

Soluzione dell'esercizio del 20 Aprile 2004

1. Modello concettuale

Osserviamo anzitutto che il sistema autostradale (la Mappa) è fatta di segmenti; che un segmento ha un casello di ingresso e uno di uscita e che può avere zero o più stazioni di rifornimento. Un'auto può essere in percorrenza di un segmento o a una stazione di rifornimento (Fig. 1).

Alternativamente, l'associazione di percorrenza potrebbe avere molteplicità 1 sul lato Segmento. In questo caso l'interpretazione sarebbe che un'auto è comunque in percorrenza di un segmento, anche quando è a una stazione di rifornimento. In questo caso la relazione di rifornimento sarebbe non più esclusiva rispetto alla precedente. Nel seguito si segue il modello di Fig. 1.

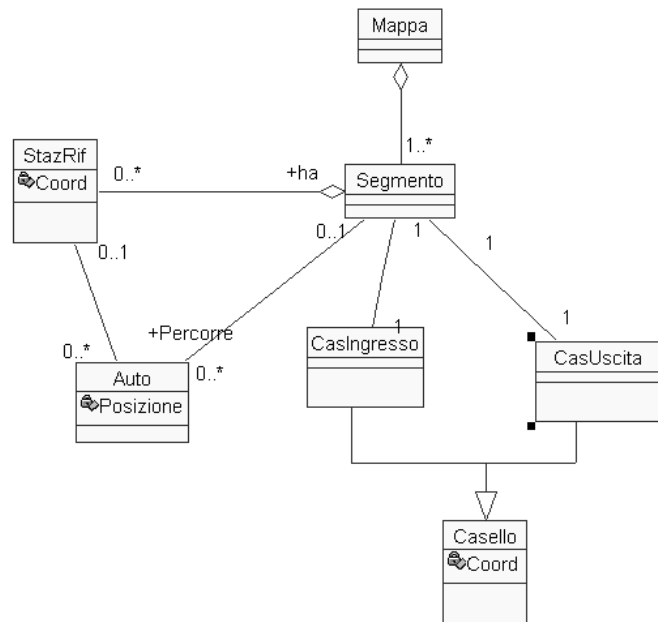


Figura 1: Modello concettuale.

2. Interfacce per il disegno

Il disegno richiede che tutti gli oggetti siano in grado di disegnarsi. Con `draw()` si indica il metodo che disegna l'oggetto cui appartiene il metodo stesso. Il disegno di un segmento è la tracciatura di una linea tra punti in cui sono i caselli. Un'auto si disegna in base alla propria posizione (una coordinata su un segmento o a una stazione). Le interfacce sono quindi quelle di Figura 2. La figura indica che per effettuare il disegno, l'oggetto di controllo (il simulatore) deve semplicemente chiamare `Mappa.draw()`.

3. Interfaccia classe auto

Ai fini della simulazione la classe auto deve prevedere questi attributi (tendenzialmente tutti privati):

- a) la velocità di marcia V ;
- b) il parametro P ;
- c) il tempo di rifornimento T_s ;
- d) l'ora di ingresso al casello;
- e) la posizione;
- f) i Km percorsi dall'ultimo rifornimento;

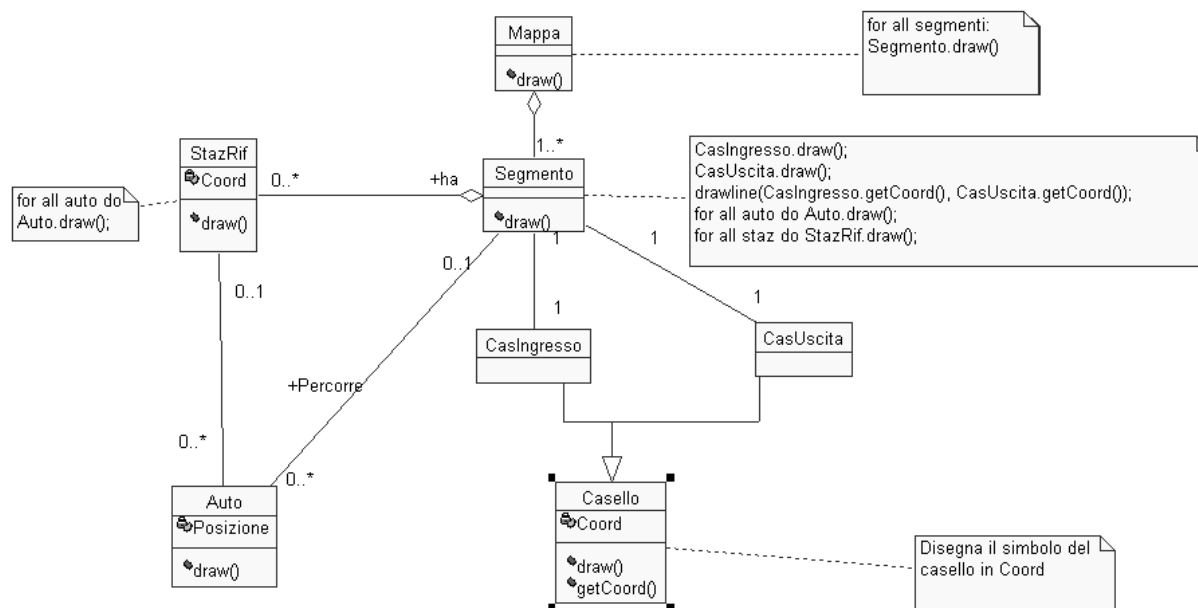


Figura 2: Interfacce per il disegno.

- g) lo stato (marcia/rifornimento/..);
- h) l'istante di tempo di inizio sosta (significativo solo in caso di sosta al rifornimento).

Gli attributi a), b) e c) sono costanti e sono caratteristici di ogni auto. Essi vengono generati dal casello di entrata attraverso un meccanismo di scelta casuale all'atto dell'istanziamento. I restanti attributi vengono aggiornati ad ogni chiamata del metodo tick() della classe auto. Il metodo tick() viene chiamato ad ogni avanzamento del tempo. Esso è il solo metodo di interfaccia necessario alla simulazione. Il metodo fa svolgere queste azioni a seconda dello stato:

- in stato di marcia:
 - calcolo della nuova posizione;
 - eventuale decisione di sosta o di uscita.

Nel primo caso si ha un cambiamento dello stato e viene preso traccia dell'istante di inizio della sosta. Secondo lo schema di Fig. 1 la macchina deve comunicare col segmento che sta lasciando e con la stazione di rifornimento, in modo che essa venga aggregata alle macchine in rifornimento alla stazione.

Nel secondo caso la macchina chiama il metodo esco() del Casello di uscita, comunicandogli i suoi dati, in modo che questo possa calcolare la durata del viaggio (e l'eventuale costo). La macchina non viene subito eliminata e passa in stato "uscita" (vedi sotto).
- in stato di sosta:
 - eventuale decisione di fine sosta. In caso affermativo passaggio allo stato di marcia e azzeramento dei Km percorsi dall'ultimo rifornimento.

In Figura 3 viene data la specifica di interfaccia di Auto.

4. Diagramma di stato classe auto

Il diagramma ha l'aspetto di Fig. 4 (Si veda anche la Nota conclusiva).

5. Casello di ingresso

Il casello di ingresso è responsabile dell'istanziamento di nuove macchine. Nelle ipotesi fatte il casello presenta



Figura 3: Interfaccia di Auto.

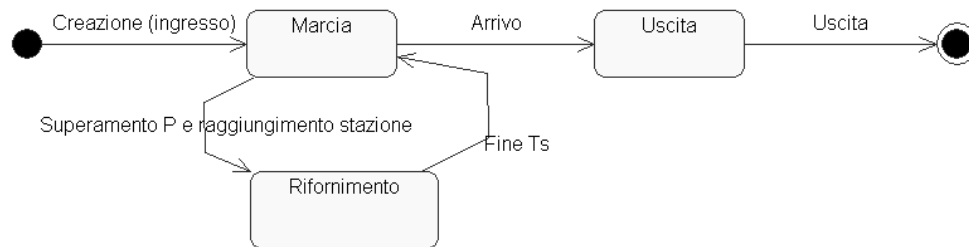


Figura 4: Diagramma di stato di un oggetto Auto.

come interfaccia il solo metodo `tick()`. Il metodo decide, in base a predefinite distribuzioni di probabilità, se deve essere istanziata una nuova macchina. In caso affermativo la macchina viene istanziata, assegnando i valori iniziali agli attributi e informando il segmento in modo che questo possa aggregarla alle auto sul segmento.

Gli attributi `V`, `P` e `Ts` sono costanti (`P` sarà scelto in modo di garantire almeno il raggiungimento della prossima stazione di servizio o dell'uscita, se sul segmento non ci sono stazioni.a meno che non si voglia considerare anche la possibilità di macchine ferme per mancanza di benzina!)

6. Casello di uscita

Il casello di uscita è responsabile dell'eliminazione della macchina che esce, del calcolo del tempo di percorrenza¹, etc. Il Casello di uscita deve presentare il metodo `esco()`, chiamato da una macchina che lo raggiunge. Poiché l'eliminazione dell'auto che ha invocato il metodo `esco()` potrebbe determinare un errore al tempo di esecuzione (dove ritornerebbe il metodo `esco()` ?), conviene supporre che il metodo faccia prendere traccia al casello di uscita della macchina da eliminare. L'eliminazione si avrà al prossimo `tick()`. Per questo motivo è stato previsto lo stato `Uscita` nel diagramma di Fig. 4.

In Figura 5 le interfacce dei due tipi di caselli.

Nota conclusiva

Per realizzare la simulazione basta immaginare che anche `Mappa`, `Segmento` e `StazRif` abbiano il metodo `tick()`.

Un passo di simulazione si effettua invocando `Mappa.tick()` che a sua volta invoca tutti i `Segmento.tick()`. Il `tick()` del segmento invoca `Auto.tick()` di tutte le auto sul segmento e i `tick()` delle `StazRif` sul segmento. `StazRif.tick()` invoca i `tick()` delle auto in stazione. Il `tick` del segmento invoca `CaselloIngresso.tick()` e `CaselloUscita.tick()`. Infine, `Auto.tick()` può invocare `esco()` del `CaselloUscita`.

La precedente spiegazione non tiene conto del fatto che la macchina per calcolare la propria posizione e altro può avere bisogno di informazioni dal Segmento su cui si trova. Non costa molto immaginare che esistano dei metodi di Segmento, non indicati, che forniscono queste indicazioni (lunghezza, distanza prossima stazione di servizio, etc..).

¹Dal punto di vista della programmazione orientata agli oggetti, il calcolo del tempo di percorrenza starebbe meglio nell'oggetto Auto. Siccome però l'auto viene eliminata, questa dovrebbe comunque comunicare al casello i suoi dati, in modo che il casello ne tenga traccia per eventuali statistiche (che, nella pratica, rappresentano il risultato effettivo della simulazione).

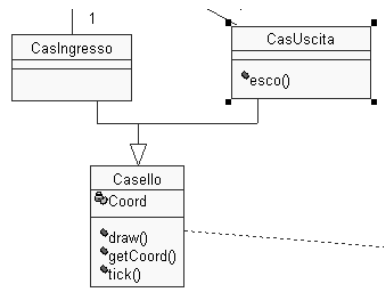


Figura 5: Interfaccia dei caselli.

Infine si osservi che lo stato di Uscita degli oggetti Auto non è strettamente necessario. Infatti la macchina potrebbe essere lasciata in stato di Marcia, per essere eliminata dal casello di uscita sul prossimo tick(). L'inserimento di questo stato conferisce maggiore coesione all'oggetto Auto, in quanto lascia al suo interno la decisione di cosa fare a seconda dello stato in cui si trova (non accadrà che vengano modificati i valori dei suoi dati di viaggio mentre si trova in uscita).