0,4.950703,2.321897, getstream(current))/**/;
    case /**\nCase: *//**/11/**/: return /** Time: *//**/weibull(0,4.950703,2.321897, getstream(current))/**/;
    case /**\nCase: *//**/15/**/: return /** Time: *//**/weibull(0,4.950703,2.321897, getstream(current))/**/;
    case /**\nCase: *//**/19/**/: return /** Time: *//**/weibull(0,4.950703,2.321897, getstream(current))/**/;
    case /**\nCase: *//**/23/**/: return /** Time: *//**/weibull(0,4.950703,2.321897, getstream(current))/**/;
    case /**\nCase: *//**/27/**/: return /** Time: *//**/weibull(0,4.950703,2.321897, getstream(current))/**/;
    case /**\nCase: *//**/8/**/: return /** Time: *//**/beta(0.000173,18.577212,1.821909,2.965345, getstream(current))/**/;
    case /**\nCase: *//**/12/**/: return /** Time: *//**/beta(0.000173,18.577212,1.821909,2.965345, getstream(current))/**/;
    case /**\nCase: *//**/16/**/: return /** Time: *//**/beta(0.000173,18.577212,1.821909,2.965345, getstream(current))/**/;
    case /**\nCase: *//**/20/**/: return /** Time: *//**/beta(0.000173,18.577212,1.821909,2.965345, getstream(current))/**/;
    case /**\nCase: *//**/24/**/: return /** Time: *//**/beta(0.000173,18.577212,1.821909,2.965345, getstream(current))/**/;
    case /**\nCase: *//**/28/**/: return /** Time: *//**/beta(0.000173,18.577212,1.821909,2.965345, getstream(current))/**/;/***//**/
    default: return /** \nDefault Time: *//***tag:default*//**/1/**/;
}
/** \nNote:
 Case values must be positive integers.
 Additional cases may be added as needed.
 Use Format:\ncase 1: return 10;*/
return 0;
```

Distribution for buying 2 items

Distribution for buying 4 items

Distribution for buying 2 items

The concession stands are connected to the theatres for the different movies. Hence, the flow out of the stands uses port by case to identify the movie type of the customer and send them to their appropriate movie theatre. For example, since we already defined the item types of the Ralph movie to be 1,2,3,4 then they are all routed to port 1 which is the Ralph movie theatre. The same logic is used for all other movies.

4. Movie theatres – This is the last stage of the project. Each movie has its own theatre. A sink is used to remove the customer groups when they reach the theatre.

Statistics
We performed several statistics for the model. The Following are what we did:

- Wq – expected waiting time in queue
    - For the queue waiting to buy tickets
    - For the queue waiting to get tickets scanned
    - For the queue waiting to buy concession items
- Lq – expected number of groups in queue
    - For the queue waiting to buy tickets
    - For the queue waiting to get tickets scanned
    - For the queue waiting to buy concession items
- Ls – expected number of groups in system. This was done for the whole flow of the system.
- Ws – expected waiting time in system. This was done for the whole flow of the system.

Perhaps, the main striking issue displayed by the statistics performed is that the number of customers in the queue waiting to get tickets scanned exceeds the criteria of small queues required by the model goals (less than 15 groups waiting in line). At the same time, this affects the waiting time of that queue because it as well does not meet the low waiting time goal (entering the movie theatre before the movie starts). Due to the slow processing of scanning the tickets, and the fast processing of buying tickets, the concession queue is always empty because customers go straight to the concession stands without waiting because the number of customers is always lower than 10 (number of concessions stands available).

Analysis of current system and recommendation of improvements

Due to the high demand on the scanning stations, we suggest that there is an increase in the number of scanning stations. This will decrease the waiting time for the groups waiting to get their tickets scanned. On the other hand, due to the fast processing of the buying ticket kiosks, we suggest that there is a reduction in the number of kiosks, as to decrease costs as well as decrease the percentage of idleness in the system.

As well as the changes to the scanning stations and ticket kiosks, we suggest the reduction of the number of concessions stands because they are almost always sitting idle throughout the whole model.

All these increases and reductions have to be both mindful of the costs, efficiency, and quality of the whole system. These recommendations have been implemented in our modified model to satisfy the model goals and increase efficiency as well as reduce costs. We have decided to add 1 more scanning station as to increase the capacity of processing in order to decrease the queue waiting time. At the same time to reduce costs we reduced the number of human ticket kiosks to only 1 instead of 3 and reduced the number of concessions stands to 5 instead of 10. We did it while maintaining and marginally improving the quality of the system. The statistics done on the modified system successfully reached the goals of the model and it clearly shows a higher performing system that should be implemented.

Finally, we do not feel like replacing our human service for electronic vending machines because at this point our current machines are fully capable to withstand the full demand of the system at a cost lower than $25,000. As well as that, our human serviced stations do not have scheduled breakdowns and therefore can run for the whole shift from start to finish.