

How Do Customers Buy Them at a Supermarket? Behavior Analysis from Real Observation and Agent Simulation

Masaki Kitazawa† Fumiaki Sato† Takashi Yamada† Masakazu Takahashi‡ Takao Terano†

†Tokyo Institute of Technology, 4259-J2-52, Nagatsuda-Cho, Midori-Ku, Yokohama-Shi,
Kanagawa 226-8502, Japan

m.kitazawa, satof, tyamada@trn.dis.titech.ac.jp, terano@dis.titech.ac.jp

‡Yamaguchi University, 2-16-1, Tokiwadai, Ube-Shi, Yamaguchi 755-8611, Japan
masakazu@yamaguchi-u.ac.jp

Abstract: This paper presents a customer behavioral model for grounding the number of purchase items in Agent-Based In-Store Simulator (ABISS). ABISS is a decision support system for retail management. Using ABISS, we are able to virtually investigate the shopping paths of the customers and analyze the effect of sales promotion. To ground the number of purchase items, first we conduct a field study using Radio Frequency Identification (RFID) and analyze the collected RFID data and Point-of-Sales (POS) data. Then we developed a decision model, which determined customers' "Shopping List", "Possession Money Limit", and "Staying time".

Index Terms: Agent-Based Simulation, Agent-Based Modeling, Customer Behaviors, Radio Frequency Identification(RFID).

1. Overview

1.1. ASSUMPTIONS AND IGNORANCE:-

While setup we assumed that customers entered the shop at equal intervals and there were no queues at register as billing time is infinitely small. In this model we ignore age effect and customer collisions.

1.2. FIELD SETUP IN TARGET STORE:-

The author team of this paper did the field setup at Shimane Prefecture, Japan. We used their data to make the model. They have done the following: The condition of target store and setups of field study is summarized in Table 1. The store layout and the placement of the RFID antenna which ABISS targets are shown in Figure 1. An order of vegetable, fish, and meat categories from the entrance is widely employed in Japan. In this store, we set 10 RFID antennas on display fixtures, and 6 RFID antennas on each registers. By the antenna arrangements, we are able to measure the whole cart moves inside of the store. The same RFID tags are also set in the 53 shopping carts. The antenna setup examples are shown in Figure 1.

This field experiment was conducted from 9:00 to 21:00 on March 3-5, 2011.

1.3. EXCEPTION HANDLING AND UNIT TESTING:-

While performing unit testing, we found few exceptions where the customers used to collide with each other. We avoided these exceptions by writing conditions to change the direction upon facing a customer or a wall or a shelf.

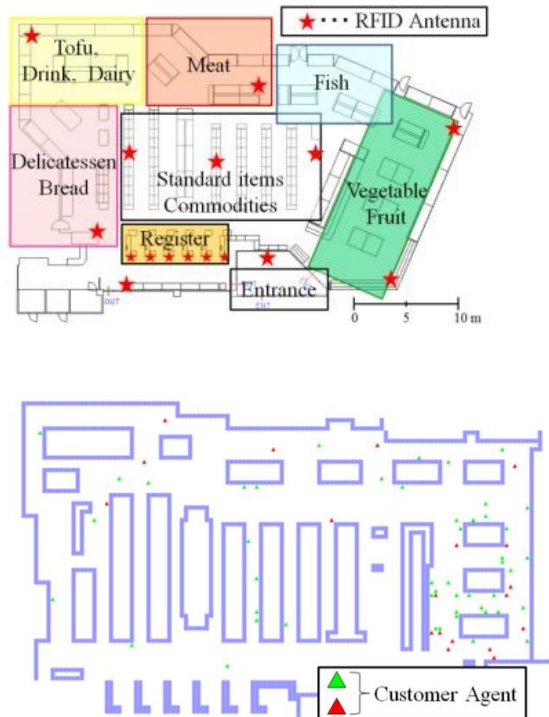


Figure 1. Snapshot of ABISS

Table 2. Condition of Target Store and Setups of Field Study

Location	Shimane Prefecture, Japan
Opening Time	9:00 – 21:00
Average of Customers Per Day	2000 customers
Experimental Period	March 3 – March 5, 2011
RFID Antennas	10 on display fixture 6 on each register
RFID Tags	53 on shopping cart (Half of shopping cart)

First we spawned the customers at the entrance of the store(yellow line in interface). Then we gave a few properties to it. Then we made it move to the items of the list. After this stage we made it move to the nearest item on the list. After successful completion of the list or allocated time or allocated money, the customer returns to the register and then checkouts.

1.4. ENTITIES, STATE VARIABLES AND SCALES:-

Agent:- customers

Environment:- register counters, walls and shelves

Customer properties:-

time-limit ; The amount of time the customer wants to stay in the store

spent-time ; The amount of time the customer spent so far.

possession-money ; The amount of money the customer holds before the start of the shopping

section-budget-list ; This list is created to assign random budgets for each section of the store
section-item-list ; This list holds all the items in section-wise format.

shopped-item-list ; This list holds all the items the customer purchases.

skipped-item-list ; This list holds all the items the customer doesn't buy, due to some reasons.

out-stock-list ;

money-spent ; The amount of money the customer has spent so far.

money-remaining ; The amount of money that is left after purchasing a product.

next-section ; This variable is used to easily retrieve the index of the required section in the
section-budget-list.

next-section-name ; We store the name of the section the customer is currently in.

next-item ; The item-id of the next item the customer wants to purchase will be stored in this.

next-item-index ; We store the index of the next item the customer wants to purchase so that it
will be easy to retrieve the item from the list [section-item-list]

head-direction ; This variable indicates the current direction of the customer.

prev-head ; This variable is just a helper variable used to align the turtle in a proper way.

align ; This variable is just a helper variable used to align the turtle in a proper way.

Global variables:- total-customers total-items-purchased total-possession-money total-staying-
time total-money-spent total-customers-finished-shopping satisfaction-level-list

Temporal scale:- 1 tick/sec. Spatial scale:-

1.5. OVERVIEW PROCESS:-

Customers enter the store through the entrance and start looking for the nearest item in the shopping list and move near that item. They move continuously until the list is empty or they are out of time or they don't have enough money. Then they move to register and checkout.

2. DESIGN CONCEPTS

2.1. BASIC PRINCIPLES:-

This section describes how the model implements a set of basic concepts that are important for designing ABMs. The concept of our model is simple, we create an interface given in the article and work with the customers with respective models. This model presents the customers' behavioral model for grounding the number of purchasing items in a supermarket and experimental results of the simulation. First, we conduct a field study using RFID and analyze the collected RFID data and Point-of-Sales (POS) data to build the model. Second, we built a decision model in which we determined "Shopping List", "Possession Money Limit", and "Staying time". The result has revealed that we ground the number of purchasing items obtained by the POS data, if we consider only "Shopping List" and "Possession Money Limit" in ABISS.

2.2. EMERGENCE:-

The model runs for 43200 steps of simulation, an error rate is calculated which shows a difference with the number of purchased items, we have found that the Shopping list and Money Model gives the least error rate. It suggests that we can ground the number of purchasing items even if we do not consider the "Staying time".

This model is run for 3 times with a different model each time. The 3 different models or cases are:- i) Shopping list and money model. ii) Shopping list and time model. iii) Shopping list, money and time model. Each model is run for 43200 steps. Then we compare the results with P.O.S data and calculate error rate.

2.3. ADAPTATION:-

The agents in the model are given different values for the values, all the necessary values are initiated, they are directed towards the patches that have their products, different models behave differently according to the conditions allotted.

Customer agents are given different properties, i.e., shopping list, possession money and time limit. The customer enters the shopping mall at the entrance and then goes to the item whose name is present in the shopping list. Possession money and staying time limit are factors other than the shopping list which lets the consumer go to the billing counter.

2.4. OBJECTIVES:-

The objective of the customer is to buy as many products as possible in the shopping list using only the possession money in a given time limit. The conditions which force customers to move to registrar is decided at model selection.

2.5. LEARNING:-

The agents make some decisions based on the money possessed, i.e., If they don't have enough money to buy a particular item, they skip that item and move to the next item in the shopping list. Also if the time allocated is over, the customers move towards registers.

2.6. SENSING:-

The customers check the expiry date and price of the item. If the item is expired, then the customer skips the item and goes to the next item. Else, he checks the price and money available, if he doesn't have enough money to buy the item, he skips the item, else he buys the item. The expiry date is given to the items randomly at the start of the simulation.

The possessed money is also given randomly to each customer at the time of entrance, and the customers have to buy items with this money only.

2.7. INTERACTIONS:-

Every customer goes in such a way that they do not collide with walls or another customer. They check 1 patch surrounding them, to find any obstacle (walls, another customer).

2.8. STOCHASTICITY:-

Time limit:- generated randomly in the range given at the slider input

Possession money:- generated randomly and the range is determined by the slider input

Section budget list:- the possessed money is split into different sections randomly and the customer uses only that much money in that section and if some money is remaining in that section budget, then that money is added to the next section budget.

2.9. OBSERVATION:-

We need to collect various values for different models, for the shopping list Money model Total Purchased items is the criteria to evaluate the models performances in different runs. System behaviour depends on the count of Purchased items Behaviour Space is needed to experiment

the values of Possession money and Staying time by running all three models. We are able to virtually investigate the shopping paths of the customers and analyze the effect of sales promotion. The output is shown in the interface and the graphs are plotted in .xlsx files. We need to analyze which factors affect the Purchasing items count and those affect sales promotion. The outputs and analysis required to differentiate and compare the models are satisfaction-level(interface), total purchased items(interface), graphs for analysis (.xlsx files). Total number of items purchased is needed to solve the problem the model was designed for.

3. DETAILS

3.1. INITIALISATION:-

We initialise the model by clicking the setup button. Then the patches are given colours, black for the boundary walls of the store, blue for shelves of the items, purple for the register(counter) section, yellow for entrance and red for exit.

Each blue patch is given its respective section name given in the research paper. Stock, expiry date, item id is given randomly at the start of the setup to differentiate between each item in a particular section.

The breed of the agents is referred to as customers.

The customers are given some properties and some of their values are given randomly at the time of entrance based on the real values range given in the research paper. The properties which are initialised at the beginning are:- Time limit:- generated randomly in the range given at the slider input

Possession money:- generated randomly and the range is determined by the slider input

Section budget list:- the possessed money is split into different sections randomly and the customer uses only that much money in that section and if some money is remaining in that section budget, then that money is added to the next section budget.

Section item list:- for each section in the section budget list, the amount is divided into a random number of items with each item having its own cost.

3.2. INPUT DATA:-

We are giving 12 inputs. They are:- Run time choice, behavioral model choice, maximum customers entering, customers entrance rate, time limit range, sample size(for sampling model), minimum and maximum possessed money, minimum and maximum section budget, minimum items and maximum items in a section.

3.3. SUBMODELS(Behaviour):-

There are three submodels in this model:- (a) Shopping List Money Model

(b) Shopping List Time Model

(c) Shopping List, Money Time Model