Proceedings of the 2015 Winter Simulation Conference
L. Yilmaz, W. K. V. Chan, I. Moon, T. M. K. Roeder, C. Macal, and M. D. Rossetti, eds.

SIMULATION MODELING OF CUSTOMER CHECKOUT CONFIGURATIONS

Manuel D. Rossetti
Anh T. Pham

University of Arkansas
Department of Industrial Engineering
4207 Bell Engineering Center
Fayetteville, AR 72701, USA

ABSTRACT

A simulation model based on a case study of a retail customer check out area and two extended models is presented. The first extended model examines the customer's criteria when picking a checkout lane. The second extension examines the checkout layout, in which the payment is separated from the checkout station. The results show no significant difference in checkout time based on the lane choice criteria. However, the average waiting time drops significantly when payment is separated from the checkout area.

1 INTRODUCTION

A successful retail business not only has good inventory and management systems, but also has good customer service. Making sure that the customers get what they need within their time limits is also a part of satisfying the customers. This has led retail stores to attempt to optimize the checkout processes for their customers. The study of "what to do for fastest checkout time" has resulted in the opening of expressed checkouts, a mixture of regular checkouts and express checkouts, and experimenting with the combination between express checkouts, mobile scanning apps, and regular checkouts. However, there are different perspectives on the studying of checkout efficiency. Many retail stores consider scanning items and processing payments as one process. In this situation, a regular cashier will scan a customer's items and process payments at the checkout station. Other retail stores have tried to break scanning and processing payment into two individual processes, which results in separate stations. Also, while many customers choose their checkout station by comparing the number of customers standing in line of various checkout stations, many others compare the number of items waiting to be scanned for their final destination before leaving the store. This paper simulates a combination of different layouts with different rules for picking the checkout lines.

A number of previous papers, regarding retail stores research using simulation, are briefly introduced in section 2. Section 3 explains the model's assumptions and configurations. Section 4 gives an overview of the implementation of the simulation models and section 5 provides initial results. The paper concludes with ideas for future work.

2 LITERATUR REVIEW

Retail store operations have long been an interest area for many researchers. This includes staffing schedules, shop layout, and customer flows, etc. The use of computer programming has been proposed not only for the retail store's manager but also for researching related topics. Melachrinoudis and Olafsson (1992) claimed: that "using a computer to schedule cashiers is a way to directly improve operations, providing the best customer service possible while eliminating unnecessary labor cost." They

978-1-4673-9743-8/15/$31.00 ©2015 IEEE

studied an example using three different computer programs to come up with a staffing schedule. The staffing schedule was based on a day-to-day, weekly basis. Moreover, Opara-Nadi (2005) compared express checkouts and regular checkouts. Different sets of data were collected and many hypothesis tests were performed. The results suggested that in terms of time efficiency, express checkouts were better. However, there are more chances for the express checkouts to have errors than the regular checkouts.

Instead of just using computer methods to solve mathematical programming problems, there have been efforts using simulation. Williams et al. (2002) attempted to use the SIMUL8 simulation software package, discrete event simulation process model, to determine the staffing policies at a retail store. Different scenarios were taken into account for the objective of minimizing the waiting time of a customer at a checkout. Alvarado and Pulido (2008) studied the combination of cashiers and baggers with respect to the expected time of the customers in the system using Promodel. Data was collected from a specific series of supermarkets in Colombia over 69 consecutive calendar days. The result suggested that there is a significant need to have the baggers when 40% of the cashiers are open. This study's results may vary from location to location, therefore the same procedure can be applied for a different store but the result is not guaranteed to be the same. Miwa and Takakuwa (2008) examined the customer flow in a retail store using Point of Sale (POS) data. ARENA 3D was used for the animation model where the author compared (using hypothesis tests) the "as-is" scenario with some "what-if" scenarios. Because the experimental retail store was the one from a university, there may be the potential to apply this procedure to a bigger box store. Schimmel (2013) studied the effects of express checkouts to waiting lines in a supermarket. The author explored two different queuing models using ARENA: M/G/s (best case) and M/G/1 (worst case), then compared the results. The conclusion was that regular checkouts and express checkouts experience larger queues even though express checkout customers may experience shorter waiting time.

Besides discrete event modeling, agent-based simulation (ABS) has been considered Chan et al. (2010) examined "a hybrid discrete-continuous simulation model with proactive, autonomous, and intelligent entities". This approach has been used to study customer behaviors within the stores. Takao et al. (2009) investigated the customer flow in the supermarket using agent-based simulation and POS data. More specifically, virtual customer footsteps were investigated to see the effect that promotions have inside the store. They were able to make some recommendations on how to locate the promotions. Kubera et al. (2010) believed that a simulation based on customer behaviors should be done through an interaction-oriented approach, especially with large-scale simulation (retail store size of 80,000 items). All the parameters in the stores became agents (client, items, entrance, checkout, indication panels, supermarket staffs, and shelves.) Schwenke et al. (2010) is distinguished from other work because there was a focus on other parameters such as the prognosis of buying behaviors, consumer driven events associated with real events. The customer is modeled as an agent with three main actions: think, move, action. A year later, Schwenke et al. (2011) showed how the authors were able to combine a data-mining algorithm with simulation to "generate" the receipt data. Positive results indicated that simulation can be a method to help with the prediction of sales in a more realistic manner. Yamane (2012) investigated the best checkout layout for a specific retail store in Japan. The authors tried to eliminate congestion issues. This can also be extended to a combination between congestion and time in queue.

In this paper, first we look at how resources affect the time in the checkout process. Then, we examine two factors (lane picking preferences and checkout station layout) and their effects on the customer's total time in the system. When picking the checkout lane, customers can either choose the lane that has the smallest number of people (number of carts) or the smaller number of items waiting to be checked-out. Also, while many stores have their checkout stations regulate three different tasks at the same time: scanning the items, bagging the items, and processing the payments; there may be the possibilities of individualizing the tasks, in which the scanning and bagging station is not the same as the payment station. The next section discusses the modeling of the system.

3 CONCEPTUAL MODELING

In order to investigate layouts and customer checkout lane selection rules, we based our model on the First Annual IIE/Arena Contest problem: The SM Superstore (Kelton et al. 2007). This allows for a realistic modeling situation from a well-known problem to serve as the basis for this research: permitting reproducibility and extensibility. Figure 1 shows the layout of the super store. First, a base model was built based closely on the original contest problem. Then, additional assumptions were added for the data that the contest problem did not provide.

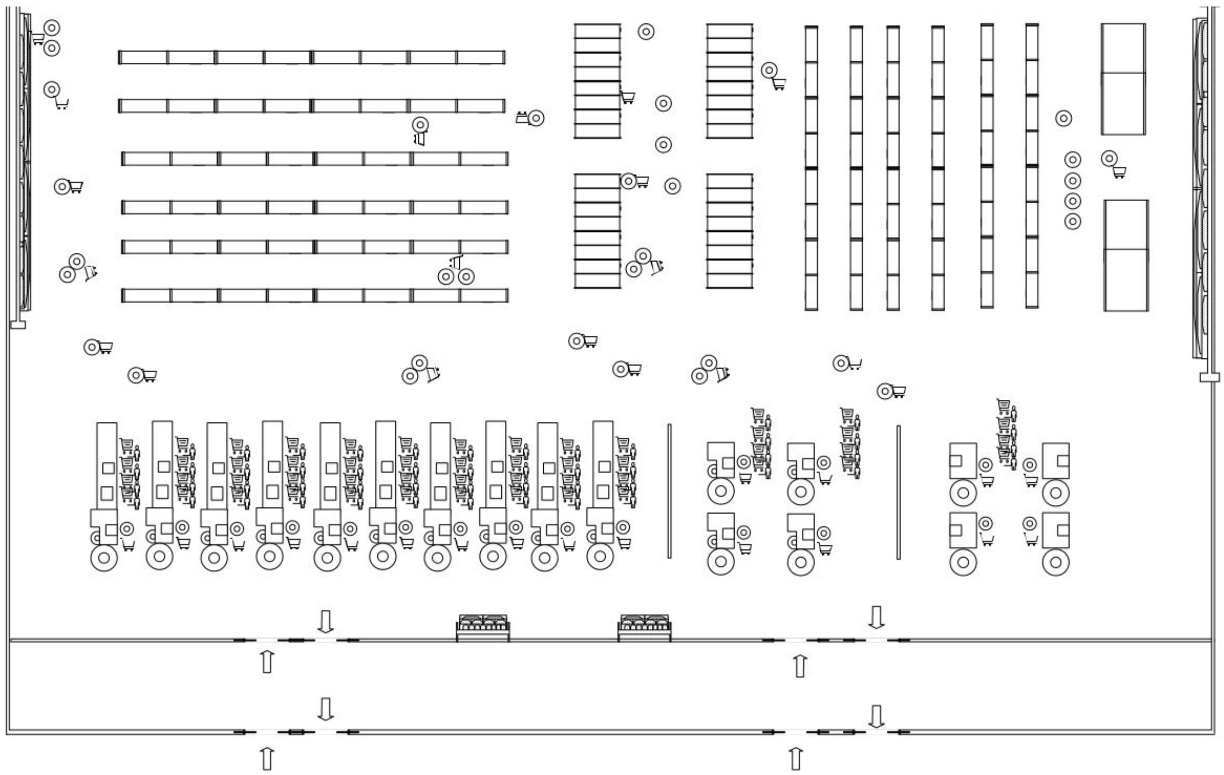

Figure 1: Super store layout.

There are several elements in this SM Superstore case study. Upon arrival, customers start shopping. Each customer shops for a number of items. When the customers finish shopping, they proceed to the check out area. The checkout process includes: picking a checkout lane, waiting in line if checkout lanes are busy, having items scanned, having items bagged, and making payment. In the base model, if no cashier is available, the customer picks the shortest checkout lane and joins the queue. The time it takes to finish the checkout process is dependent on the number of items and the payment method.

In the case study, a set of data of for the number of items purchased is provided. Also, the arrival rates during a regular business day are given. The arrival rates vary with time. Thus, a non-homogeneous Poisson process was used to model the customer arrival process. The distributions of shopping time per item, scanning time per item, bagging time per item and payment time by payment method are given. In addition, the costs of hiring a cashier per hour and the cost of employing a bagger per hour are provided. Baggers will move around and help with bagging the items. We assume that the number of cashiers is limited to 20 and number of baggers is limited to 5. The checkout lanes are single queues with single servers. More specifically, each cashier only serves a customer at a time. There are individual waiting lines for each checkout cashier. By increasing or decreasing the number of checkout lanes and/or the number of baggers, we can investigate how baggers affect the total checkout time.

3.1 Extended Model 1

In the first model extension, customers can either be in a hurry (70%) or not. Once the customer has arrived to the checkout area, the customer picks a checkout station (self checkout, express checkout, or regular checkout.). Customers may scan, bag and pay for their goods at self-checkout stations. Express checkouts are similar to regular checkouts. However, express checkouts only checkout customers with less than 20 items in their carts. Each express and regular checkout stations has a cashier. There may or may not be baggers at each checkout area. If a bagger is not available, the cashier bags items for customers. Self and express checkouts have multiple servers and a single queue. Regular checkout lanes have a single queue and a single server.

The process of picking a checkout lane is different from the base model. Customers can pick a checkout lane within their line of sight. The customer can choose to either keep walking along the checkout area to find the best checkout lane or simply stay for the best checkout lane within their view. Rush customers walk faster, they usually choose the first shortest lane available within their view. Regular customers may think twice before they actually choose a checkout queue. Regular customers may not like the checkout area that they are located within and may want to walk to a different checkout area. Since most of the checkout stations are regular checkouts, there is a high chance the customer's first set of checkout stations will be within regular checkouts. Each customer can only see and compare queues from three to four consecutive checkout stations from the nearest checkout station. Due to the structure of the layout, once a customer has entered a checkout lane and entered the queue, we assume the there is no jockeying.

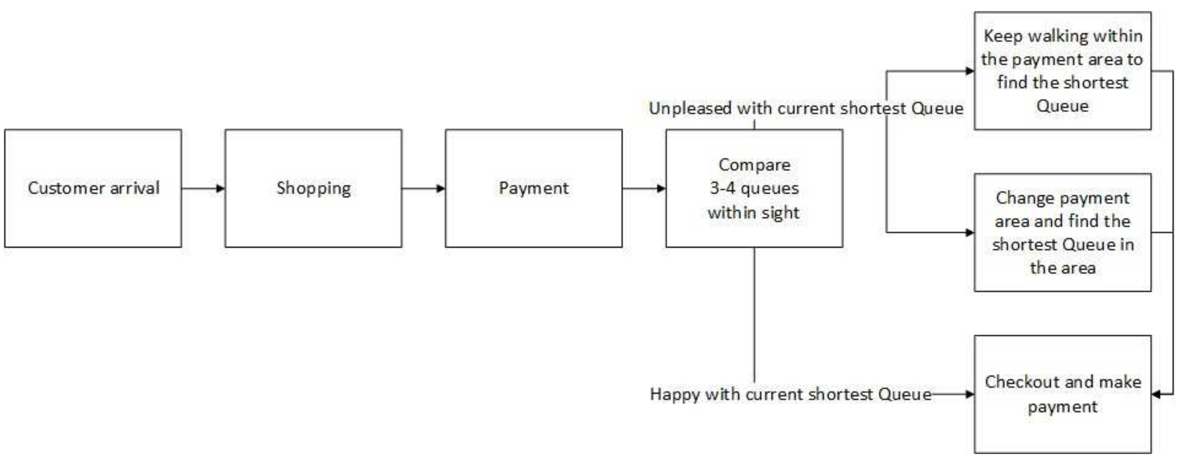

Figure 2: Customer's flow in the store.

3.2 Extended Model 2

The checkout process includes: scanning the items, bagging the items and making/processing payments. In previous models, a checkout station does all the tasks in the checkout process. In this extension, the making/processing payments task has its own area/stations. After a customer has their items scanned and bagged, the customer walks to the payment area and picks a payment station. There are two payment stations in the payment area: express payments and regular payments. Express payments only accept credit cards and act as a self pay station. Regular payments accept a check card and cash. Customers from self-checkout can directly make payments at the self-checkout area.

4 SIMULATION MODELING

This section presents an overview of the implementation details associated with the simulation model. In this research, the model is conceptualized and developed following object-oriented principles and is built upon the Java Simulation Library (JSL). Rossetti (2008) developed an object-oriented framework for simulation modeling. The JSL can facilitate the modeling of discrete-event systems in Java using the event and process world-views with full support for random variable generation and statistical output collection. In term of structure, most of the classes inherit from base classes within the JSL.

- *ModelElement* and *SchedulingElement* - A ModelElement is a base class within the JSL that enables the simulation capabilities. A SchedulingElement is a ModelElement that facilitates the scheduling of events. Every simulation element in the JSL is a specialization of ModelElement.
- *QObject*- A QObject can be used as a base class for objects that need to be placed in queues on a regular basis. Customers within the store are modeled as sub-classes of QObject.
- *EventGenerator*- Class EventGenerator functions similarly to the entity creation constructs within commercial simulation packages. An event generator is used in this model to generate customer arrivals according to a non-homogenous Poisson process.
- *RandomVariable* – Instances of RandomVariable represent randomness through the calling of the getValue() method. To construct a RandomVariable the user provides an instance of a class that implements the RandomIfc interface as the initial random source. The random components of the system, such as the number of items, shopping time, checkout time, and payment time are all built using instances of the RandomVariable class.

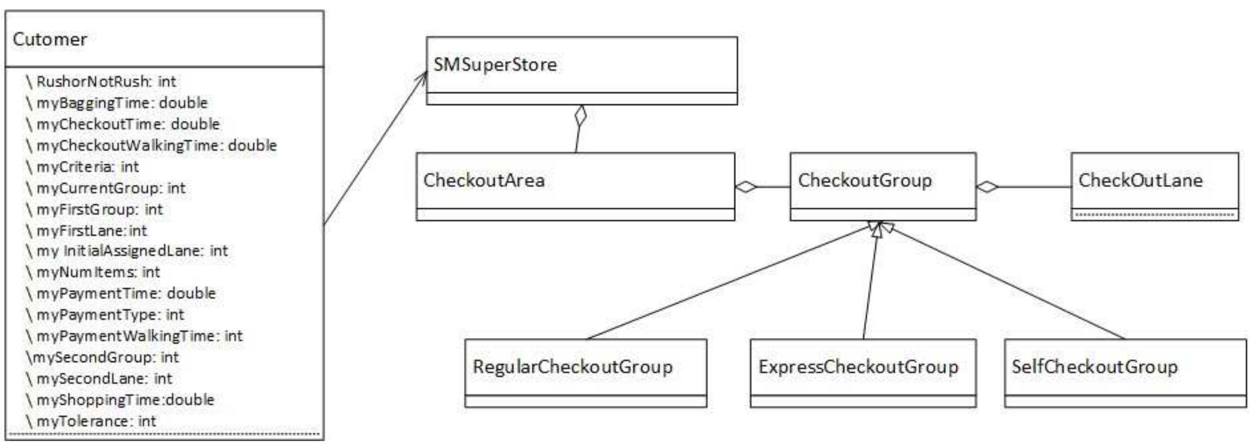

Figure 3: Relationship between classes in extended model 1.

Figure 3 illustrates the relationships between classes within the model. The following major classes are used to represent the store:

- *SMSuperStore* – The purpose of this class is to represent the entire store. An instance of this class holds instances of the other modeling elements used to represent the store and the processing of customers. The class also facilitates the inter-object communication between the modeling elements.
- *CheckoutArea* – The purpose of this class is to represent the area where the checkout process occurs. An instance of a CheckoutArea holds instances of the class CheckoutGroup.
- *CheckoutGroup* – This class serves as an abstract base class that characterizes the types of checkout stations being accessed, such as self-checkout (SelfCheckoutGroup), express checkout (ExpressCheckoutGroup), and regular checkout (RegularCheckoutGroup). Each checkout group holds one or many, individual checkout lane(s) (CheckoutLane).

- *CheckoutLane* – The purpose of this class is to represent the location where a customer is processed during checkout. An instance of a CheckoutLane uses a queue to hold waiting customers and represents the worker that is assigned to processing the customers.
- *Customer* – This class represents the entity that moves through the system and that is processed at check out lanes. A customer has a number of items, a shopping time, a check out time, a bagging time, and a payment time. These attributes are based on how many items are associated with the customer and are determined probabilistically based on the information provided in the case study.

4.1 Base Model

There are a number of key modeling issues that required implementation. This section overviews the approaches taken when implementing these issues: 1) generating customers, 2) choosing lanes, 3) processing at checkout, and 4) specific model extensions.

A non-homogeneous Poisson process was used to model the customer arrival process. The class NHPPEventGenerator was used to generate non-homogenous Poisson arrivals.

```
myNHPPGenerator = new NHPPEventGenerator(this, myPWRF, new CustomerArrival());
```

In code above, an instance of NHPPEventGenerator (myNHPPGenerator) is created. The instance of myNHPPGenerator is within the instance of SMSuperstore. A rate function was created and is stored in myPWRF. This rate function is an instance of the PiecewiseRateFunction class from the JSL which represents the piecewise constant arrival rates. An instance of CustomerArrival() is used to make instances of the customers.

The customer's attributes include: number of items, a shopping time, a check out time, a bagging time, and a payment time. All the random variables such as RushOrNot, WalkingTime, BaggingTime, NumItemGenerator, PaymentTime, ShoppingTime, and ToleranceLimit are associated with the SMSuperstore. They are subclasses of RandomVariable. RushOrNot tells if a customer is an express customer or not. WalkingTime models the walking velocity of the customer. NumItemGenerator, ShoppingTime, BaggingTime, and PaymentTime model the number of items that customer's have in their cart, the time it takes for a customer to shop, the time it takes to bag the items, and the time it takes for a customer to pay. In the Customer class, these characteristics are stored as different attributes.

When choosing a checkout lane, most customers pick the shortest checkout lane. Using this approach, the base model implements a method that compares all the available checkout lanes in the checkout area and then assigns the shortest checkout lane to the customer. Within the class CheckoutArea, the method "findShortestLine()" helps check and return the shortest checkout line.

```
private CheckOutLane findShortestLine() {
        CheckOutLane shortest = null;
        int min = Integer.MAX_VALUE;
        for (CheckOutLane t : myCheckOutLanes) {
            int n = t.getNumberInStation();
        if (n < min) {
                shortest = t;
                min = n;
            }
        }
        return (shortest);
    }
```

A CheckoutLane is a checkout lane in the CheckoutGroup. In a checkout lane, there can be one waiting line and several cashiers, or one dedicated cashier for each waiting lane. Object checkoutLane is a SingleQueueStation from the JSL. A SingleQueueStation can receive an object (customer), then check if there is a server available. If a server is not available, it enqueues the object (customer). When a server is available, the queue will be checked and if there are objects in queue, the next object will be selected according to the specified queue discipline.

In the base model, the baggers are modeled as a resource (BaggingResource). Instances of BaggingResource are associated with instances of CheckoutLane. In the checkout process, the events for modeling the activities associated with check out are modeled. For example, when a customer completes checkout and payment, they need to have their items bagged. The following code illustrates the requesting of the bagging resource for a customer. This models the baggers as a floating resource that can move between lanes as needed. If a bagger is not available, then the cashier performs the bagging task.

```
class EndCheckOutAction implements EventActionIfc {
        @Override
        public void action(JSLEvent event) {
            // customer ended checkout and payment
            Customer dc = (Customer) event.getMessage();

            // check if bagging resource is available
            if (myBaggingResource.isServerAvailable()) {
                // tell bagger to handle bagging
                myBaggingResource.receiveCustomer(dc);
                // bagger available, become idle
                myNumBusy.decrement();
                myNS.decrement();
                // check if customers in lane
                if (isQueueNotEmpty()) {
                    // if someone waiting, serve them
                    serveNext();
                }
            } else {
                // cashier handles bagging
                double st = dc.myBaggingTime;
                //schedule end of bagging
                scheduleEvent(myEndBaggingAction, st, dc);
            }
        }
    }
```

4.2 Extended Model 1

While the base model has only one type of checkout group (regular checkouts), the first model extension has three different checkout groups. There is an additional attribute that customizes the customer's behavior based on the criteria used to pick a checkout lane. The customer is able to move between check out groups. Because each customer has a small set of available checkout lanes (3-4 checkout lanes) to pick from, the customer can change to a different set of checkout lanes. Directly associated with CheckoutGroup, the class LaneChoiceDistribution holds a probabilistic distribution to model the probability of the customer moving to a different group or moving to a different checkout lane.

While standing within the check out area, a random group is assigned to the customer. This random group is "saved" as the customer's attribute "myFirstGroup". A random checkoutLane within "myFirstGroup" is then generated, called "myFirstLane". An instance of LaneChoiceDistribution returns a value that tells if the customer desires to pick a checkout lane from a different checkout group or if the customer wants to pick a checkout lane from another set of available checkout lanes. Attribute "mySecondGroup" is updated if the customer moves to another checkout group. Attribute "mySecondLane" is then also updated. Values of "myFirstGroup", "myFirstLane", "mySecondGroup", and "mySecondLane" are used to determine the additional time customers walk from one checkout group to another and from one checkout lane to another, which is the attribute "myCheckoutWalkingTime", which can effect the total time a customer spends in the store.

As mentioned in section 3, a checkout criterion refers to the criteria that the customer uses when picking the checkout lane. In general, a customer compares between the numbers of carts in checkout lanes, then picks the lanes with the shortest amount of carts; or, a customer compares the number of items in each checkout lane and pick the lane with the smallest amount of items waiting to be checked out.

4.3 Extended Model 2

In the second extended model 2, because the payment stations are separated from the check out stations customers will be sent to a separate payment area (PaymentArea) after the checkout has been completed. The structure of the PaymentArea is similar to the structure of CheckoutArea. The checkout process is divided into two parts: scanning/ checkout and making payment. Because the payment is separated from the checkout process, a new area is created. Class PaymentArea holds an instance of PaymentGroup. Class PaymentGroup holds one or more instances of paymentStations. There are two instances of PaymentGroup: Express and Regular. Similarly to CheckoutGroup, the number of payment lanes (numLanes), number of payment stations within lanes (numServersWithinLane) have to be declared.

```
PaymentGroup (PaymentArea parent, int numLanes, int numSeversWithinLane, int nameGroup, String name)
```

In the following section, the experimental methods and results are discussed.

5 EXPERIMENTAL METHODS AND RESULTS

The basic scenarios explored within the experiments are provided in Table 1. Scenario 1 represents the base case of 20 cashiers and 5 baggers. This scenario is used as a baseline to verify and validate the models. Scenarios 2-5 represent different configurations that explore the effect of lane choice and separate payment stations.

Table 1: List of scenarios.

Scenario #	Description
1	Base Model: Checkout area with 20 cashiers, 5 baggers
2	Extended model 1: Customers choose checkout lanes by number of carts; 20 available regular checkout lanes, each self checkout has 4 checkout stations, two express checkout lanes, and two express checkout stations in each express checkout lane
3	Extended model 1: Customers choose checkout lanes by number of items; 20 available regular checkout lanes, each self checkout has 4 checkout stations, two express checkout lanes, and two express checkout stations in each express checkout lane
4	Extended model 2: 50% customers choose lane by number of carts, checkout with no separate payment; 18 available regular checkout lanes, each self checkout has 4 checkout stations, two express checkout lanes, and two express checkout stations in each express checkout lane.
5	Extended model 2: 50% customers choose lane by number of carts, checkout with separate payment stations; 18 available regular checkout lanes, each self checkout has 4 checkout stations, two express checkout lanes, and two express checkout stations in each express checkout lane; 4 payment stations in total; 2 stations are express payments and two are regular payments

For all the configurations, 30 replications were executed with the run length set at 960 minutes and the warm up time set at 480 minutes in order to model one eight hour day of operation. The main outputs are the average time a customer spends in the store, the average time customers wait in queue, the average waiting time for each type of checkout station, and the average waiting time and system time for payment stations. The time from when a customer arrives to when a customer leaves the store and the time customers wait in queue at checkout lanes/payment lanes are also captured.

To verify that the model was working correctly traces were performed to ensure that the program was working as intended. In addition, by using the arrival rate associated with a single period out of the day, the performance of a single check out station was compared to the results from a M/G/1 queue to confirm that the results were as expected. In addition, a sensitivity analysis was performed by changing the number of lanes available within the base case. The results are shown in Figure 4. As indicated in the figure, the time that customers wait in line increases as the number of regular checkout lanes decreases. Hence, the average total time that customers spend in store also increases.

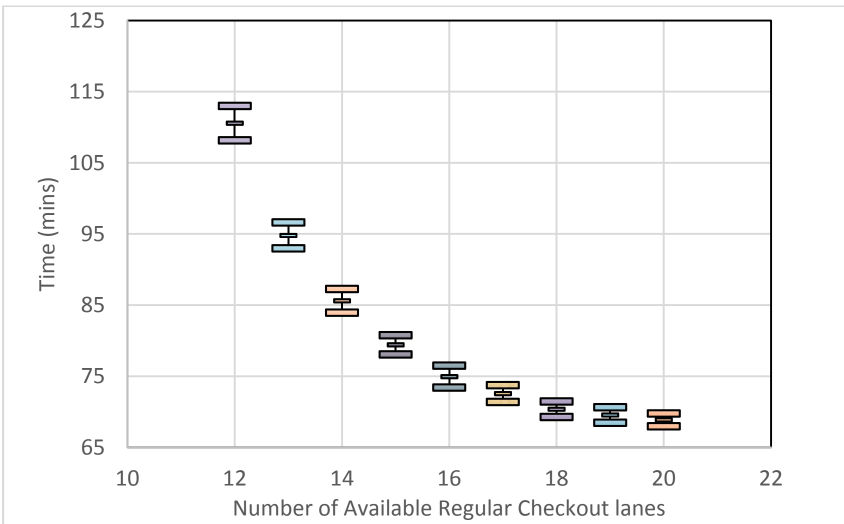

Figure 4 Total time in system vs. number of checkout lanes (CI 90%).

Table 2: Results of scenarios.

	Scenario 1 Avg. (s.d.)	**Scenario 2 Avg. (s.d.)**	**Scenario 3 Avg. (s.d.)**	**Scenario 4 Avg. (s.d.)**	**Scenario 5 Avg. (s.d.)**
#Shoppers in Store	106.85 (4.012)	106.15 (2.887)	106.15 (2.887)	106.15 (2.887)	105.86 (3.487)
Time Shopping	53.63 (.628)	53.48 (0.639)	53.48 (0.639)	53.48 (0.639)	53.55 (0.541)
Items Per Customer	89.29 (1.069)	88.99 (1.198)	89.00 (1.201)	88.96 (1.213)	89.13 (853)
Payment Time	1.38 (.016)	1.39 (0.015)	1.39 (0.016)	1.39 (0.014)	1.39 (0.015)
Checkout Time	7.04 (.144)	7.00 (0.162)	7.00 (0.167)	6.99 (0.164)	6.99 (0.125)
Bagging Time	1.86 (.022)	1.85 (0.025)	1.85 (0.025)	1.85 (0.025)	1.86 (0.018)
#In Store	140.10 (10.53)	135.85 (8.69)	134.71 (8.074)	139.20 (7.682)	128.54 (4.785)
Time In Store	70.30 (3.164)	68.27 (3.421)	67.68 (2.878)	69.77 (2.953)	64.95 (0.94)
Total Wait Time	6.37 (2.858)	5.92 (3.347)	5.33 (2.863)	7.46 (2.788)	2.49 (0.473)

Table 2 presents the results associated with the scenarios defined in Table 1. On average customers have about 90 items when shopping. The rest of the performance measures are similar across the scenarios except for the time in store and waiting time for scenario 5. Table 3 presents the results of performing a paired t-test on the time in store and total waiting time for the scenarios. Scenarios 2 and 3 allow for checking if the lane choice logic makes a difference. Scenarios 4 and 5 allow for checking if the separate payment station makes a difference.

Table 3: Paired t-test for time in store and total wait time.

	Between (2) and (3)				Between (4) and (5)			
	Time In Store		Total Wait Time		Time In Store		Total Wait Time	
Scenario	(2)	(3)	(2)	(3)	(4)	(5)	(4)	(5)
Avg.	68.27	67.68	5.92	5.33	69.77	64.95	7.46	2.49
s. d.	3.421	2.878	3.347	2.863	2.953	0.94	2.788	0.473
t-value	0.72		0.73		8.52		9.63	
p-value	0.48		0.47		0.00		0.00	

As can be seen in Table 3, the lane choice logic does not have a statistically significant effect. The time in store is a sum of shopping time, walking time, waiting time, checkout time, and payment time. The t-test results for time in store and waiting time should be equivalent since between scenarios (2) and (3), we should only expect a difference in the waiting time in checkout lines. However, having a separate payment station can significantly reduce the time spent waiting in the store. Because the payment time is a lot smaller than the sum of checkout time and bagging time (1.38 minutes vs. about 8.85 minutes), the detailed results indicate that there is not a lot of congestion in the payment area and customers get to move around more (from checkout to payment) and wait less (in both the checkout queue and the payment queue).

6 CONCLUSIONS AND FUTURE WORK

Part of this paper illustrates an application example of the JSL on an important and interesting retail situation. When comparing checkout lane selection criteria, there was no significant difference. However, highly significant results were found when having separate payment stations and a checkout area with no separate payment station. This result suggests that in a highly congested store, having separate payment stations can help reduce the congestion.

There are several further extensions that are possible within the provided modeling framework. The current models do not include jockeying. That is, once a customer picks a lane they remain in the lane. This was done because in most modern stores there are physical barriers that discourage jockeying. Thus, we modeled the customer walking along the check out areas to better pick their lanes. However, in reality, when the lines get long, customer will still jockey at the ends of the line. In addition, further work is necessary to understand the effect of single waiting lines for express checkout lanes and the perceived fairness of such configurations.

With the current framework in place, we can explore simulation optimization methods that will find optimal (minimal cost) configurations that meet desired service level requirements (e.g. probability of waiting more that 10 minutes <= 0.20). This would enable optimal staffing plans to be formed during the day and optimal configurations to be found that satisfy customer service requirements. We hope to develop a user-friendly web-application that can permit this experimentation via an on-line web portal.

REFERENCES

Alvarado, J. A., and L. M. Pulido. 2008. "Simulation and Experimental Design Applied to Sizing Supermarket Cashiers in Colombia." In *Proceedings of the 2008 Winter Simulation Conference*, edited by S. J. Mason, R. R. Hill, L. Mönch, O. Rose, T. Jefferson, and J. W. Fowler, 1356–1361. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers, Inc.

Chan, W. K. V., Y-J Son, and C. M. Macal. 2010. "Agent-Based Simulation Tutorial-Simulation of Emergent Behavior and Differences between Agent-Based Simulation and Discrete-Event Simulation." In *Proceedings of the 2010 Winter Simulation Conference*, edited by B. Johansson, S. Jain, J. Montoya-Torres, J. Hugan, and E. Yücesan, 135–150. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers, Inc.

Kelton, W. D, R. P. Sadowski, and D. T. Sturrock. 2007. *Simulation with Arena*, 4th ed., McGraw-Hill, Boston, MA, U.S.A.

Kubera, Y, P. Mathieu, and S. Picault. 2010. "An Interaction-Oriented Model of Customer Behavior for the Simulation of Supermarkets." In *IEEE International Conference on Web Intelligence and Intelligent Agent Technology (WI-IAT)*, 407–10. doi:10.1109/WI-IAT.2010.84.

Melachrinoudis, M., and M. Olafsson. 1992. "A Scheduling System for Supermarket Cashiers," *Computer & Industrial Engineering*, 23: 121–24.

Miwa, K, and S. Takakuwa. 2008. "Simulation Modeling and Analysis for in-Store Merchandizing of Retail Stores with Enhanced Information Technology." In *Proceedings of the 2008 Winter Simulation Conference*, edited by S. J. Mason, R. R. Hill, L. Mönch, O. Rose, T. Jefferson, and J. W. Fowler, 1702–1710. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers, Inc.

Opara-Nadi, G. 2005. "Electronic Self-Checkout System Vs Cashier Operated System-a Performance Based Comparative Analysis."Ph.D. thesis, Capella University.

Rossetti, M. D. 2008. "Java Simulation Library (JSL): An Open-source Object-Oriented Library for Discrete-Event Simulation in Java." *International Journal of Simulation and Process Modelling* 4, no. 1: 69-87.

Schimmel, M. 2013. "Deployment of Express Checkout Lines at Supermarkets." https://www.few.vu.nl/en/Images/werkstuk-schimmel_tcm39-335266.pdf.

Schwenke, C., V. Vasyutynskyy, and K. Kabitzsch. 2010. "Simulation and Analysis of Buying Behavior in Supermarkets." In *2010 IEEE Conference on Emerging Technologies and Factory Automation (ETFA)*, 1–4.

Schwenke, C., V. Vasyutynskyy, and K. Kabitzsch. 2011. "Analysis and Simulation of Sales Receipt Data in Supermarkets." In *2011 IEEE 16th Conference on Emerging Technologies & Factory Automation (ETFA)*, 1–8.

Terano, T., A. Kishimoto, T. Takahashi, T. Yamada, M. Takahashi. 2009. "Agent-based in-store simulator for analyzing customer behaviors in a super-market." In *Knowledge-Based and Intelligent Information and Engineering Systems.* Springer, 244–251.

Williams, E. J., K. Mohamed, and C. Lammers. 2002. "Use of Simulation to Determine Cashier Staffing Policy at a Retail Checkout." In *Proceedings 14th European Simulation Symposium, eds. Alexander Verbraeck and Wilfried Krug, 172-176.*

Yamane, S., K. Ohori, A. Obata, N. Kobayashi, and N. Yugami. 2012. "Agent-Based Social Simulation for a Checkout Layout Design of a Specific Supermarket." In *Proceedings of 13th International Workshop on Multi-Agent Based Simulation*, 153–64.

AUTHOR BIOGRAPHIES

MANUEL D. ROSSETTI is a Professor in the Industrial Engineering Department at the University of Arkansas. He received his Ph.D. in Industrial and Systems Engineering from The Ohio State University. His research and teaching interests are in the areas of simulation modeling, logistics optimization, and inventory analysis applied to manufacturing, distribution, and health-care systems. He serves as an Associate Editor for the *International Journal of Modeling and Simulation* and is active in IIE, INFORMS, and ASEE. He served as co-editor for the WSC 2004 and 2009 conference, the Publicity Chair for the WSC 2013 Conference, and is the 2015 WSC Program Chair. Dr. Rossetti is the author of

Simulation Modeling and Arena by Wiley, now in its 2nd edition. He can be contacted at rossetti@uark.edu and http://www.uark.edu/~rossetti/.

ANH T. PHAM is a graduate student at University of Arkansas. Anh obtained her B.S in Industrial and Management Engineering at South Dakota School of Mines and Technology. She is a direct admitted student to the PhD program in Industrial Engineering at the University of Arkansas. She is working toward finishing her master degree in Industrial Engineering and hopes to start her PhD in the Fall 2015. Her research interests involve simulation modeling, Markov Decision Process, and transportation logistics. Anh can be contacted at anhpham@uark.edu.