Lauren Achey
CX4230 – Project 1
Georgia Tech
February 11, 2018

Grocery Store Self-Checkout Simulation

1 Project Description

A grocery store owner wants to know if there is a way to reduce the waiting time for customers with twenty or fewer items, since many of these customers must wait in longer lines behind other customers with larger orders before being able to check out. The store owner wants to know if a self-checkout lane for customers with twenty items or less would be effective so they have requested a simulation of a self-checkout line so they can get an idea of what the wait times would be for the customers with fewer items.

While a self-checkout lane may reduce wait times for customers, the store owner is concerned that the use of a self-checkout lane may lead to increased loss or shrinkage, due to theft. An article published in 2016 reported that nearly one third of customers who use a self-checkout kiosk steal an item or multiple items [1]. After monitoring one million transactions for a year that totaled $21 million in sales, The Atlantic reported that "nearly $850,000 worth of goods left the store without being scanned and paid for" [2]. This amounted to losses of approximately 4%, nearly double the loss attributed to theft in stores without self-checkout lanes [2]. The store owner is hopeful that reducing their staff by replacing a cashier with a self-checkout kiosk will cut costs enough to outweigh the increased losses, but they want to be sure before investing in self-checkout kiosk technology.

2 Simulation Model Software Description

This simulation model is a discrete event simulation for a self-checkout lane that also tracks the losses due to theft from cheating the self-checkout kiosk. The software is divided into two categories: the simulation application and the simulation engine. The simulation engine is a generic simulation engine that keeps track of Events, including a struct for each individual event, and a Schedule function that keeps track of the Future Event List (FEL), the priority queue of events. The events are kept in an ordered linear list in timestamp order. The smallest event is removed from the front of the linear list while events with larger timestamps are added to the back of the list. This allows for fast execution of the priority queue and ensures that events are removed in smallest timestamped order which matches the first-come-first-served nature of self-checkout. The simulation engine also holds the method to run the simulation, the method that implements the main scheduling loop for events. Pseudocode for the run simulation method can be found in Table 2.

The simulation application holds the simulation specific details like the number of customer checkouts, the time per item, checkout and arrival events, and methods for calculating random variables for customer arrival times, number of items, customer transaction total, and loss percentage (used only when theft occurs during checkout). The total sales are calculated during the checkout event, as are the total losses if a loss is occurring.

2.1 Event Handler Procedures

Arrival Event {	**Checkout Event {**

// keeps track of number of customers
// schedules a checkout event
 	
}
	// processes a checkout event
	// schedules another checkout event if there are customers in the queue
	// calculates transaction total and adds it to total sales
	// calculates losses (if any) and adds it to total losses
	
	}

Table 1 - Arrival and Checkout Event Basics

2.2 Event Processing Loop

```
While (simulation not finished)
        E = smallest time stamp event in FEL
        Remove E from FEL
        Now := time stamp of E
        call event handler procedure
```

Table 2 - Pseudocode Representing the Main Event Processing Loop in the Simulation Engine [3]

3 Model Verification

The simulation model is working as anticipated and in accordance with the conceptual model. The simulation has two kinds of events, ARRIVAL and CHECKOUT, which match the events described in the conceptual model. During the Arrival function, if there are customers in the queue for the checkout, the total waiting time is updated which is expected. After updating the total waiting time, the number of events and the number of customers in the checkout area, queue plus the checkout kiosk, is updated. A preset number of arrivals are used each time the simulation is run so before scheduling a new arrival event, a check is done to make sure that the predefined number of arrivals has not been reached. This keeps the simulation from running infinitely. If the checkout kiosk is free, the Boolean is updated to false and a new checkout event is scheduled. The current time is updated to be the time of the last event.

During the Checkout function, the number of events is incremented which is expected. The number of customers in the queue and/or using the checkout kiosk is decremented, which is also an expected behavior. Before scheduling a checkout event for the next customer, a check ensures that there is another customer in the line and/or at the kiosk. This ensures that checkout events are not automatically added to the queue during each Checkout event, and instead are only added to the queue when there is a customer being simulated. If there is no one in the checkout, the kiosk is updated to be free at this point. This is expected behavior as the checkout kiosk is anticipated to be free after the most recent checkout event is completed and if there is no one else in the queue.

The “losses” occur every third customer by checking that the customer count mod three is equal to zero. Every third customer is consistent with the findings in the conceptual model that one third of customers have cheated the self-checkout kiosks. Since there is no way to determine how much a customer has cheated the system, the simulation accounts for the loss due to theft by percentage of loss for the transaction that can vary from 6 and 15%. This range can be varied simply by adjusting the values for this range in the percentage loss method that returns the random percentage.

The model parameters that can be changed represent the time taken, per item, to be scanned through the checkout and the mean interval arrival time of customers at the queue. Included in this per item scanning time is an additional few seconds per item that makes up for the time it takes for payment to be made, processed, and completed. This method of calculating the time taken was based on personal experience using a self-checkout kiosk at the local grocery store. However, there is a detail that could alter the simulation outcomes. The card processing time is relatively independent of how many items a customer has. The way the simulation code is written, each item contains a “portion” of the time that is reflected in the payment processing so more items means allocating more simulation time to that payment process which is not necessary accurate.

The other parameter, the interval arrival time of customers at the queue is based on an exponentially distributed random number with some mean, M. This is the more difficult parameter of the model to emulate as customers do not arrive quite so uniformly. While this suits the conceptual model at the most basic level, additional data to help more accurately model customer arrivals, perhaps during different times of day, may improve the accuracy.

4 Validation

Initially, customers were set to arrive around a time distribution with a mean of 2 minutes, and the time per item scanned was set to half a minute, or 30 seconds to account for scanning, bagging, and card processing time. This resulted in an average wait time of 12.4 minutes for each customer. This seemed unusually long for any of the self-checkout lines in the average grocery store. After some physical experimentation simulating a checkout process, the constant that determines the time per item scanned was reduced to one third of a minute, .33, from .5 minutes. Additionally, the mean arrival time for the arrival distribution was increased from 2 minutes to 5 minutes. This produced a very short average waiting time of 1.43 minutes which seemed too low to accurately reflect the average wait a customer experiences. Therefore, the mean time was reduced from 5 minutes to 4 minutes, and then again to 3 minutes. This produced average waiting times of 4.75 minutes and 5.17 minutes respectively, which is a more realistic approximation based on my personal experience. The various outputs can be viewed in the Appendix in Table 9.

Simulating reasonable losses due to theft proved to be more challenging. Initially, with the random percentage of loss for each third customer (the customer whose transaction would be multiplied by a loss factor and added to the total losses), between a range of 10-45%, unusually high losses were reported. The total loss for this range was above 5%, higher than any of the physical systems which have been observed. Eventually, a range of 6 – 15% was selected. This is somewhat arbitrary because there is no data that reports on the amount of loss per customer transaction that occurs due to theft at self-checkout registers. Higher percentage loss ranges produced very large loss percentages, sometimes at 12 – 20% of the total sales volume. Those percentages were clearly much too high when compared with physical systems. When using the 6 - 15% range, the total losses clustered around 4%. This matches the physical systems which have all reported roughly 4% losses at their self-checkout kiosks, nearly double the loss without them [2]. The sample outputs for losses with varying percentages can be found in Table 10 in the Appendix.

5 Conclusions

In determining the cost effectiveness of adding a self-checkout kiosk, more data on the amount stolen per transaction is necessary to develop a more accurate simulation. There is strong evidence that suggests consumers are more likely to steal at the self-checkout kiosk than they are to steal when a cashier is present at a regular checkout line. With one third of consumers shoplifting at a self-checkout kiosk, significant losses are incurred by the business who chooses to replace a regular checkout line with a self-checkout kiosk. To predict the total losses with more accuracy, it is vital to know roughly how much theft is occurring, in terms of dollar amount or fraction of total sale, because otherwise, an arbitrary percentage is randomly generated and used in place of data from the physical system itself. There is little data available to use to model the percentage range of loss, and this presents a credibility issue for the model. The store owner may wonder just how credible this percentage range is and may choose to alter the percentage ranges to suit a preconceived notion of how much theft is occurring, or to justify the store owner’s personal feelings towards implementing a self-checkout kiosk.

Businesses attempting to reduce theft have placed an attendant on duty by the self-checkout kiosks to monitor transactions, but this costs the business wages for the employee on duty, something the self-checkout lane was meant to reduce or eliminate. The business may be better off just having another cashier working on a regular checkout lane.

Other concerns for the usability of this simulation include the assumption that the self-checkout kiosk will work perfectly all the time. There are no statistics available on how often self-checkout kiosks at grocery stores break down or require maintenance. However, from personal experience, I know that this is not inconsequential for the wait time. Additionally, there are usually restricted items, like alcohol, that warrant the need for an attendant on duty to verify a driver’s license or identification before the purchase can be completed.

5 Appendix

Conceptual Model

5.1 Details of the System Under Investigation (SUI)

The customers with twenty items or fewer who will be using the self-checkout lane will use the self-checkout kiosk one at a time. Use of the self-checkout kiosk is in a first-come-first-serve order. Roughly one third of customers will steal from the self-checkout kiosk.

5.2 Assumptions and Simplifications

The time a customer arrives at the checkout line is independent of other customers in the system. The customer traffic, the time between customers arriving at the checkout line, will follow a probability distribution where intermittent arrival of customers is assumed to be independent and identically distributed (i.i.d.). For the purposes of this model, the assumption is made that all customers in the queue for the checkout kiosk will have twenty items or fewer.

Due to a lack of data on the relative frequency of self-checkout kiosk breakdowns, an assumption is made that the self-checkout kiosk works 100% of the time and does not break down. An additional simplification is made to ignore any purchases that require an attendant to verify an age, such as the purchase of alcohol or any other age restricted items.

According to the Food Marketing Institute, the average sale per customer transaction is $30.02 [4]. Therefore, assume that each customer transaction costs something in the range of $15 - $45 dollars. For simplicity, the average cost per transaction will be independent of the number of items. In accordance with the one third of customers who were reported to have stolen at the self-checkout line, every third customer transaction will have some amount of theft resulting in some "loss" from the total sale.

5.3 Structure of the Model

The model structure consists of entities, components of the model, and attributes, characteristics of the entities used for the model [5]. The entities fall in four categories, resources, consumers, queues, and groups. Groups of entities will not be necessary for this model. Generally, groups represent unordered collections of entities that have some significance to the model but do not require any ordering [6]. In this simulation, there are no unordered entities. Each customer arrives for the checkout kiosk and is placed in a priority queue on a first-come-first-serve basis.

The consumer represents an entity that seeks some kind of service from the model [6]. In this simulation, the customer is the consumer. Each customer has a single attribute, NumberOfItems, which is the number of items the customer is purchasing. The customer seeks to use the self-checkout kiosk service. There is a single queue for customers waiting to use the self-checkout kiosk. The customer who arrived first can use the kiosk first. There is also a single resource in this model, the self-checkout kiosk that the customers waiting in the queue are trying to use.

The queue has the attribute, InTheCheckout, which represents the number of customers in checkout line or currently using the checkout kiosk. The resource, the checkout kiosk, has the attribute KioskFree, which is a Boolean representing whether the kiosk is available, true if it is free for the next customer to use and false if it is currently in use. The parameter that determines how long it takes each customer at the checkout is K. K is a function of the number of items that each customer has. Tables 3 – 6 in the Appendix gives the model information in the ABCMod framework.

5.4 Behavior of the Model

The customer consumer entity has a set of behaviors that can be represented by the activity diagram in Figure 1. The life cycle diagram represents the sequence of actions and activities for a customer. The customer arrives at the checkout line. Then there is a possible delay for the customer depending on whether the kiosk is available. If it is not, the customer must wait. Once the kiosk is free, the customer can use the kiosk for checkout.

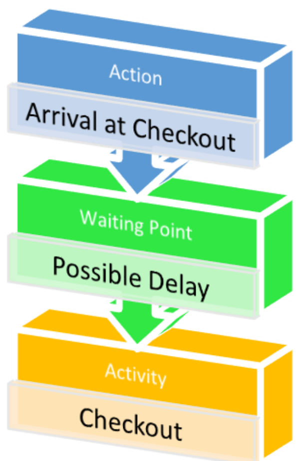

Figure 1 - Customer Life Cycle

5.5 ABCMod Framework Tables

Consumer Class: Customer	
Represents a customer going through the checkout process at the self-checkout kiosk	
Attributes	**Description**
NumberOfItems	The number of items the customer is purchasing, 0 < NumberOfItems <= 20;

Table 3 - Consumer Class of ABCmod Framework

Queue Unary: Checkout Queue	
Represents the queue of customers waiting to use the kiosk, first come first serve	
Attributes	**Description**
InTheCheckout	The number of customers waiting in the checkout line or using the checkout kiosk

Table 4 - Queue Representation of ABCmod Framework

Resource: Checkout Kiosk	
Represented the self-checkout kiosk being used by customers	
Attributes	**Description**

KioskFree	Boolean, true if the kiosk is free and false if it is currently being used

Table 5 - Resource of the ABCmod Framework

Parameters		
Name	**Description**	**Value**
C	The time the consumer uses the kiosk	20seconds * NumberOfItems;

Table 6 - Parameters of the ABCmod Framework

Action: Customer Arrival	
Input stream of customers wanting to use the checkout kiosk to checkout	
Time Sequence	Random variable time specification
Event	InTheCheckout = InTheCheckout + 1

Table 7 - Action Entity Behavior in ABCmod Framework

Activity: Checking Out	
Models a customer using the checkout kiosk to checkout	
Precondition	KioskFree AND <customer at the front of the queue>
Initiating Event	KioskFree = False
Duration	K
Terminating Event	InTheCheckout = InTheCheckout – 1 KioskFree = True

Table 8 - Activity Table for Pulse Detected in ABCmod Framework

5.6 Sample Outputs

Sample Outputs From Trials with Different Simulation Parameters – Waiting Period

```
//Parameters: A = 2.0, NARRIVALS = 10, C = 0.5;
        Welcome to the Self-Checkout Kiosk Simulation
        Initial event list:
        Event List: 1.442559
        Arrival Event: time=1.442559
        Arrival Event: time=1.829100
        Arrival Event: time=3.358532
        Checkout Event: time=3.942559
        Arrival Event: time=9.257296
        Arrival Event: time=9.770776
        Checkout Event: time=10.442559
        Arrival Event: time=11.136273
        Arrival Event: time=12.123709
        Arrival Event: time=12.773025
        Arrival Event: time=13.690981
        Checkout Event: time=14.942559
        Checkout Event: time=18.442559
        Arrival Event: time=21.892172
        Checkout Event: time=24.942559
        Checkout Event: time=26.442559
        Checkout Event: time=32.942559
        Checkout Event: time=42.442559
        Checkout Event: time=44.942559
        Checkout Event: time=48.442559
```

```
                          Number of customers = 10
                          Total waiting time = 123.651165
                          Average waiting time = 12.365116
                          20 events executed in 0.000000 seconds (Inf events per second)

//Parameters: A = 2.0, NARRIVALS = 10, C = 0.33;
                          Welcome to the Self-Checkout Kiosk Simulation
                          Initial event list:
                          Event List: 1.442559
                          Arrival Event: time=1.442559
                          Arrival Event: time=1.829100
                          Checkout Event: time=3.092559
                          Arrival Event: time=3.358532
                          Arrival Event: time=3.735380
                          Arrival Event: time=6.158255
                          Arrival Event: time=6.671735
                          Arrival Event: time=8.037232
                          Checkout Event: time=8.042559
                          Arrival Event: time=8.303656
                          Arrival Event: time=9.291092
                          Arrival Event: time=9.940409
                          Checkout Event: time=11.012559
                          Checkout Event: time=14.972559
                          Checkout Event: time=17.942559
                          Checkout Event: time=18.602559
                          Checkout Event: time=20.912559
                          Checkout Event: time=27.512559
                          Checkout Event: time=31.802559
                          Checkout Event: time=35.102559
                          Number of customers = 10
                          Total waiting time = 96.567637
                          Average waiting time = 9.656764
                          20 events executed in 0.000000 seconds (Inf events per second)


//Parameters: A = 5.0, NARRIVALS = 10, C = 0.33;
                          Welcome to the Self-Checkout Kiosk Simulation
                          Initial event list:
                          Event List: 3.606397
                          Arrival Event: time=3.606397
                          Arrival Event: time=4.572751
                          Checkout Event: time=5.256397
                          Arrival Event: time=8.396331
                          Arrival Event: time=9.338449
                          Checkout Event: time=10.206397
                          Checkout Event: time=12.186397
                          Checkout Event: time=14.496397
                          Arrival Event: time=15.395639
                          Arrival Event: time=16.061698
                          Checkout Event: time=18.365639
                          Arrival Event: time=18.530288
                          Checkout Event: time=20.675639
                          Arrival Event: time=20.825177
                          Checkout Event: time=23.645639
                          Arrival Event: time=24.658047
                          Checkout Event: time=25.955639
                          Arrival Event: time=29.857026
                          Checkout Event: time=30.245639
                          Checkout Event: time=33.545639
                          Number of customers = 10
                          Total waiting time = 14.297618
                          Average waiting time = 1.429762
```

```
        20 events executed in 0.000000 seconds (Inf events per second)


//Parameters: A = 3.0, NARRIVALS = 10, C = 0.33;
        Welcome to the Self-Checkout Kiosk Simulation
        Initial event list:
        Event List: 2.163838
        Arrival Event: time=2.163838
        Arrival Event: time=2.743651
        Checkout Event: time=3.813838
        Arrival Event: time=5.037799
        Arrival Event: time=5.603070
        Checkout Event: time=8.763838
        Arrival Event: time=9.237383
        Checkout Event: time=10.743838
        Arrival Event: time=11.285628
        Arrival Event: time=11.548513
        Arrival Event: time=13.029667
        Checkout Event: time=13.383838
        Arrival Event: time=14.003643
        Checkout Event: time=17.343838
        Checkout Event: time=18.003838
        Checkout Event: time=20.313838
        Arrival Event: time=26.305429
        Checkout Event: time=26.913838
        Checkout Event: time=31.203838
        Checkout Event: time=34.503838
        Number of customers = 10
        Total waiting time = 51.689760
        Average waiting time = 5.168976
        20 events executed in 0.000000 seconds (Inf events per second)


//Parameters: A = 4.0, NARRIVALS = 10, C = 0.33;
        Welcome to the Self-Checkout Kiosk Simulation
        Initial event list:
        Event List: 2.885117
        Arrival Event: time=2.885117
        Arrival Event: time=3.658201
        Checkout Event: time=4.535117
        Arrival Event: time=6.717065
        Arrival Event: time=7.470760
        Checkout Event: time=9.485117
        Checkout Event: time=11.465117
        Arrival Event: time=12.316511
        Arrival Event: time=12.849358
        Arrival Event: time=13.199872
        Checkout Event: time=13.775117
        Arrival Event: time=15.174744
        Checkout Event: time=16.085117
        Arrival Event: time=17.010655
        Checkout Event: time=19.055117
        Arrival Event: time=20.076951
        Checkout Event: time=21.365117
        Checkout Event: time=27.965117
        Checkout Event: time=32.255117
        Checkout Event: time=35.555117
        Number of customers = 10
        Total waiting time = 47.511941
        Average waiting time = 4.751194
        20 events executed in 0.000000 seconds (Inf events per second)
```

Table 9 - Sample Simulation Outputs from Trials Varying the Simulation Parameters

Simulation Outputs from Trials Varying Loss Percentage
Percentage of loss range: 10 – 45%: Number of customers = 50 Total waiting time = 625.293931 Average waiting time = 12.505879 Total sales = 1332.87 Total losses = 115.12 Percentage lost = 8.64%
Percentage of loss range: 10 – 30% Number of customers = 50 Total waiting time = 625.293931 Average waiting time = 12.505879 Total sales = 1332.87 Total losses = 90.91 Percentage lost = 6.82%
Percentage of loss range: 10 – 25% Number of customers = 50 Total waiting time = 625.293931 Average waiting time = 12.505879 Total sales = 1332.87 Total losses = 82.84 Percentage lost = 6.22%
Percentage of loss range: 10 – 20% Number of customers = 50 Total waiting time = 625.293931 Average waiting time = 12.505879 Total sales = 1332.87 Total losses = 74.78 Percentage lost = 5.61%
Percentage of loss range: 10 – 15% Number of customers = 50 Total waiting time = 625.293931 Average waiting time = 12.505879 Total sales = 1332.87 Total losses = 66.71 Percentage lost = 5.00%
Percentage of loss range: 6% - 15% Number of customers = 50 Total waiting time = 625.293931 Average waiting time = 12.505879 Total sales = 1332.87 Total losses = 49.71 Percentage lost = 3.73%

Table 10 - Sample Outputs from Validation Testing of Loss Percentage Ranges

6 References

[1] F. Coletta, "Are you a swiper?," August 4th, 2016. Accessed on: March 4th, 2018Available: http://www.dailymail.co.uk/news/article-3722820/Beware-swipers-ONE-shoppers-steal-self-serve-checkout-leading-criminologists-name.html

[2] R. Chun, "The Banana Trick and Other Acts of Self-checkout Thievery," *The Atlantic,* no. MARCH, 2018.
[3] R. Fujimoto, "Parallel and distributed simulation," in *Proceedings of the 2015 Winter Simulation Conference*, 2015, pp. 45-59: IEEE Press.
[4] FMI. (2018). *Supermarket Facts*. Available: https://www.fmi.org/our-research/supermarket-facts
[5] S. Robinson, "Conceptual modeling for simulation," in *Simulation Conference (WSC), 2013 Winter*, 2013, pp. 377-388: IEEE.
[6] G. Arbez and L. G. Birta, "The ABCmod conceptual modeling framework," *Conceptual Modeling for Discrete-Event Simulation*, pp. 133-178, 2011.