

Simulazione di un supermercato con Anylogic

Odore Marco

28 ottobre 2017

Docenti: Trubian Marco, Malchiodi Dario
Corso: Simulazione e Teoria delle code

Indice

1	Scopo del progetto	1
2	Agent Based modeling	1
2.1	Gli agenti Customer e genericFood	2
3	Il supermercato e i servizi	4
4	Flow Diagrams	5
4.1	Customer Flow	5

1 Scopo del progetto

L'obiettivo del progetto è quello di simulare, tramite il software Anylogic¹, diverse dinamiche riguardanti un supermercato, come ad esempio il flusso della clientela, la schedulazione del personale e i diversi servizi che possono essere presenti nell'attività.

Il tutto è realizzato tramite la versione *learning edition* del software, che presenta alcune limitazioni, come ad esempio il numero massimo di tipologie definibili per gli agenti e un numero massimo per la loro generazione durante l'esecuzione della simulazione².

2 Agent Based modeling

Data la natura complessa del problema, che possiede moltissime attività parallele e concorrenti da simulare, si è deciso di sfruttare il modello basato su agenti.

¹<https://www.anylogic.com/>

²Durante la simulazione saranno generabili un massimo di 50000 agenti complessivi e in fase di costruzione del modello non è stato possibile definire più di 10 agenti.

Nello specifico sono definiti i seguenti agenti:

- **Customer**: il cliente del supermercato.
- **Worker**: i diversi addetti dei reparti di panetteria, pescheria e macelleria.
- **Warehouseman**: i magazzinieri che si occupano di rifornire gli scaffali.
- **Cashier**: i cassieri per il servizio di pagamento.
- **Cart**: i carrelli utilizzati dai clienti.
- **GenericFood**: la risorsa utilizzata dai magazzinieri per rifornire gli scaffali.
- **AutomaticCashierMachine**: la cassa automatica per il servizio di pagamento.
- **InfoPointHelper**: gli addetti dell'info point.

La maggior parte degli agenti è definita per poterne differenziare l'aspetto all'interno della simulazione, e solo **Customer** e **GenericFood** possiedono un'ulteriore caratterizzazione.

2.1 Gli agenti **Customer** e **genericFood**

Il **Customer** possiede diverse variabili e parametri. Nello specifico:

- Variabile **ItemsToBuy**: È un dizionario con coppie Prodotto(String)/Quantità(int), che contiene i prodotti che il cliente vuole comprare e relativa quantità.
- Variabile **Bought**: Un booleano che indica se il cliente ha comprato qualcosa, inizializzato a false.
- Variabile **CounterBuy**: un contatore(int) che indica quanti prodotti il cliente ha nel carrello in quel momento.
- Variabile **NeedsInfo**: Un booleano che indica se il cliente necessita di chiedere informazioni all'infopoint.
- I parametri **needsInfoRate**, **needsMeat**, **needsBread**, **needsFish**, **needsOther**: che rappresentano le diverse probabilità di acquisto (o di richiesta info) che un cliente generico possiede entrando nel supermercato³.

L'agente è inoltre caratterizzato da uno state chart (Figura 1), con tre diversi stati:

- **InitialState**: lo stato iniziale del cliente entrando nel supermercato.
- **WantsToBuy**: lo stato del cliente quando è in fase di acquisto dei prodotti.
- **WantsToGoAway**: lo stato finale del cliente, quando decide di lasciare il supermercato.

³Ad esempio, nella simulazione è stata definita la probabilità che un cliente voglia comprare del pane entrando nel supermercato a 0.7(cioè sette clienti su dieci).

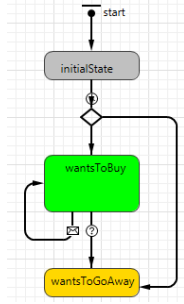


Figura 1: Il diagramma di stato dell'agente Customer.

Quando il cliente entra nello stato InitialState, viene eseguito del codice (Figura 2) che permette di inizializzare la variabile ItemsToBuy di Customer, sfruttando i diversi parametri descritti in precedenza che definiscono le probabilità di acquisto del cliente per i diversi prodotti.

Una volta inizializzata la lista della spesa, se questa è vuota il Customer passerà allo stato WantsToGoAway, altrimenti entrerà nello stato loop WantsToBuy, finché non avrà collezionato tutti gli oggetti, per poi finire anch'esso nello stato finale WantsToGoAway.

```

1  shapeBody.setFillColor(black);
2  String []list = {"meat", "bread", "fish", "generic", "generic1", "
   generic2"};
3  ArrayList<String> possibilities = new ArrayList<String>(Arrays.
   asList(list));
4  java.util.Collections.shuffle(possibilities);
5  double res;
6  for(String poss: possibilities){
7      res = uniform(0,1);
8      int quantity = uniform_discr(1,4);
9      if (res<=getParameterFromString(poss))
10     {
11         addObject(poss, quantity);
12     }
13 }
14 res = uniform(0,1);
15 if(res<=needsInfoRate)
16     needsInfo = true;
  
```

Figura 2: Inizialmente viene eseguito uno shuffle sulla lista di stringhe dei possibili oggetti da acquistare, per differenziare l'ordine di acquisto di ogni cliente, e poi viene simulata un'estrazione da una distribuzione di Bernoulli sfruttando una distribuzione uniforme e il parametro del prodotto di riferimento. Viene inoltre simulata l'estrazione da una distribuzione uniforme discreta (da 1 a 4 oggetti) per la quantità del prodotto da acquistare. Il metodo `getParameterFromString` è una ulteriore funzione custom, definita per il recupero dei valori dei parametri.

Per quanto riguarda l'agente `genericFood`, questo possiede due attributi per caratterizzarlo, ovvero:

- **Type:** Di tipo String. Specifica la tipologia di prodotto (Es. "meat").
- **Quantity:** Di tipo Int. Specifica la quantità di prodotto da acquistare.

3 Il supermercato e i servizi

Il supermercato si estende su una superficie rettangolare, con l'area centrale dedicata agli scaffali per i prodotti generici, la parte destra ai servizi al banco, la parte inferiore ai servizi di cassa, e i magazzini disposti sia sul lato superiore (2) che sul lato sinistro (1). Nella figura 3 è mostrata la planimetria del supermercato e le sue principali aree di interesse.

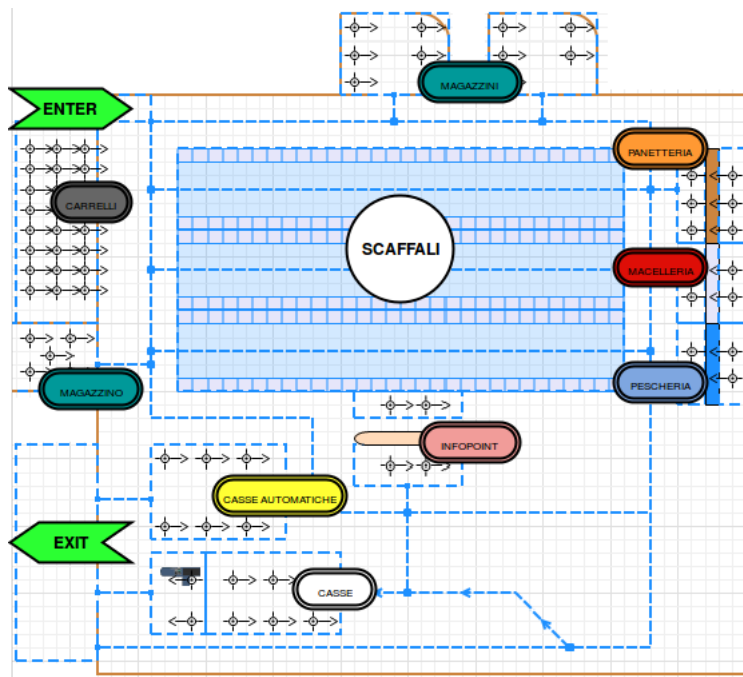


Figura 3: Le principali aree di interesse del supermercato.

Ognuno dei servizi implementati possiede un numero variabile di risorse (rappresentate dagli agenti), che vengono schedate in base all'orario del giorno (ad eccezione delle casse automatiche). Nello specifico abbiamo:

- Servizio al banco per prodotti di panetteria (agenti Worker).

- Servizio al banco per prodotti di macelleria (agenti Worker).
- Servizio al banco per prodotti di pescheria (agenti Worker).
- Servizio di infopoint (agenti InfoPointHelper).
- Servizio di pagamento con cassiere (agenti Cashier).
- Servizio di pagamento con cassa automatica (agenti AutomaticCashier-Machine).

Oltre ai servizi alla clientela sono simulate anche delle attività di rifornimento degli scaffali (agenti WarehouseWorker) per tre diverse tipologie di prodotti (agenti GenericFood), che per comodità, e a causa delle limitazioni della versione learning di anylogic, si differenziano unicamente per il colore (verde, viola, giallo).

4 Flow Diagrams

Per caratterizzare il comportamento dei clienti (agente Customer) e dei prodotti generici da scaffale (genericFood) all'interno del supermercato, sono definiti 4 diagrammi di flusso, di cui 3 dedicati alle tre diverse tipologie di prodotti.

4.1 Customer Flow

I tempi di arrivo (e generazione) dell'agente Customer nel supermercato sono definiti nel flow (Figura 4) all'interno di un blocco sorgente, chiamato **customer-Source**, grazie ad uno scheduling temporale sulle 24 ore. Ogni ora del giorno è infatti caratterizzata da un *rate*⁴ di arrivo specifico. Questo ha permesso di simulare il diverso flusso di clientela che caratterizza la giornata di un supermercato, modulando ad esempio su frequenze più basse gli arrivi notturni, per poi aumentarli durante il giorno (con dei picchi nelle ore di punta).

Il percorso che il Customer seguirà all'interno del flow (e quindi all'interno del supermercato) è definito in base al suo stato interno e quindi al valore delle sue variabili (itemsToBuy, Bought, etc).

⁴Parametro di una distribuzione esponenziale, con media $\frac{1}{rate}$

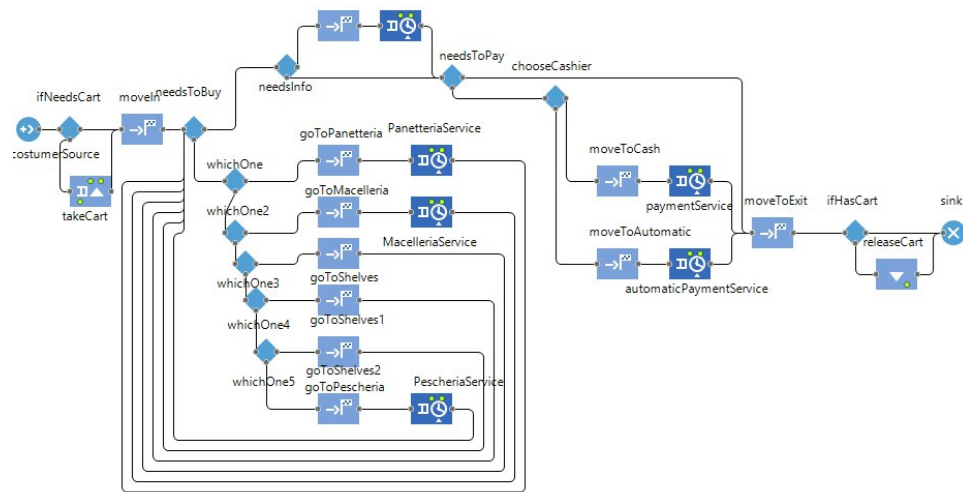


Figura 4: Lo schema di flusso del customer nel supermercato.