

Simulazione di un supermercato con Anylogic

Odore Marco

25 ottobre 2017

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

Indice

1	Scopo del progetto	1
2	Agent Based modeling	2
3	Gli agenti Customer e genericFood	2
4	Il supermercato e i servizi	4

1 Scopo del progetto

L'obiettivo del progetto è stato 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 stato 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.

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.

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.

Nello specifico sono stati definiti:

- **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 stata definita per poterne differenziare l'aspetto all'interno della simulazione, e solo **Customer** e **GenericFood** possiedono un'ulteriore caratterizzazione.

3 Gli agenti Customer e genericFood

L'agente **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 info) che un cliente generico possiede entrando nel supermercato³.

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).

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 del cliente quando decide di andare via.

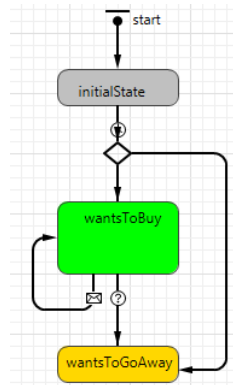


Figura 1: Il diagramma di stato dell'agente Customer.

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 che definiscono le probabilità di acquisto del cliente per i diversi prodotti.

```

1 shapeBody.setFillColors(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 per simulare la quantità del prodotto da acquistare.

4 Il supermercato e i servizi

Nella figura 2 è mostrata la planimetria del supermercato e le sue principali aree di interesse.

I servizi simulati per i clienti della struttura sono i seguenti:

- Servizio al banco per prodotti di panetteria.
- Servizio al banco per prodotti di macelleria.
- Servizio al banco per prodotti di pescheria.
- Servizio di infopoint.
- Servizio di pagamento con cassiere.
- Servizio di pagamento con cassa automatica.

Sono state inoltre simulate delle attività di rifornimento degli scaffali per tre diverse tipologie di prodotti.

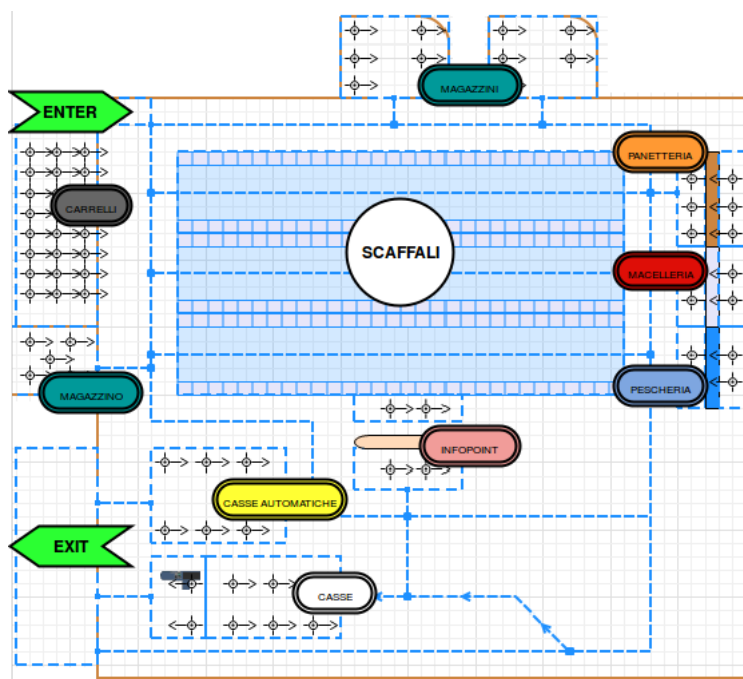


Figura 3: Le principali aree di interesse del supermercato.